

IBM DB2 10.1  
para Linux, UNIX y Windows

*Guía y consulta de DB2 Workload  
Management*





IBM DB2 10.1  
para Linux, UNIX y Windows

*Guía y consulta de DB2 Workload  
Management*



**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información general contenida en el apartado Apéndice E, "Avisos", en la página 563.

**Nota de edición**

Este manual es la traducción del original en inglés *IBM DB2 10.1 for Linux, UNIX, and Windows DB2 Workload Management Guide and Reference* (SC27-3891-00).

Este documento contiene información propiedad de IBM. Se proporciona según un acuerdo de licencia y está protegido por la ley de la propiedad intelectual. La información contenida en esta publicación no incluye ninguna garantía de producto, por lo que ninguna declaración proporcionada en este manual deberá interpretarse como tal.

Puede realizar pedidos de publicaciones de IBM en línea o a través del representante de IBM de su localidad.

- Para solicitar publicaciones en línea, vaya a IBM Publications Center en <http://www.ibm.com/shop/publications/order>
- Para encontrar al representante local de IBM que le corresponde, vaya a la sección Worldwide Contacts de IBM Directory en <http://www.ibm.com/planetwide/>

Para realizar pedidos de publicaciones de DB2 desde DB2 Marketing and Sales, en los EE.UU. o en Canadá, llame al 1-800-IBM-4YOU (426-4968).

Cuando envía información a IBM, está otorgando a IBM el derecho no exclusivo de utilizar o distribuir la información de cualquier forma que considere adecuada sin incurrir por ello a ninguna obligación para con usted.

© Copyright IBM Corporation 2007, 2012.

# Contenido

**Acerca de este manual . . . . . ix**

**Capítulo 1. Introducción a los conceptos de la gestión de carga de trabajo de DB2 . . . . . 1**

Dominios de la gestión de carga de trabajo . . . . . 1  
Autorización (WLMADM) de administrador de gestión de carga de trabajo . . . . . 3  
Preguntas frecuentes sobre la gestión de carga de trabajo de DB2. . . . . 4

**Capítulo 2. Identificación del trabajo . . 15**

Actividades . . . . . 15  
Sentencias DDL de gestión de carga de trabajo . . 18  
Identificación del trabajo por origen con cargas de trabajo . . . . . 19  
    Asignación de cargas de trabajo . . . . . 24  
    Cargas de trabajo por omisión . . . . . 26  
    Creación de una carga de trabajo . . . . . 31  
    Alteración de una carga de trabajo . . . . . 33  
    Habilitación de una carga de trabajo . . . . . 35  
    Inhabilitación de una carga de trabajo . . . . . 36  
    Descarte de una carga de trabajo . . . . . 37  
    Cómo permitir apariciones de una carga de trabajo para acceder a la base de datos . . . . . 37  
    Cómo impedir que las apariciones de una carga de trabajo accedan a la base de datos. . . . . 38  
    Concesión del privilegio USAGE en una carga de trabajo . . . . . 39  
    Revocación del privilegio USAGE en una carga de trabajo . . . . . 40  
    Ejemplo: Asignación de cargas de trabajo . . . . . 41  
    Ejemplo: Asignación de cargas de trabajo cuando los atributos de la carga de trabajo sólo tienen un valor . . . . . 45  
    Ejemplo: Asignación de una carga de trabajo para una unidad de trabajo cuando existen varias cargas de trabajo . . . . . 47  
    Ejemplo: Asignación de una carga de trabajo cuando los atributos de la carga de trabajo tienen varios valores. . . . . 50  
Identificación de trabajos por tipo, por coste o por medio de los datos a los que se accede con clases de trabajo . . . . . 52  
    Clases de trabajo y conjuntos de clases de trabajo 55  
    Orden de evaluación de clases de trabajo en un conjunto de clases de trabajo . . . . . 63  
    Asignación de actividades a clases de trabajo . . 64  
    Clasificaciones de trabajo soportadas por los umbrales . . . . . 65  
    Ejemplo: Análisis de cargas de trabajo por tipo de actividad . . . . . 66  
    Ejemplo: Utilización de un conjunto de clases de trabajo para gestionar tipos específicos de actividades . . . . . 67

Ejemplo: Cómo trabajar con una clase de trabajo definida con la palabra clave ALL . . . . . 68

**Capítulo 3. Gestión de actividades. . . 73**

Asignación de recursos con clases de servicio . . . 73  
Superclases y subclases de servicio por omisión 76  
Correlación de actividades con clases de servicio 78  
Prioridad de agente de clases de servicio . . . 82  
Prioridad de captación previa de clases de servicio. . . . . 83  
Prioridad de agrupación de almacenamientos intermedios de clases de servicio . . . . . 84  
Estados de las conexiones y de las actividades en una clase de servicio . . . . . 85  
Entidades de nivel del sistema cuya gestión no realizan las clases de servicio . . . . . 87  
Creación de una clase de servicio . . . . . 87  
Alteración de una clase de servicio . . . . . 89  
Descarte de una clase de servicio . . . . . 93  
Ejemplo: Utilización de clases de servicio . . . 94  
Ejemplo: Análisis de una ralentización de sistema relacionada con la clase de servicio . . . . . 99  
Ejemplo: Investigación del uso de agentes por parte de una clase de servicio . . . . . 101  
Aplicación de controles a tipos de actividades con conjuntos de acciones de trabajo . . . . . 102  
    Cómo las clases de trabajo, los conjuntos de clases de trabajo, las acciones de trabajo y los conjuntos de acciones de trabajo funcionan conjuntamente y se asocian con otros objetos de DB2 . . . . . 103  
    Acciones de trabajo y conjuntos de acciones de trabajo. . . . . 105  
    Acciones de trabajo y el dominio de conjunto de acciones de trabajo . . . . . 117  
    Umbrales que pueden utilizarse en acciones de trabajo. . . . . 121  
    Aplicación de acciones de trabajo a actividades de la base de datos . . . . . 121  
    Control de simultaneidad a nivel de carga de trabajo mediante conjuntos de acciones de trabajo. . . . . 123  
    Comparación de cargas de trabajo y conjuntos de acciones de trabajo . . . . . 125  
    Ejemplo: Utilización de un conjunto de acciones de trabajo de base de datos y un umbral de base de datos . . . . . 127  
    Ejemplo: Utilización de conjuntos de acciones de trabajo para determinar los tipos de trabajo que están en ejecución . . . . . 129  
Control del trabajo con umbrales . . . . . 130  
    Dominio de umbral y ámbito de implantación 134  
    Orden de evaluación de umbrales . . . . . 135  
    Creación de un umbral . . . . . 138  
    Alteración de un umbral . . . . . 139

Descarte de un umbral . . . . .	139
Ejemplo: Utilización de umbrales . . . . .	140
Umbrales de conexión . . . . .	142
Umbrales de actividad . . . . .	143
Umbrales agregados . . . . .	153
Umbrales de unidad de trabajo . . . . .	160
Disminución de la prioridad del trabajo en curso	161
Scripts de ejemplo de disminución de la	
prioridad . . . . .	166
Nueva correlación de actividades entre	
subclases de servicio . . . . .	173
Visión general del asignador de gestión de carga	
de trabajo . . . . .	176
Asignador de gestión de carga de trabajo . . . . .	177
Cuotas de CPU fijas . . . . .	200
Cuotas de CPU flexibles . . . . .	209
Límite de CPU . . . . .	215
Utilización de recursos de CPU mínima para	
que se considere activa la clase de servicio . . . . .	230
Nivel de simultaneidad de asignación . . . . .	234
Supervisión y ajuste del rendimiento del	
asignador de gestión de carga de trabajo . . . . .	237
Cancelación de actividades . . . . .	241

## Capítulo 4. Supervisión e intervención 243

Supervisión en tiempo real con funciones de tabla	243
Ejemplo: Utilización de funciones de tabla de la	
gestión de carga de trabajo de DB2 . . . . .	248
Ejemplo: Supervisión del comportamiento actual	
del sistema en diferentes niveles utilizando las	
funciones de tabla de la gestión de carga de	
trabajo de DB2 . . . . .	250
Ejemplo: Obtención de estadísticas puntuales de	
clases de servicio . . . . .	253
Ejemplo: Agregación de datos utilizando	
funciones de tabla de gestión de carga de	
trabajo de DB2 . . . . .	254
Ejemplo: Determinación de las actividades que	
un umbral de WLM coloca en cola y su orden	
en la cola . . . . .	254
Supervisión histórica con supervisores de sucesos	
de WLM . . . . .	256
Datos de supervisión disponibles . . . . .	262
Procedimientos almacenados de gestión de carga	
de trabajo de DB2 . . . . .	264
Estadísticas para objetos de la gestión de carga de	
trabajo de DB2 . . . . .	265
Recopilación de estadísticas y supervisión con	
nueva correlación de actividades . . . . .	276
Histogramas en la gestión de carga de trabajo	
Herramienta de análisis histórico . . . . .	288
Recopilación de las estadísticas de gestión de	
carga de trabajo utilizando un supervisor de	
sucesos de estadísticas . . . . .	288
Restablecimiento de estadísticas en objetos de	
gestión de carga de trabajo de DB2 . . . . .	291
Métricas de supervisión para la gestión de carga de	
trabajo de DB2 . . . . .	292
Funciones de tabla de gestión de carga de trabajo e	
integración del supervisor de instantáneas . . . . .	293
Supervisión de violaciones de umbral . . . . .	294

Cómo generar notificaciones por correo	
electrónico para violaciones de umbral . . . . .	296
Recopilación de datos para actividades	
individuales . . . . .	299
Importación de información de actividad en el	
Asesor de diseño . . . . .	301
Cancelación de actividades . . . . .	302
Directrices para capturar información sobre una	
actividad malintencionada e investigarla . . . . .	302
Modelado del rendimiento de gestión de carga de	
trabajo . . . . .	303
Ejemplo: Captura de información sobre una	
actividad para su posterior análisis . . . . .	304

## Capítulo 5. Integración con los gestores de carga de trabajo del sistema operativo . . . . . 307

Integración del gestor de carga de trabajo de AIX	
con la gestión de carga de trabajo de DB2 . . . . .	307
Integración de la gestión de carga de trabajo de	
Linux con la gestión de carga de trabajo de DB2. . . . .	314

## Capítulo 6. Guía de aprendizaje de gestión de carga de trabajo de DB2. . 321

Ejercicio 1: Iniciación a la supervisión básica	
utilizando objetos de la gestión de carga de trabajo	
de DB2 por omisión . . . . .	321
Ejercicio 2: Aislamiento de actividades mediante	
clases de servicio y cargas de trabajo . . . . .	327
Ejercicio 3: Utilización de umbrales para controlar	
actividades malintencionadas y utilización del	
supervisor de violaciones de umbral. . . . .	334
Ejercicio 4: Diferenciación de actividades por tipo	
de actividad . . . . .	337
Ejercicio 5: Utilización de histogramas para clases	
de servicio . . . . .	345
Ejercicio 6: Investigación de retardos con funciones	
de tabla de WLM . . . . .	354
Ejercicio 7: Cancelación de una actividad en curso	
Ejercicio 8: Descubrimiento de qué tipos de	
actividades se ejecutan en el sistema. . . . .	358
Ejercicio 9: Captura de información detallada	
acerca de una actividad en ejecución . . . . .	361
Ejercicio 10: Generación de datos históricos y de	
informes . . . . .	363
Ejercicio 11: Utilización de agregados ampliados	
para clases de servicio . . . . .	367

## Capítulo 7. Casos de ejemplo de gestión de carga de trabajo . . . . . 375

Aplicación de ejemplo de gestión de carga de	
trabajo . . . . .	375
Caso de ejemplo: Investigación de una	
ralentización del sistema relacionada con la carga	
de trabajo . . . . .	376
Caso de ejemplo: Agregación de métricas de	
actividad a través de actividades hijas . . . . .	377
Caso de ejemplo: Identificación de actividades que	
tardan demasiado tiempo en completarse . . . . .	380

Caso de ejemplo: Cómo cancelar actividades que han estado en cola durante más de una hora . . .	383
Caso de ejemplo: Identificación de actividades con un coste estimado bajo y un tiempo de ejecución alto. . . . .	386
Caso de ejemplo: Cancelación de todas las actividades que se ejecutan en una subclase de servicio . . . . .	386
Caso de ejemplo: Desconexión de todas las aplicaciones correlacionadas o ejecución de actividades en una clase de servicio . . . . .	387
Caso de ejemplo: Ajuste de una configuración de gestión de carga de trabajo de DB2 cuando hay información de planificación de capacidad disponible . . . . .	389
Caso de ejemplo: Ajuste de una configuración de la gestión de carga de trabajo de DB2 cuando no hay información de planificación de capacidad disponible . . . . .	391
<b>Capítulo 8. Consulta . . . . .</b>	<b>397</b>
Procedimientos y funciones de tabla. . . . .	397
WLM_CANCEL_ACTIVITY - Cancelar una actividad. . . . .	397
WLM_CAPTURE_ACTIVITY_IN_PROGRESS - Recopilar información para supervisor de sucesos de actividades . . . . .	398
WLM_COLLECT_STATS - Coleccionar y restaurar estadísticas de gestión de carga de trabajo. . . . .	400
WLM_GET_ACTIVITY_DETAILS - Devolver información detallada sobre una actividad específica. . . . .	402
Función de tabla WLM_GET_QUEUE_STATS - Devolver estadísticas de cola de umbral . . . . .	408
Función de tabla	
WLM_GET_SERVICE_CLASS_AGENTS - Listar los agentes que se ejecutan en una clase de servicio . . . . .	412
WLM_GET_SERVICE_CLASS	
_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo . . . . .	419
Función de tabla	
WLM_GET_SERVICE_SUBCLASS_STATS - Devolver estadísticas de subclases de servicio . . . . .	423
WLM_GET_SERVICE_SUPERCLASS_STATS - Devolver estadísticas de superclases de servicio . . . . .	430
WLM_GET_WORK_ACTION_SET_STATS - Devolver estadísticas de conjunto de acciones de trabajo. . . . .	432
WLM_GET_WORKLOAD	
_OCCURRENCE - Devolución de una lista de actividades . . . . .	433
Función de tabla	
WLM_GET_WORKLOAD_STATS - Devolver estadísticas de carga de trabajo . . . . .	438
Procedimiento WLM_SET_CLIENT_INFO - Establecer información de cliente. . . . .	441
Elementos de supervisor de gestión de carga de trabajo. . . . .	444

Elemento de supervisor act_cpu_time_top - Tiempo de CPU de actividad máximo . . . . .	444
Elemento de supervisor act_exec_time - Tiempo de ejecución de la actividad . . . . .	444
Elemento de supervisor act_remapped_in - Actividades incluidas en una nueva correlación . . . . .	445
Elemento de supervisor act_remapped_out - Actividades excluidas en una nueva correlación . . . . .	445
Elemento de supervisor act_rows_read_top - Lectura máxima de filas de actividad . . . . .	445
Elemento de supervisor act_throughput - Rendimiento de las actividades . . . . .	446
Elemento de supervisor act_total - Total de actividades . . . . .	447
Elemento de supervisor activate_timestamp - Indicación de la fecha y la hora de activación . . . . .	447
Elemento de supervisor activity_collected - Actividad recopilada . . . . .	448
Elemento de supervisor activity_id - ID de actividad . . . . .	448
Elemento de supervisor activity_secondary_id - ID secundario de actividad. . . . .	449
Elemento de supervisor activity_type - Tipo de actividad . . . . .	450
Elemento de supervisor	
agg_temp_tablespace_top - Espacio de tablas temporal agregado máximo . . . . .	450
Elemento de supervisor arm_correlator - Correlacionador de medición de respuestas de aplicación . . . . .	451
Elemento de supervisor bin_id - Identificador de intervalo de histograma . . . . .	451
Elemento de supervisor bottom - Fin de intervalo de histograma . . . . .	451
Elemento de supervisor concurrent_act_top - Actividad simultánea máxima . . . . .	452
Elemento de supervisor	
concurrent_connection_top - Máximo de conexiones simultáneas . . . . .	452
Elemento de supervisor concurrent_wlo_act_top - Actividad de WLO simultánea máxima . . . . .	453
Elemento de supervisor concurrent_wlo_top - Máximo de instancias de carga de trabajo simultáneas . . . . .	453
Elemento de supervisor	
currentdbcoordactivities_db_threshold_id - ID de umbral de base de datos de actividades simultáneas de coordinador de base de datos. . . . .	454
Elemento de supervisor	
currentdbcoordactivities_db_threshold	
_queued - Umbral de base de datos de actividades simultáneas de coordinador de base de datos en cola . . . . .	454
Elemento de supervisor	
currentdbcoordactivities_db	
threshold_value - Valor de umbral de base de datos de actividades simultáneas de coordinador de base de datos . . . . .	455

Elemento de supervisor concurrentdbcoordactivities_db_ threshold_violated - Umbral violado de base de datos de actividades simultáneas de coordinador de base de datos . . . . .	455
Elemento de supervisor concurrentdbcoordactivities_subclass_ threshold_id - ID de umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos . . . . .	456
Elemento de supervisor concurrentdbcoordactivities_subclass_ threshold_queued - Umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos en cola . . . . .	456
Elemento de supervisor concurrentdbcoordactivities_subclass_ threshold_value - Valor de umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos . . . . .	456
Elemento de supervisor concurrentdbcoordactivities_subclass_ threshold_violated - Umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos violado . . . . .	457
Elemento de supervisor concurrentdbcoordactivities_superclass_ threshold_id - ID de umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos . . . . .	457
Elemento de supervisor concurrentdbcoordactivities_superclass_ threshold_queued - Umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos en cola . . . . .	458
Elemento de supervisor concurrentdbcoordactivities_superclass_ threshold_value - Valor de umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos . . . . .	458
Elemento de supervisor concurrentdbcoordactivities_superclass_ threshold_violated - Umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos violado . . . . .	459
Elemento de supervisor concurrentdbcoordactivities_wl_was_threshold _id - ID de umbral de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos . . . . .	459
Elemento de supervisor concurrentdbcoordactivities_wl_was_threshold _queued - Umbral en cola de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos . . . . .	460

Elemento de supervisor concurrentdbcoordactivities_wl_was_threshold _value - Valor de umbral de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos . . . . .	460
Elemento de supervisor concurrentdbcoordactivities_wl_was_threshold _violated - Umbral violado de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos . . . . .	461
Elemento de supervisor coord_act_aborted_total - Total de actividades de coordinador abortadas .	461
Elemento de supervisor coord_act_completed_total - Total de actividades de coordinador completadas . . . . .	462
Elemento de supervisor coord_act_est_cost_avg - Promedio de coste estimado de actividad de coordinador . . . . .	463
Elemento de supervisor coord_act_exec_time_avg - Promedio de tiempo de ejecución de las actividades de coordinador .	463
Elemento de supervisor coord_act_interarrival_time_avg - Promedio de tiempo de llegada de las actividades de coordinador . . . . .	464
Elemento de supervisor coord_act_lifetime_avg - Promedio de vida útil de las actividades de coordinador . . . . .	465
Elemento de supervisor coord_act_lifetime_top - Vida útil máxima de las actividades de coordinador . . . . .	466
Elemento de supervisor coord_act_queue_time_avg - Promedio de tiempo en cola de las actividades de coordinador . . . . .	467
Elemento de supervisor coord_act_rejected_total - Total de actividades de coordinador rechazadas . . . . .	468
Elemento de supervisor coord_partition_num - Número de partición coordinadora . . . . .	469
Elemento de supervisor cost_estimate_top - Coste estimado máximo . . . . .	470
Elemento de supervisor cpu_limit - Límite de CPU de asignador de WLM . . . . .	470
Elemento de supervisor cpu_share_type - Tipo de cuota de CPU de asignador de WLM . . . . .	470
Elemento de supervisor cpu_shares - Cuotas de CPU de asignador de WLM . . . . .	471
Elemento de supervisor cpu_utilization - Utilización de CPU . . . . .	471
Elemento de supervisor cpu_velocity - Velocidad de CPU . . . . .	472
Elemento de supervisor db_work_action_set_id - ID de conjunto de acciones de trabajo de base de datos . . . . .	473
Elemento de supervisor db_work_class_id - ID de clase de trabajo de base de datos . . . . .	473

Elemento de supervisor	Elemento de supervisor stmt_invocation_id -
destination_service_class_id - de ID de clase de	Identificador de invocación de sentencia . . . . . 492
servicio de destino . . . . . 474	Elemento de supervisor temp_tablespace_top -
Elemento de supervisor	Espacio de tablas temporal máximo . . . . . 493
estimated_cpu_entitlement - Titularidad de CPU	Elemento de supervisor thresh_violations -
estimada . . . . . 474	Número de violaciones de umbral . . . . . 493
Elemento de supervisor histogram_type - Tipo	Elemento de supervisor threshold_action -
de histograma . . . . . 474	Acción de umbral . . . . . 495
Elemento de supervisor last_wlm_reset - Hora	Elemento de supervisor threshold_domain -
del último restablecimiento. . . . . 476	Dominio de umbral . . . . . 495
Elemento de supervisor num_remaps - Número	Elemento de supervisor threshold_maxvalue -
de nuevas correlaciones . . . . . 476	Valor máximo de umbral . . . . . 496
Elemento de supervisor	Elemento de supervisor threshold_name -
num_threshold_violations - Número de	Nombre del umbral . . . . . 497
violaciones de umbral . . . . . 476	Elemento de supervisor threshold_predicate -
Elemento de supervisor number_in_bin -	Predicado de umbral . . . . . 497
Número en intervalo . . . . . 477	Elemento de supervisor threshold_queuesize -
Elemento de supervisor parent_activity_id - ID	Tamaño de cola de umbral . . . . . 498
de la actividad padre. . . . . 477	Elemento de supervisor thresholdid - ID de
Elemento de supervisor parent_uow_id - ID de	umbral . . . . . 498
unidad de trabajo padre. . . . . 478	Elemento de supervisor time_completed - Hora
Elemento de supervisor prep_time - Tiempo de	de finalización . . . . . 499
preparación . . . . . 478	Elemento de supervisor time_created - Hora de
Elemento de supervisor	creación . . . . . 499
queue_assignments_total - Asignaciones totales	Elemento de supervisor time_of_violation -
en cola . . . . . 479	Hora de violación . . . . . 499
Elemento de supervisor queue_size_top -	Elemento de supervisor time_started - Hora de
Tamaño de cola máximo. . . . . 480	inicio . . . . . 500
Elemento de supervisor queue_time_total -	Elemento de supervisor top - Principio de
Tiempo total en cola . . . . . 480	intervalo de histograma . . . . . 500
Elemento de supervisor request_exec_time_avg -	Elemento de supervisor
Promedio de tiempo de ejecución de peticiones . 481	total_disp_run_queue_time - Tiempo total en
Elemento de supervisor routine_id - ID de	cola de ejecución de asignador . . . . . 500
rutina . . . . . 481	Elemento de supervisor uow_completed_total -
Elemento de supervisor rows_fetched - Filas	Total de unidades de trabajo completadas . . . 502
captadas . . . . . 482	uow_comp_status - Estado de finalización de la
Elemento de supervisor rows_modified - Filas	unidad de trabajo . . . . . 503
modificadas . . . . . 482	uow_elapsed_time - Tiempo transcurrido de la
Elemento de supervisor rows_returned - Filas	unidad más reciente de trabajo . . . . . 504
devueltas. . . . . 484	Elemento de supervisor uow_id - ID de unidad
Elemento de supervisor rows_returned_top -	de trabajo . . . . . 505
Máximo de filas reales devueltas . . . . . 486	Elemento de supervisor uow_lifetime_avg -
Elemento de supervisor sc_work_action_set_id -	Promedio de vida útil de la unidad de trabajo . 506
ID de conjunto de acciones de trabajo de clase	Elemento de supervisor uow_lock_wait_time -
de servicio . . . . . 486	Tiempo total que la unidad de trabajo ha
Elemento de supervisor sc_work_class_id - ID	esperado en bloqueos . . . . . 506
de clase de trabajo de clase de servicio . . . . 487	Elemento de supervisor uow_log_space_used -
Elemento de supervisor section_env - Entorno	Espacio utilizado de anotaciones cronológicas de
de sección . . . . . 487	unidad de trabajo . . . . . 507
Elemento de supervisor service_class_id - ID de	Elemento de supervisor uow_start_time -
clase de servicio . . . . . 488	Indicación de fecha y hora de inicio de la
Elemento de supervisor service_subclass_name -	unidad de trabajo . . . . . 507
Nombre de subclase de servicio . . . . . 489	uow_status - Estado de la unidad de trabajo . . 509
Elemento de supervisor	Elemento de supervisor uow_stop_time -
service_superclass_name - Nombre de	Indicación de fecha y hora de detención de
superclase de servicio . . . . . 490	unidad de trabajo . . . . . 509
Elemento de supervisor source_service_class_id	Elemento de supervisor uow_throughput -
- ID de clase de servicio . . . . . 491	Rendimiento de las unidades de trabajo . . . 510
Elemento de supervisor statistics_timestamp -	Elemento de supervisor uow_total_time_top -
Indicación de la fecha y la hora de las	Tiempo total de UOW máximo . . . . . 510
estadísticas . . . . . 491	

Elemento de supervisor wl_work_action_set_id - Identificador de conjunto de acciones de trabajo de carga de trabajo . . . . .	511
Elemento de supervisor wl_work_class_id - Identificador de clase de trabajo de carga de trabajo. . . . .	512
Elemento de supervisor wlm_queue_assignments_total - Asignaciones totales en cola de gestor de carga de trabajo . . . . .	512
Elemento de supervisor wlm_queue_time_total - Tiempo total del gestor de carga de trabajo en cola . . . . .	514
Elemento de supervisor wlo_completed_total - Instancias de carga de trabajo completadas totales. . . . .	515
Elemento de supervisor work_action_set_id - ID de conjunto de acciones de trabajo . . . . .	516
Elemento de supervisor work_action_set_name - Nombre de conjunto de acciones de trabajo . . . . .	516
Elemento de supervisor work_class_id - ID de clase de trabajo. . . . .	517
Elemento de supervisor work_class_name - Nombre de clase de trabajo. . . . .	517
Elemento de supervisor workload_id - ID de la carga de trabajo . . . . .	518
Elemento de supervisor workload_name - Nombre de carga de trabajo . . . . .	519
Elemento de supervisor workload_occurrence_id - Identificador de instancia de carga de trabajo . . . . .	520
Elemento de supervisor workload_occurrence_state - Estado de la instancia de carga de trabajo . . . . .	521
Mandatos . . . . .	522
SET WORKLOAD. . . . .	522
Parámetros de configuración . . . . .	523
wlm_collect_int - Intervalo de recopilación de gestión de carga de trabajo . . . . .	523
wlm_dispatcher - Asignador de gestión de carga de trabajo . . . . .	524
wlm_disp_concur - Simultaneidad de hebras de asignador de gestor de carga de trabajo . . . . .	525
wlm_disp_cpu_shares - Cuotas de CPU de asignador de gestor de carga de trabajo . . . . .	526
wlm_disp_min_util - Utilización de CPU mínima del asignador del gestor de carga de trabajo. . . . .	526

Vistas de catálogo . . . . .	527
SYSCAT.HISTOGRAMTEMPLATEBINS . . . . .	527
SYSCAT.HISTOGRAMTEMPLATES . . . . .	528
SYSCAT.HISTOGRAMTEMPLATEUSE . . . . .	528
SYSCAT.SERVICECLASSES. . . . .	529
SYSCAT.THRESHOLDS . . . . .	532
SYSCAT.WORKACTIONS . . . . .	534
SYSCAT.WORKACTIONSETS . . . . .	537
SYSCAT.WORKCLASSES . . . . .	538
SYSCAT.WORKCLASSETS . . . . .	539
SYSCAT.WORKLOADAUTH . . . . .	539
SYSCAT.WORKLOADCONNATTR . . . . .	540
SYSCAT.WORKLOADS . . . . .	540

**Apéndice A. Normas de denominación  
generales . . . . . 545**

**Apéndice B. Roles . . . . . 547**

**Apéndice C. Contextos fiables y  
conexiones fiables . . . . . 549**

**Apéndice D. Visión general de la  
información técnica de DB2 . . . . . 553**

Biblioteca técnica de DB2 en copia impresa o en formato PDF . . . . .	554
Visualización de la ayuda para estados de SQL desde el procesador de línea de mandatos. . . . .	556
Acceso a diferentes versiones del Centro de información de DB2 . . . . .	556
Actualización del Centro de información de DB2 instalado en el sistema o en el servidor de intranet . . . . .	557
Actualización manual del Centro de información de DB2 instalado en el sistema o en el servidor de intranet . . . . .	558
Guías de aprendizaje de DB2 . . . . .	560
Información de resolución de problemas de DB2 . . . . .	560
Términos y condiciones . . . . .	561

**Apéndice E. Avisos . . . . . 563**

**Índice. . . . . 567**

---

## Acerca de este manual

Este manual proporciona información sobre las características y las funciones de gestión de carga de trabajo DB2 que pueden ayudarle a obtener un entorno de ejecución estable y predecible que cumpla sus objetivos empresariales. Con la gestión de carga de trabajo de DB2, se gestionan tanto peticiones como recursos. En este manual también se proporciona información sobre la supervisión y la resolución de problemas de la carga de trabajo en el servidor de datos.



---

## Capítulo 1. Introducción a los conceptos de la gestión de carga de trabajo de DB2

Un buen sistema de gestión de carga de trabajo ayuda a cumplir con eficiencia los objetivos del entorno en el que se produce el trabajo. Podrá ver ejemplos de la necesidad de disponer de un buen sistema de gestión de carga de trabajo con sólo echar un vistazo a su alrededor.

Por ejemplo, veamos en un supermercado de alimentación. Deben tenerse en cuenta distintas actividades: servicio al cliente, reposición de productos, mantenimiento de inventarios, etc. Y deben definirse algunos objetivos simples. Los propietarios del establecimiento desean maximizar el número de clientes que recorren el supermercado y la cantidad de clientes que realizan compras, y cumplir ambos objetivos de forma que los clientes salgan del local satisfechos y con ganas de volver. Los propietarios del establecimiento también desean asegurarse de que tienen suficientes existencias a disposición de sus clientes (pero no demasiadas, pues el material no vendido es un problema). Los propietarios del establecimiento realizan el seguimiento de las compras de sus clientes y utilizan esta información para crear reclamos publicitarios diseñados para inducirles a volver. Los mecanismos de supervisión realizan el seguimiento del inventario y envían notificaciones cuando las existencias se agotan. Se han instalado dispositivos de seguridad para detectar hurtos. Se han abierto cajeros de compra rápida para que los compradores que sólo desean comprar algunos artículos puedan hacerlo sin tener que esperar detrás de otros clientes que realizan compras de muchos artículos. Si todos estos objetivos se cumplen y todos estos procedimientos operativos funcionan correctamente, los clientes estarán satisfechos, y posiblemente con ganas de volver en lugar de cambiar a otro supermercado. Estos objetivos y procedimientos operativos son todos los aspectos de la gestión de carga de trabajo.

En un entorno de servidor de datos, podrá ver que la necesidad de disponer de una gestión eficiente del trabajo es incluso mayor, en especial ahora que los servidores de datos han de asumir más carga que nunca. Las cajas registradoras generan miles de inserciones de datos, constantemente se generan informes para determinar si se cumplen los objetivos de ventas, se utilizan aplicaciones de proceso por lotes que se ejecutan para cargar los datos recopilados y han de llevarse a cabo tareas de administración, como la realización de copias de seguridad y las reorganizaciones que se ejecutan para proteger los datos y hacer posible que la ejecución del servidor sea óptima. Todas estas operaciones utilizan el mismo sistema de base de datos y compiten por los mismos recursos.

Para garantizar las mejores expectativas de cumplimiento de los objetivos de ejecución de un servidor de datos, es de vital importancia disponer de un sistema de gestión de carga de trabajo eficiente.

---

### Dominios de la gestión de carga de trabajo

La gestión de carga de trabajo tiene tres dominios bien definidos: identificación del trabajo que entra en el servidor de datos, gestión del trabajo cuando éste se ejecuta y supervisión para garantizar la utilización eficiente del servidor de datos.

Para obtener una correcta gestión de carga de trabajo con el gestor de carga de trabajo de DB2 existen determinados aspectos que deben considerarse, empezando

por la definición de los objetivos. En el ejemplo del supermercado de alimentación que se describe en Capítulo 1, “Introducción a los conceptos de la gestión de carga de trabajo de DB2”, en la página 1, los objetivos podrían incluir la maximización del gasto que realizan los clientes, la minimización de los hurtos y la garantía de que los clientes abandonan el supermercado satisfechos y que volverán.

En un entorno de servidor de datos, también debe definir objetivos. A veces los objetivos son claros, en especial cuando se originan desde los objetivos del acuerdo de nivel de servicio (SLA). Por ejemplo, las consultas de una aplicación en particular puede que no consuman más del 10% de los recursos totales de procesador. Los objetivos también deben asociarse a una hora del día en particular. Por ejemplo, un programa de utilidad de proceso por lotes que se ejecuta en horario nocturno podría haber completado la carga de datos a las 08:00 horas aproximadamente para que los informes de ventas diarios puedan generarse puntualmente. En otras situaciones, los objetivos pueden ser de difícil cuantificación. Un objetivo podría ser mantener la satisfacción de los usuarios de la base de datos y evitar que se acumule una actividad de base de datos de tal envergadura que llegue a obstaculizar su trabajo diario. Tanto si los objetivos pueden cuantificarse como si no, entender éstos es de vital importancia al considerar las etapas de la gestión de carga de trabajo que se indican a continuación:

### **Identificación**

Si desea lograr un objetivo para un trabajo, primero debe ser capaz de identificar los detalles relacionados con el trabajo. En el supermercado de alimentación, puede identificar la información del comprador por medio de las tarjetas de crédito y de las tarjetas de débito, o un artículo que no ha pasado por caja mediante la utilización de una etiqueta de seguridad activa adherida al artículo. En el caso del servidor de datos, debe decidir cómo desea identificar el trabajo que entra en el sistema. Puede utilizar el nombre de la aplicación que envía el trabajo, el ID de autorización que envía el trabajo o una combinación de elementos que proporcionen alguna forma de identificación.

### **Gestión**

La etapa de gestión incluye los mecanismos que han de hacer posible el progreso estable hacia los objetivos y las acciones que han de realizarse si no se cumple un objetivo. Un ejemplo de mecanismo es la gestión de la comprobación de precios en una caja rápida del supermercado. Las cajas rápidas deben ofrecer rapidez y agilidad y garantizar la satisfacción de los clientes, y si un envase de leche tiene un precio incorrecto y es necesario comprobar el precio, la caja rápida puede dejar de ser ágil. El problema se gestiona realizando una comprobación rápida del precio, posiblemente abriendo otra caja, e intentando solucionar el problema del precio de modo que no vuelva a producirse. En el servidor de datos, puede que observe que el rendimiento general desciende cuando se ejecutan sentencias de SQL que se han escrito sin suficiente información, que se produce un pico en el volumen durante las horas de máxima actividad o que existe demasiada competición por los mismos recursos entre distintas aplicaciones. La etapa de gestión incluye los mecanismos que han de hacer posible la asignación de recursos para lograr los objetivos y las acciones que han de realizarse si no se cumple un objetivo. Estos mecanismos de gestión de carga de trabajo, que indirecta o directamente controlan los recursos de CPU, incluyen lo siguiente:

1. Umbrales de simultaneidad, aplicados con un conjunto de acciones de trabajo definido en la carga de trabajo, para controlar la simultaneidad del trabajo de entrada

2. La capacidad de trasladar un trabajo de una clase de servicio a otra actualmente sólo resulta eficaz cuando el asignador de gestión de carga de trabajo se utiliza en esas clases de servicio
3. Asignador de gestión de carga de trabajo para asignar recursos de CPU de forma específica para que las cargas de trabajo asignadas a las clases de servicio puedan ejercer un control más granular de los recursos de CPU cuando los primeros mecanismos de gestión de carga de trabajo no son adecuados

### Supervisión

La supervisión es importante por un par de razones. En primer lugar, para determinar si se logra un objetivo, debe disponer de un mecanismo que realice el seguimiento del progreso hacia ese objetivo. Asimismo, la supervisión ayuda a identificar los problemas que podrían impedir la obtención del objetivo. En un supermercado, el encargado puede observar el flujo de clientes, recibir alertas automáticamente relacionadas con problemas como los hurtos o un descenso de inventario de un producto en particular que merece inmediata atención o realizar análisis basados en patrones de compra históricos para determinar la ubicación óptima del producto en el supermercado. Respecto al servidor de datos, con frecuencia existen objetivos explícitos de tiempo de respuesta de las actividades de la base de datos y es importante disponer de un mecanismo que permita obtener estas métricas, y observar las tendencias.

En la siguiente figura se representan las fases de la gestión de carga de trabajo:

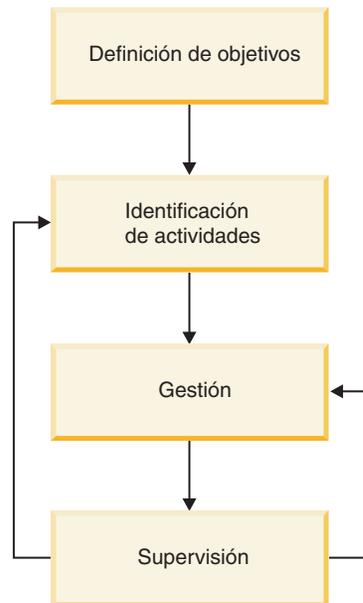


Figura 1. Fases de la gestión de carga de trabajo

---

## Autorización (WLMADM) de administrador de gestión de carga de trabajo

Para poder gestionar los objetos de la carga de trabajo de una base de datos específica, necesita la autorización (WLMADM) de administrador de gestión de carga de trabajo. Esta autorización le permite crear, modificar, descartar, comentar, otorgar acceso a, y revocar acceso de, objetos de gestión de carga de trabajo de DB2.

Los objetos de gestión de carga de trabajo son objetos del sistema similares a las agrupaciones de almacenamientos intermedios y a los espacios de tablas. Como tales, no tienen propietarios asociados.

El administrador de seguridad, que posee la autorización SECADM, o un usuario con autorización ACCESSCTRL, pueden otorgar la autoridad WLMADM a un usuario, grupo o rol.

La autorización WLMADM proporciona la capacidad de llevar a cabo las tareas siguientes:

- Emitir las sentencias CREATE, ALTER, COMMENT ON y DROP para los objetos de gestión de carga de trabajo de DB2 siguientes:
  - Plantillas de histograma
  - Clases de servicio
  - Umbrales
  - Conjuntos de acciones de trabajo
  - Conjuntos de clases de trabajo
  - Cargas de trabajo
- Emitir las sentencias GRANT y REVOKE para privilegios de carga de trabajo

Si cuenta con autorización de administrador de bases de datos (DBADM), también tiene implícita la autorización de WLMADM.

---

## Preguntas frecuentes sobre la gestión de carga de trabajo de DB2

En estas Preguntas frecuentes se proporcionan respuestas a las preguntas más comunes relacionadas con la gestión de carga de trabajo de DB2.

### General

- ¿En qué plataformas de DB2 puedo utilizar la gestión de carga de trabajo de DB2?
- No estoy en AIX. ¿Significa esto que no tengo ningún control sobre los recursos de procesador o la actividad de E/S?
- Ahora que Query Patroller se ha dejado de mantener y DB2 Governor ha quedado en desuso, ¿cómo puedo migrar al gestor de carga de trabajo de DB2?
- ¿Existe alguna forma de que WebSphere Application Server pase los campos de información de cliente que la carga de trabajo de DB2 utiliza?
- ¿Por qué mi trabajo no se asigna a la carga de trabajo correcta?
- ¿Por qué el gestor de carga de trabajo de DB2 afecta a REORGCHK, IMPORT, EXPORT y a otros mandatos del CLP?
- ¿Existe alguna forma de cambiar la clase de servicio a la que se ha asignado un trabajo mientras éste se ejecuta?
- La mayor parte de mi trabajo por lotes se realiza utilizando scripts de CLP con el mismo ID ¿cómo puedo identificar de forma exclusiva éstos para poder gestionarlos diferenciándolos?
- ¿Cuándo debería utilizar la cláusula COLLECT AGGREGATE ACTIVITY DATA frente a la cláusula COLLECT ACTIVITY DATA?
- ¿Cómo funciona la gestión de carga de trabajo de DB2 con la nueva función de WPAR de AIX?

- ¿Qué relación existe entre la variable de registro DB2\_OPT\_MAX\_TEMP\_SIZE y un umbral de DB2 basado en SQLTEMPSPACE?

#### **Licencias**

- ¿Cuáles son los requisitos de licencia para el gestor de carga de trabajo de DB2?

#### **Supervisión**

- ¿Qué información se obtiene de los distintos supervisores de sucesos que están asociados con la gestión de carga de trabajo?

#### **Gestión de carga de trabajo del sistema operativo (AIX WLM y Linux WLM)**

- ¿Por qué podría llegar a desear utilizar AIX WLM o Linux WLM?
- No estoy en AIX. ¿Significa esto que no tengo ningún control sobre los recursos de procesador o la actividad de E/S?
- ¿Puedo utilizar AIX WLM para gestionar la actividad de E/S?
- ¿Puedo utilizar AIX WLM para gestionar el uso de memoria?
- ¿Cómo funciona DB2 WLM con la nueva función de WPAR de AIX?

#### **Plataformas**

- ¿En qué plataformas de DB2 puedo utilizar el gestor de carga de trabajo de DB2?
- No estoy en AIX. ¿Significa esto que no tengo ningún control sobre los recursos de procesador o la actividad de E/S?
- ¿Cómo funciona la gestión de carga de trabajo de DB2 con la nueva función de WPAR de AIX?
- ¿Por qué podría llegar a desear utilizar AIX WLM o Linux WLM?

#### **Query Patroller y Governor**

- ¿Cómo afectan estas nuevas funciones a Query Patroller y a DB2 Governor?
- Ahora que Query Patroller y DB2 Governor están en desuso ¿cómo puedo migrar al gestor de carga de trabajo de DB2?

#### **Umbrales**

- ¿Puedo crear varios umbrales de simultaneidad CONCURRENTDBCOORDACTIVITIES para el mismo conjunto de trabajo?
- ¿Cómo puedo determinar las actividades que un umbral de gestión de carga de trabajo coloca en cola y el orden de las actividades de la cola?

#### **Asignador de gestión de carga de trabajo**

- ¿Necesito utilizar el asignador de gestión de carga de trabajo?
- ¿Qué cambios en el comportamiento podría ver al activar el asignador de gestión de carga de trabajo?
- Con la introducción del asignador de gestión de carga de trabajo ¿dejarán de necesitarse o de ser útiles umbrales de simultaneidad como CONCURRENTDBCOORDACTIVITIES?

### **¿En qué plataformas de DB2 puedo utilizar la gestión de carga de trabajo de DB2?**

La gestión de carga de trabajo de DB2 está disponible en todas las plataformas que reciben el soporte de DB2 9.5 para Linux, UNIX y Windows o posterior. La firme

integración opcional (que se ofrece entre la gestión de carga de trabajo de DB2 en el nivel de clase de servicio y las posibilidades de gestión de carga de trabajo del sistema operativo), está disponible en las plataformas AIX y en cualquier plataforma Linux basada en el kernel 2.6.26 o superior.

## ¿Necesito utilizar el asignador de gestión de carga de trabajo?

La mayoría de las configuraciones de gestión de carga de trabajo empiezan con umbrales de simultaneidad, que afectan al consumo de todos los recursos a través del control de la cantidad de trabajo que puede empezar a ejecutarse en un momento dado. En algunas situaciones, sin embargo, un umbral de simultaneidad no tiene la capacidad de limitar con eficacia la cantidad total de recursos de proceso que se consumen, y el trabajo de alta prioridad se ve afectado; por ejemplo, cuando se restringe un trabajo complejo a la ejecución de una sola consulta que, a pesar de todo, consume los suficientes recursos como para interrumpir otros trabajos de mayor prioridad. En circunstancias como ésta, el asignador de gestión de carga de trabajo se utiliza para controlar explícitamente el consumo de CPU y para proteger el trabajo de mayor prioridad.

Puede utilizar el asignador de gestión de carga de trabajo para todas o cualquiera de las situaciones siguientes:

- Desea gestionar la cuota de recursos de CPU entre varios usuarios o aplicaciones y utiliza un sistema operativo que no dispone de un gestor de carga de trabajo de sistema operativo (SO) que se integre con la gestión de carga de trabajo de DB2 por medio del campo `outbound_correlator` de cada clase de servicio.
- Desea gestionar la cuota de recursos de CPU entre varios usuarios o aplicaciones y no dispone de privilegio de usuario root en el sistema operativo.
- Desea gestionar la cuota de recursos de CPU entre varios usuarios o aplicaciones en un entorno con varios miembros en varios sistemas y la gestión de ello por medio del WLM de SO en cada sistema requiere demasiada administración.
- Desea gestionar la cuota de recursos de CPU entre varios usuarios o aplicaciones utilizando cuotas fijas para limitar determinadas clases de servicio, aún cuando la CPU se utiliza por debajo de sus posibilidades, y esto no está disponible en su WLM de SO y no produce el resultado deseado.

## ¿Cómo afectan estas nuevas funciones a Query Patroller y a DB2 Governor?

El gestor de carga de trabajo de DB2 incorpora un enfoque independiente para la gestión de carga de trabajo y no depende de ni interactúa con Query Patroller o DB2 Governor de ningún modo. Query Patroller se ha dejado de mantener a partir del release de Versión 10.1. DB2 Governor ha empezado a estar en desuso en el release de DB2 Versión 9.7 y, aunque sigue siendo posible utilizar sus funciones, ha dejado de ser un elemento central de la estrategia de gestión de carga de trabajo de DB2. No se ha planificado realizar ninguna inversión adicional para DB2 Governor en futuros releases.

Cuando DB2 9.5 o posterior se instala por primera vez, la clase de servicio de usuario por omisión se define automáticamente y todo el trabajo de entrada se envía a ésta para su ejecución. Aunque DB2 Governor puede observar los agentes de cualquier clase de servicio, está permitido ajustar la prioridad de agente sólo de los agentes de la clase de servicio de usuario por omisión.

## **No estoy en AIX. ¿Significa esto que no tengo ningún control sobre los recursos de procesador o la actividad de E/S?**

Los usuarios de todas las plataformas también pueden controlar los recursos de procesador y la actividad de E/S entre las clases de servicio utilizando SQL, como las sentencias CREATE y ALTER SERVICE CLASS, por ejemplo.

Para controlar el uso de CPU cuando se ha habilitado el asignador de gestión de carga de trabajo, utilice el atributo de límite de CPU de la clase de servicio de DB2 para limitar la cantidad de recursos de CPU que una clase de servicio puede consumir. Si también se ha habilitado el parámetro de configuración de gestión de bases de datos (**wlm\_disp\_cpu\_shares**) de las cuotas de CPU del asignador de gestión de carga de trabajo, puede utilizar el atributo de cuotas de CPU de la clase de servicio de DB2 para especificar la cuota de recursos de CPU que puede consumir una clase de servicio en relación con el consumo de CPU de otras clases de servicio. En AIX y algunas plataformas Linux, puede proporcionar (o sustituir) estas soluciones aprovechando las capacidades de gestión de carga de trabajo que ofrecen dichos sistemas operativos para controlar el consumo de CPU.

Para la actividad de E/S, los usuarios de todas las plataformas pueden establecer el atributo de prioridad de agrupación de almacenamientos intermedios o de captador previo de una clase de servicio de DB2 en el valor alto, medio o bajo. Por omisión, todas las clases de servicio se ejecutan con una prioridad media.

## **¿Puedo utilizar AIX o Linux WLM o el asignador de gestión de carga de trabajo de DB2 para gestionar la actividad de E/S?**

Actualmente, ni AIX WLM ni Linux WLM dan soporte a los controles de actividad de E/S directa en las hebras. No obstante, es posible controlar de manera indirecta la actividad de E/S mediante umbrales de simultaneidad, o utilizar el asignador de gestión de carga de trabajo de DB2, AIX WLM o Linux WLM para manipular los recursos de CPU. Cuantos más recursos de CPU estén disponibles para una hebra que se está ejecutando, con menos frecuencia solicitará dicha hebra recursos de E/S.

Puede condicionar el comportamiento de la agrupación de almacenamientos intermedios utilizando el atributo BUFFERPOOL PRIORITY de cualquier clase de servicio de DB2. También puede controlar la actividad de E/S del captador previo de DB2 utilizando el atributo PREFETCH PRIORITY de cualquier clase de servicio de DB2.

## **¿Puedo utilizar AIX o Linux WLM para gestionar el uso de la memoria?**

El servidor de datos de DB2 utiliza principalmente memoria compartida a la que acceden más de un agente de distintas clases de servicio. Por este motivo, no es posible dividir la asignación de memoria entre distintas clases de servicio utilizando AIX o Linux WLM.

La memoria (por ejemplo, sortheap) que se consume durante la ejecución de una sentencia de SQL puede estar indirectamente influida por el uso de umbrales de simultaneidad, porque el consumo no comienza hasta que la sentencia tiene permiso para ejecutarse. No obstante, y a diferencia de la actividad de E/S, el hecho de restringir el consumo de CPU no afecta a la cantidad de memoria que se consume. De hecho, restringir el consumo de CPU puede exacerbar la condición de

la memoria, porque las consultas se ejecutarán con más lentitud y permanecerán en la memoria asignada por más tiempo.

### **¿Existe alguna forma de que WebSphere Application Server pase los campos de información de cliente que la carga de trabajo de DB2 utiliza?**

La Versión 6.0 y la Versión 6.1 de WebSphere Application Server pueden establecer o pasar la información de cliente de los campos CLIENT INFO al servidor de datos de DB2, explícitamente mediante las aplicaciones (consulte Paso de información de cliente a una base de datos) o implícitamente estableciendo que WebSphere Application Server realice la acción (consulte Información del cliente establecida implícitamente).

### **¿Puedo crear varios umbrales de simultaneidad CONCURRENTDBCOORDACTIVITIES para el mismo conjunto de trabajo?**

Puede crear uno o varios umbrales de simultaneidad CONCURRENTDBCOORDACTIVITIES que se apliquen al mismo conjunto de actividades definiendo éstos en el nivel de la base de datos, la clase de servicio en la que se ejecuta el trabajo o en un conjunto de acciones de trabajo que se aplique en el nivel de la base de datos o de la carga de trabajo. Tenga en cuenta que cada nuevo umbral de simultaneidad que se aplique a una actividad implica la actividad general adicional que conlleva la imposición de ese umbral de simultaneidad. Compruebe si realmente necesita más de un nivel de umbral de simultaneidad.

### **¿Por qué mi trabajo no se asigna a la carga de trabajo correcta?**

Existen diversas razones por las que una conexión podría no correlacionarse con la carga de trabajo deseada. Las más comunes son no haber otorgado el privilegio USAGE en la carga de trabajo, haber escrito incorrectamente los atributos de conexión sensibles a las mayúsculas y minúsculas o la existencia de una definición de carga de trabajo coincidente que ocupa una posición anterior en el orden de evaluación.

Para que una conexión pueda asignarse a una carga de trabajo, primero los atributos de la conexión deben coincidir con los de la definición de la carga de trabajo, y el ID de autorización de sesión debe tener el privilegio USAGE en la carga de trabajo. Una omisión común es crear la carga de trabajo pero no otorgar a los usuarios el privilegio USAGE en la carga de trabajo (consulte "Sentencia GRANT (privilegios de carga de trabajo)" en *Consulta de SQL*). Sólo los usuarios que disponen de autorización ACCESSCTRL, SECADM o WLMADM pueden otorgar el privilegio de uso de carga de trabajo a otros usuarios. Los usuarios que tienen autorización ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM disponen de privilegio de uso implícito para todas las cargas de trabajo.

Los atributos de conexión para las cargas de trabajo son sensibles a las mayúsculas y minúsculas. Por ejemplo, si el ID de usuario del sistema se ha especificado en mayúsculas, el atributo de conexión SYSTEM\_USER que especifique también ha de indicarse en mayúsculas.

Para determinar la razón por la que una conexión no se correlaciona con la carga de trabajo esperada, debe recopilar alguna información. ¿Con qué carga de trabajo

se correlaciona el trabajo? ¿Se encuentra esa carga de trabajo antes o después de la que cree que habría de utilizarse cuando consulta las definiciones de carga de trabajo en el orden de evaluación? (Sugerencia: intente seleccionar las definiciones de carga de trabajo clasificadas en orden ascendente por el valor de la columna EVALUATIONORDER en SYSCAT.WORKLOADS).

Si desconoce los atributos de conexión de la conexión de destino, puede averiguar los valores de la conexión de varias formas distintas:

- Emita una consulta para el sistema utilizando la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES mientras la conexión está activa
- Abra un cursor en una conexión y utilice el procedimiento almacenado WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS para ese cursor para que la información de actividad se capture para el supervisor de sucesos de actividades. (Sugerencia: no olvide crear y activar el supervisor de sucesos de información de actividades).
- Active la recopilación de información de actividad detallada para la carga de trabajo que la conexión utiliza, emita una sentencia con el fin de capturar la información de actividad y desactive la recopilación.

### **¿Por qué el gestor de carga de trabajo de DB2 afecta a REORGCHK, IMPORT, EXPORT y a otros mandatos del CLP?**

Los umbrales de gestión de carga de trabajo de DB2 afectan a estos mandatos del CLP porque el motor de base de datos no puede distinguir entre las peticiones del sistema que se originan con estos programas de utilidad y las demás peticiones que inician directamente los usuarios del componente frontal interactivo del CLP.

### **¿Existe alguna forma de cambiar la clase de servicio a la que se ha asignado un trabajo mientras éste se ejecuta?**

Sí, puede cambiar la subclase de servicio en la que se ejecuta una actividad por otra subclase de servicio de la misma superclase de servicio padre definiendo un umbral CPUTIMEINSC, DATATAGINSC o SQLROWSINSC con la acción REMAP ACTIVITY en la subclase de servicio original. Inicialmente, la gestión de carga de trabajo de DB2 correlaciona una actividad con una clase de servicio basándose en la definición de carga de trabajo relacionada para la conexión, la modifica según procede si existe un conjunto de acciones de trabajo en esa clase de servicio y, a continuación, establece el agente de DB2 para que se ejecute en la clase de servicio asignada. Cuando una actividad infringe un umbral que tiene definida una acción REMAP ACTIVITY, el agente vuelve a correlacionarse automáticamente con la subclase de servicio de destino especificada (en la misma superclase) cuando se ha detectado la violación de umbral y la actividad continúa su ejecución en la nueva subclase de servicio.

### **La mayor parte de mi trabajo por lotes se realiza utilizando scripts de CLP con el mismo ID ¿cómo puedo identificar de forma exclusiva éstos para poder gestionarlos diferenciándolos?**

Dispone de dos opciones:

Se ha añadido una mejora al CLP para que el nombre de la aplicación cliente se establezca automáticamente en el nombre de archivo del script de CLP, con un prefijo **CLP** delante de éste (el valor de este campo en el servidor puede verse en el registro especial CURRENT CLIENT\_APPLNAME). Por ejemplo, si el nombre de

archivo del script de CLP es **batch.db2**, el CLP establece el valor del registro especial CURRENT CLIENT\_APPLNAME en **CLP batch.db2** cuando se ejecuta ese script. Con esta función, es posible asociar scripts de CLP distintos a las cargas de trabajo basándose en el nombre de la aplicación cliente.

Por ejemplo, para crear una carga de trabajo para el archivo de CLP batch1.db2, puede emitir la sentencia de DDL siguiente:

```
CREATE WORKLOAD batch1 CURRENT CLIENT_APPLNAME ('CLP batch1.db2')
SERVICE CLASS class1
```

Para crear una carga de trabajo para el archivo de CLP **batch2.db2**, puede emitir la sentencia de DDL siguiente:

```
CREATE WORKLOAD batch2 CURRENT CLIENT_APPLNAME ('CLP batch2.db2')
SERVICE CLASS class2
```

Puesto que estos dos archivos por lotes se asocian a distintas cargas de trabajo, pueden asignarse a distintas clases de servicio y pueden gestionarse diferenciándolos.

Otra opción es el nuevo procedimiento almacenado WLM\_SET\_CLIENT\_INFO, que le permite establecer los valores de cualquier campo de información de cliente en el servidor utilizando una simple sentencia CALL de SQL. Mediante la inserción de una sentencia CALL en cualquiera de los scripts de CLP existentes puede identificar éstos de forma exclusiva utilizando estos campos y correlacionarlos con definiciones de carga de trabajo distintas.

Para obtener más información, consulte “Procedimiento WLM\_SET\_CLIENT\_INFO” en *Rutinas y vistas administrativas*.

## **¿Cuándo debería utilizar la cláusula COLLECT AGGREGATE ACTIVITY DATA frente a la cláusula COLLECT ACTIVITY DATA?**

La respuesta depende de por qué se desea la supervisión y qué ha de hacerse con la información.

La información de actividad agregada se expande a todo el conjunto de trabajo que se ha ejecutado en el ámbito que cubre la cláusula, y captura las características de resumen de este conjunto; no captura detalles específicos acerca de las actividades individuales. La cláusula COLLECT AGGREGATE ACTIVITY DATA puede especificarse en cargas de trabajo de DB2, clases de servicio de DB2 y conjuntos de acciones de trabajo de DB2. Para la supervisión del funcionamiento normal, utilice la cláusula COLLECT AGGREGATE ACTIVITY DATA porque es muy ligera, un supervisor de sucesos puede obtenerla automáticamente para un registro histórico y proporciona importante información acerca de los patrones de tiempo de respuesta generales. Si se necesita información más detallada relacionada con un tipo de trabajo específico, utilice las acciones COUNT ACTIVITY o COLLECT AGGREGATE ACTIVITY DATA dentro de un conjunto de acciones de trabajo de DB2 para recopilar información más granular (con mínima actividad general) acerca de los distintos tipos de trabajo que se ejecutan en una carga de trabajo, una clase de trabajo o una base de datos.

Por su parte, la información de actividad contiene información detallada acerca de cada una de las actividades que se ejecutan en el ámbito que la sentencia COLLECT ACTIVITY DATA abarca. Esta cláusula puede especificarse en las cargas de trabajo de DB2, las clases de servicio de DB2, los conjuntos de acciones de trabajo de DB2 y los umbrales de DB2. Permite obtener un análisis más detallado

de las actividades individuales que se capturan, para entender el flujo y el tipo de sentencias de SQL que una nueva aplicación envía, por ejemplo, o para investigar oportunidades de ajuste del rendimiento con herramientas como el recurso Explain o el Asesor de diseño. Puesto que captura mucha más información para cada actividad a la que afecta, el impacto de la utilización de esta cláusula es más alto en las actividades afectadas que el de otros métodos de supervisión y debe controlarse cuidadosamente.

## **¿Cómo funciona la gestión de carga de trabajo de DB2 con la nueva función de WPAR de AIX?**

Todos los aspectos de la gestión de carga de trabajo de DB2 funcionarán dentro de una WPAR de AIX porque las WPAR de AIX no dan soporte a la utilización de funciones de AIX WLM; la opción de integrar estrechamente las clases de servicio de DB2 con las clases de servicio de AIX WLM no ofrece ningún beneficio en este entorno.

## **¿Qué relación existe entre la variable de registro DB2\_OPT\_MAX\_TEMP\_SIZE y los umbrales de DB2 basados en SQLTEMPSPACE?**

No existe ninguna relación directa entre estas dos cosas. La variable de registro **DB2\_OPT\_MAX\_TEMP\_SIZE** es una directiva para que el compilador de consultas limite la cantidad de espacio de tablas temporal que una consulta puede utilizar. Esto puede dar lugar a que el optimizador seleccione un plan más caro (potencialmente menos eficiente) pero que utilice menos espacio en los espacios de tablas temporales del sistema. Un umbral de DB2 basado en **SQLTEMPSPACE** no afecta al tipo de plan que el optimizador selecciona. Simplemente da lugar a que el servidor de datos de DB2 supervise el uso de espacio de tablas temporal del sistema mediante esa consulta en cada miembro de base de datos y genere una violación de umbral si se ha excedido el límite especificado durante el proceso normal.

## **Ahora que Query Patroller se ha dejado de mantener y DB2 Governor ha quedado en desuso, ¿cómo puedo migrar al gestor de carga de trabajo de DB2?**

Tras la presentación del gestor de carga de trabajo de DB2 como solución de gestión de carga de trabajo estratégica de DB2 Versión 9.5, Query Patroller se ha dejado de mantener en el release de Versión 10.1 y DB2 Governor está en desuso desde el release de DB2 Versión 9.7, y podría eliminarse en un futuro release.

Aunque DB2 Governor sigue recibiendo soporte en este release, debería empezar a adoptar las nuevas funciones y posibilidades del gestor de carga de trabajo de DB2, incluidas las que se presentan en este release. Tenga en cuenta que con el gestor de carga de trabajo de DB2 dispone de muchas más opciones, y debería explorarlas, que podrían requerir que se replanteara el enfoque para controlar el trabajo en el servidor de datos de DB2 en términos de gestión de la carga de trabajo actual. El artículo sobre recomendaciones de DB2, *Implementing DB2 workload management in a data warehouse*, contiene un suplemento especialmente diseñado para aquellos que realizan una migración desde Query Patroller. Los temas de tareas pertinentes también están disponibles en la sección *Tareas relacionadas*.

Para facilitar la migración desde DB2 Query Patroller hasta el gestor de carga de trabajo de DB2, se ha empezado a incluir un script de ejemplo (`qpwlmmig.pl`) a

partir de DB2 V9.7 Fixpack 1. Para obtener información adicional, consulte una de las siguientes tareas para obtener detalles acerca de la migración desde Query Patroller hasta el gestor de carga de trabajo de DB2:

- Migración de Query Patroller al gestor de carga de trabajo de DB2 utilizando el script de ejemplo
- Migración de Query Patroller al gestor de carga de trabajo de DB2

## **¿Cuáles son los requisitos de licencia para el gestor de carga de trabajo de DB2?**

El uso de un subconjunto de capacidades de gestión de carga de trabajo en el servidor de datos de DB2 está restringido mediante licencia. Este subconjunto con licencia se conoce como gestor de carga de trabajo de DB2, y controla la creación de cualquier clase de servicio, carga de trabajo, umbral o conjunto de acciones de trabajo. El acceso a este subconjunto de prestaciones de gestión de carga de trabajo exige uno de los siguientes productos bajo licencia:

- DB2 Enterprise Server Edition para Linux, UNIX y Windows
- DB2 Advanced Enterprise Server Edition para Linux, UNIX y Windows
- Database Enterprise Developer Edition para Linux, UNIX y Windows
- IBM® InfoSphere Warehouse, todas las ediciones
- IBM Smart Analytics System

La licencia no restringe las siguientes funciones de gestión de carga de trabajo:

- Utilizando o modificando las clases de servicio y las cargas de trabajo por omisión; esto incluye todas las posibilidades de supervisión
- Creando, modificando o descartando plantillas de histograma
- Utilizando los procedimientos almacenados o las funciones de tabla de gestión de carga de trabajo de DB2
- Creando, activando, deteniendo o descartando supervisores de sucesos de gestión de carga de trabajo
- Otorgando, modificando o revocando privilegios de carga de trabajo

## **¿Qué información se obtiene de los diferentes supervisores de sucesos que están asociados con la gestión de carga de trabajo?**

Los supervisores de sucesos de violaciones de umbral, estadísticas y actividades capturan información acerca de las violaciones de umbral, estadísticas operativas y datos de actividad agregada e individual (consulte: “Supervisión histórica con supervisores de sucesos de WLM” en la página 256).

Cada supervisor de sucesos recopila uno o más grupos de datos lógicos (consulte el tema acerca de las correlaciones de tipos de sucesos con grupos de datos lógicos, “Event type mappings to logical data groups” en *Database Monitoring Guide and Reference*) y existen uno o varios elementos de supervisor en cada grupo de datos lógicos (consulte el tema acerca de los grupos de datos lógicos del supervisor de sucesos y elementos de supervisor, “Event monitor logical data groups and monitor elements” en *Database Monitoring Guide and Reference*).

Por ejemplo, para descubrir qué información ha recopilado el supervisor de sucesos de violaciones de umbral, empiece por consultar la Tabla 3 del tema acerca de las correlaciones de tipos de sucesos con grupos de datos lógicos (“Event type mappings to logical data groups”). En esta tabla se muestra que el supervisor de

sucesos de violaciones de umbral recopila información en un único grupo de datos lógicos denominado `event_thresholdviolations` (tenga en cuenta que algunos supervisores de sucesos, como el supervisor de sucesos de actividad, recopilan información en varios grupos de datos lógicos). A continuación, busque el grupo de datos lógicos `event_thresholdviolations` en el tema acerca de los grupos de datos lógicos del supervisor de sucesos y elementos de supervisor (“Event monitor logical data groups and monitor elements”). En este tema se muestra de qué elementos de supervisor se informa en el grupo de datos lógicos `event_thresholdviolations`, que incluye lo siguiente:

- **activate\_timestamp**: Indicación de fecha y hora de activación
- **activity\_collected**: Actividad recopilada
- **activity\_id**: ID de actividad
- **agent\_id**: Descriptor de contexto de aplicación (ID de agente)
- **appl\_id**: ID de aplicación
- **coord\_partition\_num**: Número de partición coordinadora
- **destination\_service\_class\_id**: ID de clase de servicio de destino
- **source\_service\_class\_id**: ID de clase de servicio de origen
- **threshold\_action**: Acción de umbral
- **threshold\_maxvalue**: Valor máximo de umbral
- **threshold\_predicate**: Predicado de umbral
- **threshold\_queuesize**: Tamaño de cola de umbral
- **thresholdid**: ID de umbral
- **time\_of\_violation**: Hora de violación
- **uow\_id**: ID de unidad de trabajo

El enfoque descrito brevemente en este ejemplo puede utilizarse para descubrir qué datos recopila cada supervisor de sucesos.

### ¿Cómo puedo determinar las actividades que un umbral de gestión de carga de trabajo coloca en cola y el orden de las actividades de la cola?

Para ello, primero ha de crear una vista utilizando la función de tabla `WLM_GET_SERVICE_CLASS_AGENTS` y, a continuación, ejecutar sentencias para obtener una lista de las actividades en cola ordenadas por hora de entrada en cola. Para ver ejemplos en los que se describe cómo hacerlo, consulte “Ejemplo: Determinación de las actividades que un umbral de WLM coloca en cola y su orden en la cola” en la página 254.

### ¿Qué cambios en el comportamiento podría ver al activar el asignador de gestión de carga de trabajo?

Cuando activa el asignador de gestión de carga de trabajo por medio del parámetro de configuración del gestor de bases de datos `wlm_dispatcher` y ha estado utilizando la prioridad de agente para establecer la prioridad del trabajo de modo que una clase de servicio tenga más prioridad que otra, esta prioridad de agente no puede utilizarse mientras el asignador de gestión de carga de trabajo está habilitado. Como resultado de ello, todas las clases de servicio se tratarán como si tuvieran la prioridad de agente por omisión.

Si habilita las cuotas de CPU por medio del parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares` y no especifica cuotas de CPU o límites de CPU para las clases de servicio, todas las clases de servicio recibirán la misma

cuota flexible de los recursos de CPU del sistema. El efecto de que todas las clases de servicio reciban la misma cuota flexible de recursos de CPU podría dar como resultado una asignación de recursos de CPU a las clases de servicio diferente respecto al release anterior de DB2. Como resultado de esto, debería considerar la posibilidad de establecer cuotas de CPU o valores de límite de CPU adecuados para la carga de trabajo. Para obtener más información acerca de cómo determinar los valores para las cuotas de CPU y los límites de CPU, consulte: “Asignador de gestión de carga de trabajo” en la página 177.

### **Con la introducción del asignador de gestión de carga de trabajo ¿dejarán de necesitarse o de ser útiles umbrales de simultaneidad como CONCURRENTDBCOORDACTIVITIES?**

El asignador de gestión de carga de trabajo de DB2 y los umbrales de simultaneidad pueden utilizarse conjuntamente. Los umbrales de simultaneidad siguen siendo muy útiles para controlar la cantidad de trabajo que se ejecuta. Para cada actividad que empieza a ejecutarse, el gestor de bases de datos DB2 proporciona otros recursos a esa actividad, además de recursos de CPU, que la actividad suele conservar mientras está en ejecución. Estos recursos que no son de CPU incluyen, entre otros, el agente de DB2, memoria de clasificación, espacio de tablas temporal, bloqueos y E/S. Al impedir que una actividad empiece a ejecutarse, esos recursos que no son de CPU adicionales no se consumen y están disponibles para otras actividades.

Asimismo, pueden aplicarse umbrales de simultaneidad en distintos puntos del gestor de bases de datos DB2 para determinar el origen del trabajo que está ejecutándose. Por ejemplo, la colocación de un umbral de simultaneidad en grandes colas que proceden de una carga de trabajo específica limita el consumo de cuota de los recursos disponibles para esa carga de trabajo en particular en una clase de servicio, en comparación con otras cargas de trabajo que contribuyen a la misma clase de servicio.

En resumen, los umbrales de simultaneidad pueden utilizarse para controlar cuándo han de iniciar su ejecución las actividades y cuándo han de empezar a consumir los recursos de CPU y que no son de CPU del sistema. El asignador de gestión de carga de trabajo puede utilizarse para controlar la cantidad de recursos de CPU que estas actividades llegan a consumir después de haberse iniciado.

### **¿Por qué podría llegar a desear utilizar AIX WLM o Linux WLM?**

Aunque utilice el asignador de gestión de carga de trabajo de DB2 para controlar el consumo de CPU de las cargas de trabajo de DB2, a continuación se indican diversas razones por las que también podría desear utilizar AIX WLM o Linux WLM:

- Los gestores de carga de trabajo de sistema operativo (SO) proporcionan supervisión del consumo de recursos en el sistema operativo.
- Los gestores de carga de trabajo de SO pueden proporcionar control para todos los procesos o hebras de todo el sistema principal o LPAR, no sólo de las hebras del gestor de bases de datos DB2. Esto puede ser de ayuda cuando es necesario poder controlar los procesos que compiten con el gestor de bases de datos DB2 en la obtención de recursos.

---

## Capítulo 2. Identificación del trabajo

Un aspecto clave de la implementación satisfactoria de una solución de gestión de carga de trabajo de DB2 consiste en identificar el trabajo.

Existen tres soluciones para la identificación del trabajo: cargas de trabajo, clases de trabajo y códigos de datos.

- Las cargas de trabajo pueden utilizarse para identificar la fuente de trabajo. Puede identificar la fuente durante la conexión o la transacción, utilizando los atributos de sesión clave, como el nombre de la aplicación o el ID de autorización del sistema que emitió el trabajo, por ejemplo.
- Una clase de trabajo le permite identificar el trabajo indicando con precisión determinadas características del trabajo. Puede definir una clase de trabajo para identificar un trabajo de interés, como sentencias que sólo modifican datos sobre el servidor de datos, como las sentencias INSERT, UPDATE o DELETE.
- Los códigos de datos están disponibles como espacio de tablas y como códigos de datos de agrupaciones de almacenamientos intermedios. Le permiten identificar el trabajo indirectamente por el tipo de datos a los que se accede durante la ejecución. Al codificar los datos de interés, puede identificar el trabajo que está accediendo a dichos datos.

---

### Actividades

Una forma de supervisar y controlar las cargas de trabajo es en función de las diferentes actividades. Cada vez que su servidor de datos de DB2 ejecuta el plan de acceso para una sentencia de SQL o XQuery o ejecuta el programa de utilidad de carga, se crea una actividad correspondiente.

Para la supervisión de la carga de trabajo, los elementos de supervisor de uso habitual proporcionan información en términos de unidades de actividad. Por ejemplo, puede obtener información sobre el volumen y el tiempo de respuesta de la carga de trabajo de elementos de supervisor tales como el elemento de supervisor de tiempo de ejecución de actividades (coord\_act\_exec\_time) y la marca de límite superior para actividades simultáneas (concurrent\_act\_top).

Para el control de la carga de trabajo, se aplica la mayoría de los controles y umbrales de la carga de trabajo a cada actividad. Por ejemplo, el umbral ACTIVITYTOTALTIME controla el tiempo máximo que su servidor de datos puede dedicar a procesar una actividad.

### **Sentencias o mandatos que desencadenan actividades en su servidor de datos**

Las sentencias o mandatos siguientes desencadenan actividades su servidor de datos:

- Todas las sentencias de DML
- Todas las sentencias de DDL
- La sentencia de CALL
- El programa de utilidad de carga

## El ciclo de vida de las actividades

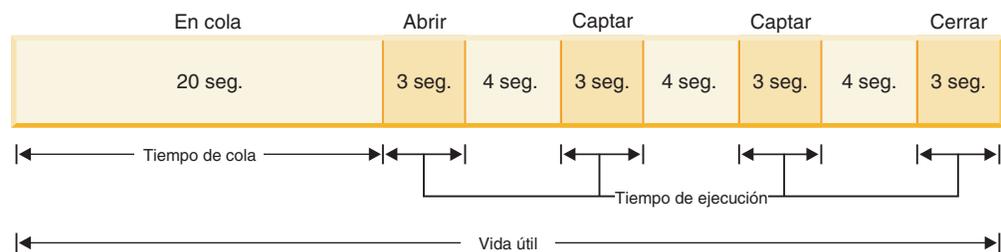
El ciclo de vida de una actividad para una sentencia de DML no incluye el proceso que tiene lugar antes o fuera de la ejecución del plan de acceso. Esto implica que la supervisión basada en actividades no cubre operaciones tales como la conexión con la base de datos o la compilación de SQL en un plan de acceso.

Durante su ciclo de vida, una actividad puede pasar tiempo en diferentes estados, que notifica el elemento de supervisor de sucesos `activity_state`. Algunos de los estados que puede tener una actividad son:

- EXECUTING - Este estado indica que el agente coordinador está trabajando en la actividad. Una actividad que presenta una situación de espera de bloqueo se notifica como en ejecución.
- IDLE - Este estado indica que el agente coordinador espera la siguiente petición de un cliente.
- QUEUED - Algunos umbrales incluyen una cola incorporada. Este estado indica que la actividad espera en la cola su turno para empezar su ejecución.

Los datos de supervisión de la actividad se acumulan al final de la vida útil de una actividad.

En la figura siguiente se muestra el desglose de la vida útil de una consulta de ejecución larga en tiempo en cola y tiempo de ejecución:



## Tipos de sentencias de SQL y desarrollo de aplicaciones

En este apartado se describen las actividades que se crean para las diferentes sentencias de SQL y se identifican los puntos de inicio y finalización en la vida útil de estas actividades. Puede utilizar esta información para comprender cómo se supervisan y controlan las sentencias de SQL a través de las actividades.

**Sentencias SELECT:** una sentencia SELECT se representa mediante una actividad. Esto incluye cualquier petición de cursor, como operaciones FETCH, y subselecciones y subconsultas. La actividad empieza cuando su servidor de datos de DB2 empieza a procesar una sentencia o petición de cursor OPEN, y finaliza cuando su servidor de datos completa el proceso de una sentencia o petición de cursor CLOSE.

**Sentencias SELECT que utilizan cursores WITH HOLD:** cuando se utiliza un cursor WITH HOLD, una aplicación puede abrir un cursor dentro de una unidad de trabajo y cerrarlo en una unidad de trabajo posterior. El cursor permanece abierto durante varias unidades de trabajo. La actividad correspondiente existe mientras el cursor esté abierto, puesto que el ciclo de vida de la actividad finaliza sólo después de que se cierre el cursor.

Sentencia CALL y procedimientos almacenados: una sentencia CALL se representa mediante una actividad, pero la carga útil del procedimiento almacenado puede abarcar actividades anidadas como sigue:

*Tabla 1. Contenido de los procedimientos almacenados y actividades que crean*

Contenido del procedimiento almacenado	Actividades adicionales creadas
Una única sentencia de SQL	Una
Ninguna sentencia de SQL en el procedimiento almacenado	0
Procedimientos de SQL, varias sentencias de SQL y lógica en bucle	Varias actividades, cada una correspondiente a cada invocación de cada sentencia
Una llamada a otro procedimiento almacenado	Actividades para ese procedimiento almacenado

La actividad asociada con la sentencia CALL empieza cuando su servidor de datos de DB2 inicia el proceso de la sentencia o petición, y finaliza cuando se completa el proceso del procedimiento almacenado.

Desencadenantes y UDF: cuando una sentencia de SQL llama a un desencadenante o UDF, no se crea ninguna actividad adicional. El trabajo que realiza ese desencadenante o UDF se acumula en la actividad para la sentencia de SQL que lo ha llamado. Los casos en los que el desencadenante o UDF ejecuta sentencias de SQL adicionales se tratan como cualquier ejecución de sentencias, es decir, se crea una actividad por cada sentencia.

Sentencia PREPARE: no se crea ninguna actividad, porque las actividades no se crean hasta que se ejecuta el plan de acceso.

## Actividades anidadas

Las actividades anidadas no afectan de forma significativa la supervisión y el control de las cargas de trabajo basado en actividades, pero se aplica información adicional.

Las actividades que pueden tener actividades anidadas dentro de ellas son:

- Un procedimiento almacenado
- Un bloque anónimo
- Una rutina autónoma
- Una actividad de DML que ejecuta una UDF
- Una carga de cursor (una actividad de carga que tiene anidada la actividad de cursor)
- Una actividad de DML sujeta a un desencadenante que contiene alguna de las actividades listadas anteriormente como parte de la definición de desencadenante

Las actividades anidadas se notifican en la información de supervisión del modo siguiente:

- Una actividad anidada se indica mediante un ID de UOW principal distinto de 0 y un ID de actividad principal distinto de 0.
- Una actividad anidada no se tiene en cuenta para histogramas o estadísticas derivadas de histogramas.

- Los datos de una actividad anidada tampoco se notifican como parte de las métricas de la actividad principal. Por ejemplo, si el procedimiento que ejecuta una sentencia CALL realiza una inserción que consume 10 segundos de tiempo de procesador, dicho tiempo de procesador se contará solamente como métrica de tiempo de procesador para la actividad de inserción, y no se contará como métrica de tiempo de procesador para la actividad CALL principal.

El control de la carga de trabajo considera las actividades anidadas del modo siguiente:

- Una actividad anidada dentro de un desencadenante o UDF no contribuye al umbral CONCURRENTDBCOORDACTIVITIES.
- Una actividad de cursor anidada dentro de una actividad de carga no contribuye al umbral CONCURRENTDBCOORDACTIVITIES.

## Actividades y el programa de utilidad de carga

La ejecución del programa de utilidad de carga generará varias actividades, una de las cuales será una actividad de carga y varias otras serán de tipo READ, WRITE o OTHER. En el caso de una carga desde un cursor, se creará una actividad adicional para el cursor desde el que se carga la actividad de carga. Esta actividad de cursor es una actividad anidada de la actividad de carga.

## Supervisión de sucesos de actividad

Al supervisar actividades con un supervisor de sucesos de actividad en un entorno de base de datos de varios miembros, conviene tener en cuenta cómo se captura la información de los sucesos de actividades. Si captura sucesos de actividades para todos los miembros, puede que vea más sucesos para una actividad determinada en miembros que no son el miembro coordinador. Podrían grabarse varios registros por actividad porque la existencia de actividad en los miembros remotos no siempre es constante, en función de la secuencia de sucesos de una sección (el formato ejecutable de una sentencia de SQL). Como resultado de ello, puede que se capturen varios registros para la actividad para un miembro que no es el miembro de coordinación. Para entender el proceso que la actividad realiza en el miembro remoto debe tener en cuenta todos los registros para la actividad. Por ejemplo, puede que desee agregar métricas de todos los registros para la actividad en el miembro remoto.

Por otra parte, la actividad sólo se registraría una única vez en los datos de sucesos recopilados en el miembro de coordinación.

---

## Sentencias DDL de gestión de carga de trabajo

Las sentencias DDL de la gestión de carga de trabajo son las sentencias CREATE, ALTER y DROP que se utilizan para trabajar con clases de servicio, cargas de trabajo, conjuntos de clases de trabajo, conjuntos de acciones de trabajo, umbrales e histogramas.

Las sentencias DDL de gestión de carga de trabajo de DB2 son las siguientes:

- CREATE SERVICE CLASS, ALTER SERVICE CLASS y DROP SERVICE CLASS
- CREATE WORKLOAD, ALTER WORKLOAD y DROP WORKLOAD
- GRANT USAGE ON WORKLOAD y REVOKE USAGE ON WORKLOAD
- CREATE THRESHOLD, ALTER THRESHOLD y DROP THRESHOLD

- CREATE WORK CLASS SET, ALTER WORK CLASS SET y DROP WORK CLASS SET
- CREATE WORK ACTION SET, ALTER WORK ACTION SET y DROP WORK ACTION SET
- CREATE HISTOGRAM TEMPLATE, ALTER HISTOGRAM TEMPLATE y DROP HISTOGRAM TEMPLATE

Las sentencias DDL de gestión de carga de trabajo difieren de otras sentencias DDL de DB2 en los aspectos siguientes:

- Sólo se permite una única sentencia DDL de gestión de carga de trabajo no confirmada en todos los miembros de bases de datos. Si ya existe una sentencia DDL de gestión de carga de trabajo no confirmada, las sentencias DDL de gestión de carga de trabajo subsiguientes esperarán hasta que la sentencia DDL de gestión de carga de trabajo no confirmada se confirme o se retrotraiga. Las sentencias DDL de gestión de carga de trabajo se procesan en el orden en que se emiten.
- Cada sentencia DDL de gestión de carga de trabajo debe ir seguida de una sentencia COMMIT o ROLLBACK.
- En una transacción XA no puede emitirse una sentencia DDL de gestión de carga de trabajo. Después de que una conexión emita una sentencia DDL de gestión de carga de trabajo, esa misma conexión debe emitir una sentencia COMMIT o ROLLBACK inmediatamente después de la sentencia DDL de gestión de carga de trabajo. Con las transacciones XA, varias conexiones pueden unir una transacción, y cualquiera de las conexiones puede confirmar o retrotraer la transacción. En esta situación, es imposible garantizar que el entorno de gestión de carga de trabajo se implemente correctamente.
- DB2 para z/OS no reconoce las sentencias DDL de gestión de carga de trabajo de DB2 Database para Linux, UNIX y Windows.

---

## Identificación del trabajo por origen con cargas de trabajo

Las cargas de trabajo identifican el trabajo de entrada basándose en su origen para que pueda supervisarse o gestionarse posteriormente a través de la gestión de carga de trabajo de DB2 independientemente del resto del trabajo. El origen se determina mediante la utilización de los atributos de la conexión de base de datos en la que se envía en trabajo.

Primero se evalúan los atributos de una conexión cuando ésta se establece, y luego la conexión se correlaciona con una definición de carga de trabajo. Esta relación entre una conexión y una definición de carga de trabajo específica se denomina *aparición de la carga de trabajo*. Si cualquiera de los atributos cambia mientras dura esa conexión, la asignación de la carga de trabajo vuelve a evaluarse al iniciarse la siguiente unidad de trabajo tras haberse producido el cambio. Si se encuentra una nueva definición de carga de trabajo que se corresponde mejor con la conexión en ese momento, la aparición de la carga de trabajo antigua (para la carga de trabajo asignada anteriormente) finaliza y se inicia una nueva aparición para la definición de carga de trabajo recién asignada. Aunque cada conexión se asigna a una única carga de trabajo cada vez, puede haber múltiples conexiones (apariciones de carga de trabajo) asignadas simultáneamente a la misma carga de trabajo. Para obtener más información, consulte “Asignación de cargas de trabajo” en la página 24.

Por ejemplo, para asignar todas las conexiones que ha creado la aplicación Accounts a una carga de trabajo REPORTING, que correlaciona las actividades de

esas conexiones para que se ejecuten en la clase de servicio Marketing, emita una sentencia CREATE WORKLOAD, como la siguiente:

```
CREATE WORKLOAD REPORTING APPLNAME('Accounts') SERVICE CLASS Marketing
```

Esto creará la carga de trabajo siguiente:



Figura 2. Carga de trabajo REPORTING

A continuación, otorgue el privilegio USAGE en la carga de trabajo REPORTING como PUBLIC:

```
GRANT USAGE ON WORKLOAD REPORTING TO PUBLIC
```

Para asignar todas las actividades que ha creado la aplicación Accounts en las conexiones que pertenecen al grupo de usuarios de sesión Deptmgr a la carga de trabajo SUMMARY, que correlaciona las actividades con la clase de servicio HumanResources, emita una sentencia como la siguiente:

```
CREATE WORKLOAD SUMMARY SESSION_USER_GROUP('Deptmgr') APPLNAME('Accounts')  
SERVICE CLASS HumanResources
```

Esto creará la carga de trabajo siguiente:

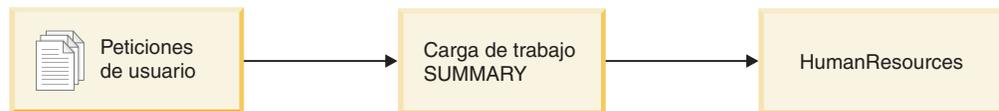


Figura 3. Carga de trabajo SUMMARY

A continuación, otorgue el privilegio USAGE en la carga de trabajo SUMMARY como PUBLIC:

```
GRANT USAGE ON WORKLOAD SUMMARY TO PUBLIC
```

Puede ver las definiciones de carga de trabajo consultando la vista SYSCAT.WORKLOADS y puede ver los atributos de conexión que ha especificado para cada carga de trabajo consultando la vista SYSCAT.WORKLOADCONNATTR. Puede ver quién tiene autorización para utilizar una carga de trabajo consultando la vista SYSCAT.WORKLOADAUTH. Para ver qué apariciones de carga de trabajo existen en el sistema en un momento dado, utilice la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES o MON\_GET\_WORKLOAD.

SYSDEFAULTUSERWORKLOAD es la carga de trabajo por omisión. Cualquier conexión que no se haya asignado a una carga de trabajo definida por el cliente durante la evaluación de la carga de trabajo se asigna a esta carga de trabajo por omisión, que garantiza que todas las conexiones de base de datos se asocian a una carga de trabajo. Por omisión, el trabajo asignado a la carga de trabajo por omisión SYSDEFAULTUSERWORKLOAD se ejecuta en la clase de servicio SYSDEFAULTUSERCLASS.

## Atributos de conexión de base de datos que reciben soporte

Debe especificar un atributo de conexión de base de datos, como mínimo, en la definición de carga de trabajo, y cada atributo de conexión puede tener uno o más valores. Si no especifica un valor para un atributo de conexión específico en la definición de carga de trabajo, el servidor de datos no examinará ese atributo durante la evaluación de la carga de trabajo.

Tabla 2. Atributos de conexión en una definición de carga de trabajo

Atributo de conexión	Descripción
Dirección	Dirección de comunicación real que el cliente utiliza para comunicarse con el servidor de bases de datos. El único protocolo que recibe soporte es TCP/IP. La dirección debe ser una dirección IPv4, una dirección IPv6 o un nombre de dominio seguro.
Nombre de aplicación	Nombre de la aplicación que se ejecuta en el cliente y que el servidor de datos conoce. El nombre de aplicación equivale al valor que se muestra en el campo Nombre de la aplicación en la salida del supervisor del sistema. Consulte el elemento de supervisor <b>app1_name</b> para obtener más información.
ID de autorización del sistema	ID de autorización del usuario que se ha conectado con la base de datos, tal como se ha establecido en el registro especial SYSTEM_USER. Puede cambiar el valor de SYSTEM_USER conectándose como un usuario con un ID de autorización diferente.
ID de autorización de sesión	ID de autorización que se utiliza para la sesión actual de la aplicación, tal como se ha establecido en el registro especial SESSION_USER. Puede cambiar el valor de SESSION_USER utilizando un contexto fiable o la sentencia SET SESSION AUTHORIZATION.
Grupo de ID de autorización de sesión	Grupos a los que pertenece el usuario de la sesión actual.
Rol de ID de autorización de sesión	Roles otorgados al usuario de la sesión actual.
ID de usuario cliente	ID de usuario cliente de la información de cliente, tal como se ha establecido en el registro especial CURRENT CLIENT_USERID (o CLIENT_USERID). Puede cambiar el valor del ID de usuario cliente utilizando uno de los mecanismos definidos que proporcionan los clientes de DB2 específicos, la API sqlseti (establecer información de cliente) o el procedimiento WLM_SET_CLIENT_INFO.

Tabla 2. Atributos de conexión en una definición de carga de trabajo (continuación)

Atributo de conexión	Descripción
Nombre de aplicación cliente	Nombre de aplicación de la información de cliente, tal como se ha establecido en el registro especial CURRENT CLIENT_APPLNAME (o CLIENT APPLNAME). Puede cambiar el valor del nombre de la aplicación cliente utilizando uno de los mecanismos definidos que proporcionan los clientes de DB2 específicos, la API sqleseti o el procedimiento WLM_SET_CLIENT_INFO.
Nombre de estación de trabajo cliente	Nombre de estación de trabajo de la información de cliente, tal como se ha establecido en el registro especial CURRENT CLIENT_WRKSTNNAME (o CLIENT WRKSTNNAME). Puede cambiar el valor del nombre de la estación de trabajo cliente utilizando uno de los mecanismos definidos que proporcionan los clientes de DB2 específicos, la API sqleseti o el procedimiento WLM_SET_CLIENT_INFO.
Serie de contabilidad cliente	Serie de contabilidad de la información de cliente, tal como se ha establecido en el registro especial CURRENT CLIENT_ACCTNG (o CLIENT ACCTNG). Puede cambiar el valor de la serie de contabilidad cliente utilizando la API sqleseti API o el procedimiento WLM_SET_CLIENT_INFO.

## Utilización de caracteres comodín en los atributos de conexión

Algunos atributos de conexión dan soporte a la especificación de un asterisco (\*) como carácter comodín en las sentencias CREATE WORKLOAD y ALTER WORKLOAD. Puede utilizar caracteres comodín en aquellas situaciones en las que un atributo de conexión puede aceptar varios valores similares, cuya coincidencia puede buscarse mediante una expresión regular que utilice caracteres comodín, sin definir atributos de conexión para cada uno de los valores posibles.

El carácter comodín del asterisco (\*) busca la coincidencia con cero o más caracteres. Si necesita buscar una coincidencia con un asterisco, utilice un doble asterisco (\*\*) para especificar el asterisco como carácter literal.

Por ejemplo: si tiene varias aplicaciones de cuentas por cobrar (*accrec01, accrec02 ... accrec15*) y desea que todas éstas pertenezcan a la misma carga de trabajo para que el gestor de carga de trabajo de DB2 las gestione por igual, defina el atributo de conexión `CURRENT CLIENT_APPLNAME('accrec*')` de modo que exista coincidencia con todas estas aplicaciones cuando cree o altere la carga de trabajo. De forma similar, la coincidencia para una aplicación de cuentas por cobrar *acc\*rec* (un nombre que incluye un carácter de asterisco) se obtiene con el atributo de conexión `CURRENT CLIENT_APPLNAME('acc**rec')`.

A continuación se indican los atributos de conexión de carga de trabajo que dan soporte a la utilización de caracteres comodín:

- APPLNAME
- CURRENT CLIENT\_ACCTNG
- CURRENT CLIENT\_APPLNAME
- CURRENT CLIENT\_USERID
- CURRENT CLIENT\_WRKSTNNAME

## **Establecimiento de la información de cliente para identificar peticiones**

Por omisión, muchos servidores de aplicaciones configuran conexiones con la misma información y pasan la misma información de cliente, si la hay, para todas las peticiones de clientes que puedan manejar. Algunos productos, como WebSphere y Cognos, ofrecen la posibilidad de aplicar la información exclusiva sobre cada petición a través de los campos de información del cliente. Esta información identifica de manera exclusiva la petición del usuario final en DB2. La mayoría del resto de los productos ofrecen un método para que el servidor de aplicaciones pueda personalizarse de manera que la información exclusiva del cliente pueda enviarse a DB2 antes de iniciar el proceso de una petición de usuario final.

Mediante la especificación de atributos de cliente exclusivos desde el servidor de aplicaciones, se permite el tratamiento especializado de peticiones en DB2 y la asignación de peticiones de diferentes clientes a diferentes cargas de trabajo (y a diferentes clases de servicio).

## **Orden de evaluación de los atributos de conexión**

A medida que analiza las características de uso del entorno, puede utilizar la sentencia CREATE WORKLOAD para crear cargas de trabajo propias y correlacionar éstas con clases de servicio específicas. Cuando crea la carga de trabajo, define los valores que han de utilizarse para evaluar los atributos de conexión durante la asignación de la carga de trabajo y también el orden en el que ha de evaluarse la carga de trabajo en relación con las otras cargas de trabajo. Puesto que más de una carga de trabajo puede coincidir con los atributos de la conexión de entrada, ser capaz de cambiar el orden de evaluación le permite determinar qué carga de trabajo coincidente ha de seleccionarse. Si el usuario de la sesión tiene o no el privilegio USAGE para la carga de trabajo también determinar qué carga de trabajo ha de seleccionarse. Para obtener más información, consulte "Asignación de cargas de trabajo" en la página 24.

En la siguiente figura se muestra cómo se evalúan varias peticiones para las cargas de trabajo siguiendo el orden A, B, C y D, y se asignan a continuación a cargas de trabajo específicas y se ejecutan en la clase de servicio que tiene aplicación. A las peticiones para las que no existe coincidencia con una carga de trabajo existente, la coincidencia se busca con la carga de trabajo SYSDEFAULTUSERWORKLOAD. Para obtener información acerca de los tipos de actividades que se ejecutan en la clase de mantenimiento por omisión y en la clase del sistema por omisión, consulte "Superclases y subclases de servicio por omisión" en la página 76.

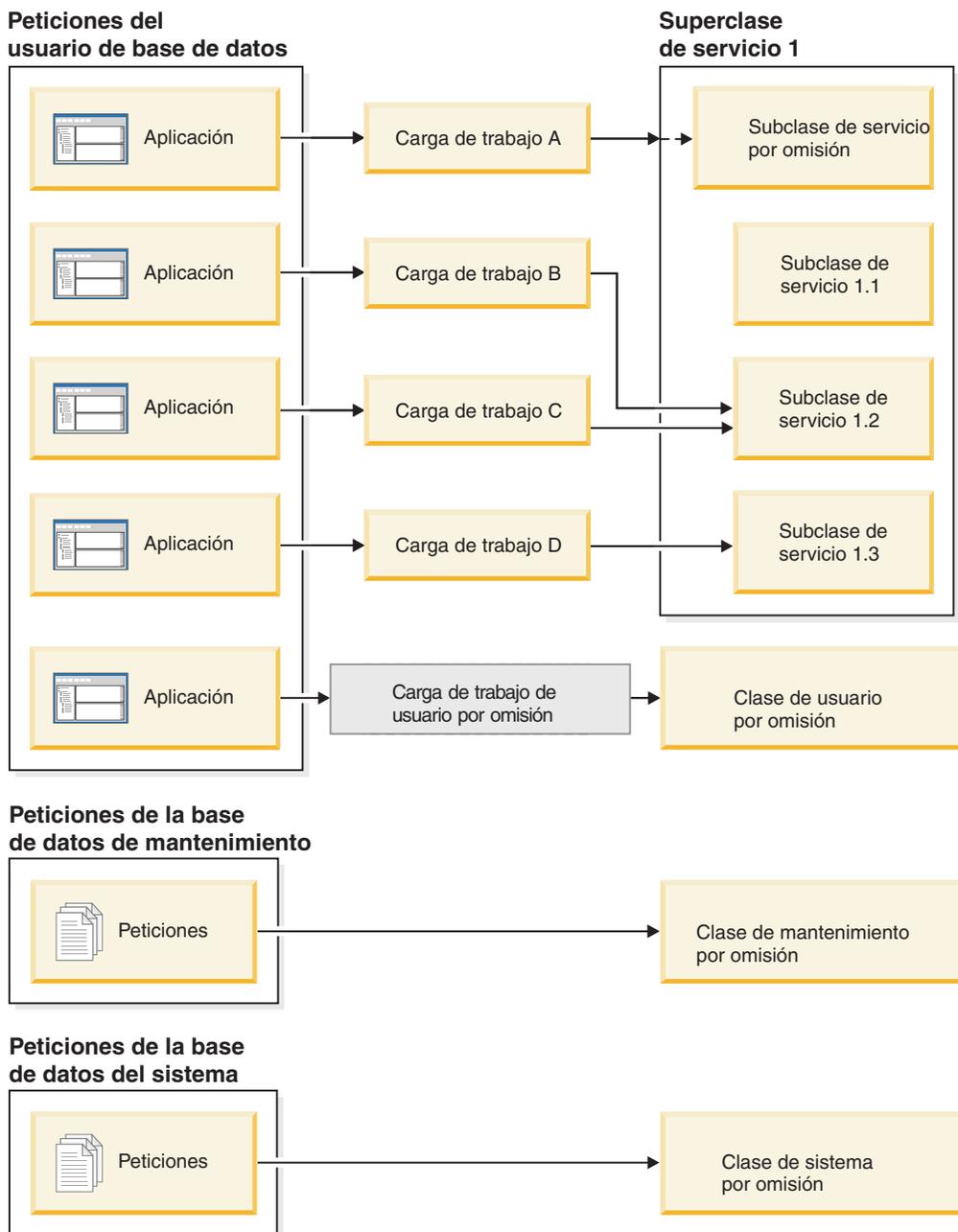


Figura 4. Clases de servicio y cargas de trabajo

## Asignación de cargas de trabajo

Al principio de la primera unidad de trabajo después de que se establezca la conexión con la base de datos, el servidor de datos asigna la conexión a una carga de trabajo evaluando los atributos de conexión de cada carga de trabajo que está habilitada.

El orden en que se evalúan las cargas de trabajo viene determinado por el valor de la columna EVALUATIONORDER de cada carga de trabajo en la tabla SYSCAT.WORKLOADS. Si se encuentra una carga de trabajo con atributos de conexión coincidentes, el servidor de datos comprueba si el usuario de la sesión

actual tiene privilegios USAGE sobre la carga de trabajo. Si el usuario tiene privilegios USAGE sobre la carga de trabajo coincidente, la asignación de la carga de trabajo se completará y se asignará la conexión a esta carga de trabajo. Si el usuario no tiene privilegios USAGE sobre la carga de trabajo coincidente, el servidor de datos continuará evaluando las cargas de trabajo hasta que encuentre una carga de trabajo coincidente en la que el usuario de la sesión tenga privilegios USAGE. Si no se encuentra ninguna carga de trabajo coincidente, el servidor intentará utilizar la carga de trabajo SYSDEFAULTUSERWORKLOAD. Si el usuario de la sesión actual no tiene privilegios USAGE en esa carga de trabajo, se devolverá un error SQL4707N y se rechazará la unidad de trabajo. De lo contrario, la conexión se asignará a la carga de trabajo SYSDEFAULTUSERWORKLOAD.

Puede establecer el orden de evaluación utilizando la palabra clave POSITION de la sentencia CREATE WORKLOAD o ALTER WORKLOAD, como sigue:

- Especificando la posición absoluta de la carga de trabajo en el orden de evaluación, como se muestra en el ejemplo siguiente:

```
CREATE WORKLOAD...POSITION AT 2
```

POSITION AT 2 significa que la carga de trabajo debe colocarse en segundo lugar en el orden de evaluación. Una carga de trabajo coincidente que tenga una posición superior en el orden de evaluación se evaluará primero. Es decir, si las cargas de trabajo en la posición 2 y la posición 3 coinciden, la carga de trabajo en la posición 2 se evaluará antes que la carga de trabajo en la posición 3.

Si la posición que especifica en la sentencia CREATE WORKLOAD o ALTER WORKLOAD es mayor que el número total de cargas de trabajo existentes, la carga de trabajo se colocará antepenúltima en el orden de evaluación, antes de la carga de trabajo SYSDEFAULTUSERWORKLOAD. El efecto es el mismo que especificar POSITION LAST en la sentencia CREATE WORKLOAD o ALTER WORKLOAD.

- Utilizando la palabra clave POSITION BEFORE *nombre-carga-trabajo* o POSITION AFTER *nombre-carga-trabajo*, donde *nombre-carga-trabajo* es la carga de trabajo existente. Esta palabra clave especifica la posición de una carga de trabajo nueva o modificada relativa a otra carga de trabajo en el orden de evaluación, como se muestra en el ejemplo siguiente:

```
ALTER WORKLOAD...POSITION BEFORE workload2
```

Si no especifica la palabra clave POSITION, por omisión la nueva carga de trabajo se colocará después de las demás cargas de trabajo definidas en el orden de evaluación, pero antes de la carga de trabajo SYSDEFAULTUSERWORKLOAD, que siempre se considera la última.

## Reasignación de cargas de trabajo

Una conexión puede cambiar potencialmente su carga de trabajo asignada en cada límite de unidad de trabajo, si cambia uno de sus atributos de conexión o el conjunto de las definiciones de la carga de trabajo. Un límite de unidad de trabajo es el punto en el que una conexión anula su asociación con su transacción actual. Los sucesos siguientes provocan un límite de unidad de trabajo: confirmación, retroacción, XA\_END (satisfactoria), confirmación XA y retroacción XA.

La asignación de cargas de trabajo se vuelve a evaluar al principio de una nueva unidad de trabajo si el servidor de datos detecta que se ha producido uno de los sucesos siguientes:

- Un atributo de conexión relevante ha cambiado. Consulte la tabla en el apartado “Identificación del trabajo por origen con cargas de trabajo” en la página 19 para

obtener una lista de atributos de conexión que puede especificar en una definición de carga de trabajo. La reevaluación de la carga de trabajo también se produce si el ID de autorización de sesión actual cambia debido a una conmutación de la conexión de la base de datos a causa de un contexto de confianza. Para obtener más información, consulte Contextos fiables y conexiones fiables.

- Ha creado o modificado una carga de trabajo.
- Ha otorgado el privilegio USAGE sobre una carga de trabajo a un usuario, grupo o rol, o ha revocado el privilegio USAGE sobre una carga de trabajo a un usuario, grupo o rol.

Una conexión no puede reasignarse a una carga de trabajo diferente mientras haya todavía activa una actividad que abarque un límite de unidad de trabajo. Una actividad puede ser una operación que mantiene recursos entre diferentes unidades de trabajo, como una operación de carga, un procedimiento almacenado o una función de tabla, o bien un cursor WITH HOLD. La aparición de la carga de trabajo actual se ejecuta hasta que todas las actividades se completan. La reasignación de la carga de trabajo se produce al principio de la siguiente unidad de trabajo.

El intento de asignar la carga de trabajo da como resultado un error SQL4707N si se da uno de los dos casos siguientes:

- El servidor de datos intenta asignar la conexión a una carga de trabajo cuyo acceso a la base de datos no se permite. Para obtener más información, consulte “Cómo impedir que las apariciones de una carga de trabajo accedan a la base de datos” en la página 38.
- El servidor de datos intenta asignar la conexión a la carga de trabajo SYSDEFAULTUSERWORKLOAD, pero el usuario de la sesión actual no tiene el privilegio USAGE sobre esta carga de trabajo.

Si tiene autorización ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM, puede asignar su conexión de base de datos a la carga de trabajo SYSDEFAULTADMWORKLOAD, la carga de trabajo de administración por omisión. Consulte “Realización de correcciones mediante la carga de trabajo de administración por omisión” en la página 29 para obtener más información.

## Transacciones XA y reasignación de cargas de trabajo

Las llamadas XA, como XA\_END (satisfactoria), confirmación XA y retroacción XA emiten DB2 COMMIT o ROLLBACK, que indica el final de una unidad de trabajo. Puesto que la reevaluación de la carga de trabajo puede producirse al principio de una unidad de trabajo, estas llamadas XA pueden iniciar la reevaluación de la carga de trabajo, aunque la razón de la reevaluación de la carga de trabajo no esté directamente relacionada con la transacción XA en sí.

## Cargas de trabajo por omisión

La carga de trabajo de usuario por omisión SYSDEFAULTUSERWORKLOAD proporciona una carga de trabajo para su servidor de datos al que están asignadas inicialmente todas las conexiones. La carga de trabajo de administración por omisión SYSDEFAULTADMWORKLOAD le permite emprender acciones administrativas correctivas que de otro modo no pueden realizarse. Ambas cargas de trabajo se crean en el momento de crear la base de datos y no se pueden descartar.

## Carga de trabajo de usuario por omisión (SYSDEFAULTUSERWORKLOAD)

Las conexiones asignadas a la carga de trabajo de usuario por omisión se correlacionan con la superclase de servicio de usuario por omisión SYSDEFAULTUSERCLASS, que proporciona el entorno de ejecución por omisión. Puede correlacionar conexiones con clases de servicio definidas por el usuario creando cargas de trabajo definidas por el usuario. Además, puede modificar SYSDEFAULTUSERWORKLOAD para que correlacione las conexiones con una clase de servicio diferente de SYSDEFAULTUSERCLASS.

Para visualizar la carga de trabajo SYSDEFAULTUSERWORKLOAD puede consultar la tabla SYSCAT.WORKLOADS.

En la tabla siguiente se muestran las columnas que se devuelven para la carga de trabajo SYSDEFAULTUSERWORKLOAD en la vista SYSCAT.WORKLOADS, junto con valores y la especificación de si puede modificar estos valores. Consulte "Asignación de cargas de trabajo" en la página 24 para obtener información sobre cómo asignar una conexión a la carga de trabajo SYSDEFAULTUSERWORKLOAD.

Tabla 3. Entrada SYSDEFAULTUSERWORKLOAD en SYSCAT.WORKLOADS

Columna	Valor	Modificable mediante la sentencia ALTER WORKLOAD si tiene autorización DBADM o WLMADM (y SQLADM para cláusulas COLLECT)
WORKLOADID	1	No
WORKLOADNAME	SYSDEFAULTUSERWORKLOAD	No
EVALUATIONORDER	Antepenúltimo	No
CREATE_TIME	Indicación de fecha y hora de creación de la base de datos	No
ALTER_TIME	Indicación de fecha y hora de la última actualización de la definición de carga de trabajo	No (pero el servidor de datos modifica esta columna cuando actualiza la definición de carga de trabajo)
ENABLED	Y	No
ALLOWACCESS	Y	Sí
SERVICECLASSNAME	SYSDEFAULTSUBCLASS	Sí
PARENTSERVICECLASSNAME	SYSDEFAULTUSERCLASS	Sí
COLLECTAGGACTDATA	N	Sí
COLLECTACTDATA	N	Sí
COLLECTACTPARTITION	C	Sí
COLLECTDEADLOCK	W	Sí
COLLECTLOCKTIMEOUT	W	Sí
COLLECTLOCKWAIT	N	Sí
LOCKWAITVALUE	0	Sí
COLLECTACTMETRICS	N	Sí
COLLECTUOWDATA	N	Sí
EXTERNALNAME	NULL	No
REMARKS	BLANK	Sí

Para obtener más información, consulte SYSCAT.WORKLOADS.

## Carga de trabajo de administración por omisión (SYSDEFAULTADMWORKLOAD)

Esta carga de trabajo permite a los usuarios ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM consultar la base de datos y realizar tareas administrativas o de supervisión en cualquier momento, pero suele utilizarse en estos casos:

- No se permite el acceso a la base de datos a la carga de trabajo a la que está asignado el administrador (es decir, se ha especificado la palabra clave DISALLOW DB ACCESS de la sentencia CREATE WORKLOAD o ALTER WORKLOAD para la carga de trabajo).
- Se ha infringido un umbral, lo que impide al administrador realizar trabajo en la base de datos.

La carga de trabajo SYSDEFAULTADMWORKLOAD es diferente de las otras cargas de trabajo en los aspectos siguientes:

- No puede descartarla ni inhabilitarla.
- No puede especificar DISALLOW DB ACCESS para ella.
- Ningún umbral se aplica a las apariciones de esta carga de trabajo y a las actividades que contiene.
- Puede ejecutar esta carga de trabajo únicamente en la superclase de servicio SYSDEFAULTUSERCLASS. Consulte “Superclases y subclases de servicio por omisión” en la página 76 para obtener más información.
- Para asignar una conexión a esta carga de trabajo, puede utilizar el mandato **SET WORKLOAD** de la interfaz CLP o invocar el procedimiento almacenado WLM\_SET\_CLIENT\_INFO (y especificar SYSDEFAULTADMWORKLOAD para el parámetro **client\_workload**). Para obtener más información, consulte “Realización de correcciones mediante la carga de trabajo de administración por omisión” en la página 29.

Para visualizar la carga de trabajo SYSDEFAULTADMWORKLOAD puede consultar la tabla SYSCAT.WORKLOADS. En la tabla siguiente se muestran las columnas que se devuelven para la carga de trabajo SYSDEFAULTADMWORKLOAD en la vista de catálogo SYSCAT.WORKLOADS, junto con los valores y la especificación de si puede modificarlos:

Tabla 4. Entrada SYSDEFAULTADMWORKLOAD en SYSCAT.WORKLOADS

Columna	Valor	Modificable mediante la sentencia ALTER WORKLOAD si tiene autorización DBADM o WLMADM (y SQLADM para cláusulas COLLECT)
WORKLOADID	2	No
WORKLOADNAME	SYSDEFAULTADMWORKLOAD	No
EVALUATIONORDER	Último	No
CREATE_TIME	Indicación de fecha y hora de creación de la base de datos	No

Tabla 4. Entrada SYSDEFAULTADMWORKLOAD en SYSCAT.WORKLOADS (continuación)

Columna	Valor	Modificable mediante la sentencia ALTER WORKLOAD si tiene autorización DBADM o WLMADM (y SQLADM para cláusulas COLLECT)
ALTER_TIME	Indicación de fecha y hora de la última actualización de la definición de carga de trabajo	No (pero el servidor de datos modifica esta columna cuando actualiza la definición de carga de trabajo)
ENABLED	Y	No
ALLOWACCESS	Y	No
SERVICECLASSNAME	SYSDEFAULTSUBCLASS	No
PARENTSERVICECLASSNAME	SYSDEFAULTUSERCLASS	No
COLLECTAGGACTDATA	N	Sí
COLLECTACTDATA	N	Sí
COLLECTACTPARTITION	C	Sí
COLLECTDEADLOCK	W	Sí
COLLECTLOCKTIMEOUT	W	Sí
COLLECTLOCKWAIT	N	Sí
LOCKWAITVALUE	0	Sí
COLLECTACTMETRICS	N	Sí
COLLECTUOWDATA	N	Sí
EXTERNALNAME	NULL	No
REMARKS	BLANK	Sí

Para obtener más información, consulte SYSCAT.WORKLOADS.

### Realización de correcciones mediante la carga de trabajo de administración por omisión

La carga de trabajo de administración por omisión SYSDEFAULTADMWORKLOAD es una definición de carga de trabajo especial suministrada por DB2 que no está sujeta a ningún umbral de DB2. Utilice esta carga de trabajo para realizar las correcciones oportunas que no puedan realizarse de otra manera, como modificar las definiciones de umbral prohibitivas que impiden que las actividades se ejecuten en una carga de trabajo.

#### Antes de empezar

Utilice el mandato SET WORKLOAD (o el procedimiento WLM\_SET\_CLIENT\_INFO) para asignar una conexión a la carga de trabajo de administración por omisión SYSDEFAULTADMWORKLOAD.

Aunque no se requiere una autorización especial para utilizar el mandato **SET WORKLOAD**, necesita la autorización ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM para asignar una conexión a la carga de trabajo de administración por omisión. De lo contrario, se devuelve SQL0552N durante la asignación de la carga de trabajo.

## Acerca de esta tarea

Puesto que esta carga de trabajo no se ve afectada por los umbrales, tiene un control de gestión de carga de trabajo limitado y no se recomienda para enviar trabajo diario de forma rutinaria.

## Procedimiento

Para asignar una conexión a la carga de trabajo de administración por omisión, emita el mandato **SET WORKLOAD** como se indica a continuación:

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

El momento en que el mandato surte efecto depende de cuándo lo emita:

- Si emite el mandato SET WORKLOAD TO SYSDEFAULTADMWORKLOAD antes de establecer la conexión con la base de datos, una vez establecida la conexión se asigna a SYSDEFAULTADMWORKLOAD al principio de la primera unidad de trabajo.
- Si emite el mandato SET WORKLOAD TO SYSDEFAULTADMWORKLOAD al principio de una unidad de trabajo, después de establecer una conexión a la base de datos, la conexión se asigna a SYSDEFAULTADMWORKLOAD cuando se envía la primera petición que no sea la petición sqleseti (Establecer información de cliente).
- Si emite el mandato SET WORKLOAD TO SYSDEFAULTADMWORKLOAD en medio de una unidad de trabajo, una vez establecida la conexión, ésta se asigna a SYSDEFAULTADMWORKLOAD al principio de la siguiente unidad de trabajo.

Cuando se asigna una conexión a SYSDEFAULTADMWORKLOAD, se lleva a cabo la reasignación de la carga de trabajo al principio de la siguiente unidad de trabajo si se produce una de estas situaciones:

- Se revoca la autorización SYSADM o DBADM desde el usuario de la sesión. En esta situación, se devuelve SQL0552N.
- Se emite un mandato SET WORKLOAD TO AUTOMATIC. Este mandato indica que la siguiente unidad de trabajo no debe asignarse a la carga de trabajo SYSDEFAULTADMWORKLOAD y que se llevará a cabo una evaluación de carga de trabajo normal al principio de la siguiente unidad de trabajo. Para obtener más información, consulte “Asignación de cargas de trabajo” en la página 24.

## Ejemplo

El ejemplo siguiente muestra de qué manera puede utilizar la carga de trabajo SYSDEFAULTADMWORKLOAD para realizar las correcciones oportunas cuando no pueden realizarse otras correcciones.

Si crea un umbral de simultaneidad muy restrictivo para que no pueda ejecutarse ninguna actividad, porque el umbral siempre se vería sobrepasado, el mismo umbral puede impedir que corrija el problema. Para poder alterar el umbral restrictivo, primero debe establecer la carga de trabajo para que el trabajo se ejecute en la carga de trabajo de administración por omisión. Puesto que las actividades que se ejecutan en esta carga de trabajo no están sujetas a los umbrales, puede corregir el problema y establecer de nuevo el valor por omisión para la carga de trabajo.

El umbral que constituye la causa del problema se crea accidentalmente con la sentencia siguiente. La simultaneidad debería haberse establecido en 100 pero se estableció en 0. Este umbral impide de manera eficaz que se ejecute ninguna actividad:

```
CREATE THRESHOLD PROHIBITIVE FOR DATABASE ACTIVITIES
  ENFORCEMENT DATABASE WHEN CONCURRENTDBCOORDACTIVITIES > 0
  STOP EXECUTION
```

**Nota:** Esta sentencia solo sirve para mostrar cómo puede crearse un umbral extraordinariamente restrictivo. Esta sentencia no debe emitirse.

Si intenta ejecutar incluso una sentencia SELECT sencilla, se devuelve un error porque la simultaneidad está establecida en 0:

```
SELECT * FROM SYSCAT.TABLES
```

```
SQL4712N El umbral "PROHIBITIVE" se ha sobrepasado. Código de razón = "6".
SQLSTATE=5U026
```

Antes de realizar las correcciones oportunas, debe establecer la carga de trabajo con el valor por omisión de carga de trabajo de administración:

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

Esta sentencia solo puede emitirla alguien que tenga autorización ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM y hace que cualquier conexión se asigne a la carga de trabajo SYSDEFAULTADMWORKLOAD, donde las actividades no están sujetas al umbral restrictivo.

Ahora el problema podrá corregirse modificando el umbral de manera que las actividades puedan ejecutarse:

```
ALTER THRESHOLD PROHIBITIVE WHEN CONCURRENTDBCOORDACTIVITIES > 100 STOP EXECUTION
```

Una vez corregido, vuelva a cambiar la carga de trabajo para que la conexión deje de estar asignada a SYSDEFAULTADMWORKLOAD, sino a cualquier otra carga de trabajo a la que estuviera asignada con anterioridad:

```
SET WORKLOAD TO AUTOMATIC
```

La misma sentencia SELECT que se utilizó previamente debería finalizar ahora correctamente:

```
SELECT * FROM SYSCAT.TABLES
```

```
...
```

```
DB20000I El mandato SQL ha finalizado correctamente.
```

## Creación de una carga de trabajo

Utilice la sentencia CREATE WORKLOAD para añadir una carga de trabajo a los catálogos.

### Antes de empezar

Para crear una carga de trabajo, se necesita la autorización WLMADM o DBADM.

Consulte los temas siguientes para obtener más información sobre los requisitos:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Apéndice A, “Normas de denominación generales”, en la página 545

## Procedimiento

Para crear una carga de trabajo:

1. Especifique una o más de las propiedades siguientes para la carga de trabajo utilizando la sentencia CREATE WORKLOAD:
  - El nombre de la carga de trabajo.
  - Los atributos de conexión. Para que se produzca una coincidencia, la conexión de entrada debe proporcionar atributos de conexión que coincidan con los que especificó para la carga de trabajo. Para obtener más información, consulte “Identificación del trabajo por origen con cargas de trabajo” en la página 19. Cuando especifique los atributos de conexión, observe que los valores tienen los operadores OR y los atributos tienen los operadores AND: por ejemplo, UserID (jose OR sonia OR laura) AND Application (SAS).
  - Un valor que indica si se permite que las apariciones de esta carga de trabajo accedan a la base de datos. Por omisión, las apariciones de esta carga de trabajo sí pueden acceder a la base de datos.
  - Un valor que indica si la carga de trabajo está habilitada o inhabilitada. Por omisión, la carga de trabajo está habilitada.
  - La clase de servicio donde va a ejecutarse el trabajo enviado por las apariciones de esta carga de trabajo. La superclase de servicio SYSDEFAULTUSERCLASS es el valor por omisión. Si especifica una superclase de servicio definida por el usuario, el trabajo enviado por las apariciones de la carga de trabajo se ejecuta en la subclase de servicio SYSDEFAULTSUBCLASS de la superclase de servicio.

**Nota:** No puede especificar la subclase de servicio SYSDEFAULTSUBCLASS bajo cualquier superclase de servicio, incluida la superclase de servicio SYSDEFAULTUSERCLASS.

Si ha definido una carga de trabajo que apunta a una superclase de servicio pero no desea que el trabajo que ésta envía se ejecute en la subclase de servicio SYSDEFAULTSUBCLASS por omisión, puede cambiar la definición de la carga de trabajo para que apunte directamente a una subclase de servicio definida por el usuario, o puede utilizar un conjunto de acciones de trabajo definido en la superclase de servicio para correlacionar partes individuales de trabajo con una subclase de servicio distinta. Para obtener más información, consulte “Acciones de trabajo y conjuntos de acciones de trabajo” en la página 105.

- La posición de la carga de trabajo relativa a otras cargas de trabajo cuando se guardan en memoria caché. La posición de la nueva carga de trabajo determina el orden en el que se evalúan durante la asignación de cargas de trabajo. Por omisión, la nueva carga de trabajo se coloca al final, lo que significa que se evalúa en último lugar, inmediatamente antes de tomar en consideración la carga de trabajo de usuario por omisión. Para obtener más información, consulte “Asignación de cargas de trabajo” en la página 24.
- El nivel de recopilación de la supervisión de métricas de actividad para las actividades enviadas por conexiones asociadas con esta carga de trabajo. La configuración de la recopilación de las métricas de actividad por omisión de una carga de trabajo es NONE. Tenga en cuenta que la configuración efectiva de la recopilación de actividades correspondiente a las actividades es la combinación del nivel de recopilación de las métricas de actividad de la carga de trabajo y el parámetro de configuración `mon_act_metrics` de la base de datos.

- El tipo de información de actividad que se va a recopilar. Por omisión, no se envía información sobre las actividades asociadas con la carga de trabajo al supervisor de sucesos de actividades.
  - La información de actividad agregada que se va a recopilar. La información de actividad agregada que se utiliza para la carga de trabajo solo cambia después de confirmar la operación CREATE WORKLOAD.
  - La información sobre sucesos de tiempo de espera de bloqueo que se va a recopilar. Por omisión, se envían datos sobre un suceso de bloqueo al supervisor de sucesos de bloqueo si está activo; esto ocurre cuando se produce el suceso de bloqueo, pero no se envían sucesos de tiempo de espera de bloqueo que hayan tenido lugar anteriormente (WITHOUT HISTORY).
  - La información sobre puntos muertos que se va a recopilar. Por omisión, se envían datos sobre un suceso de punto muerto al supervisor de sucesos de bloqueo si está activo; esto ocurre cuando se produce el suceso de punto muerto, pero no se envían sucesos de punto muerto que hayan tenido lugar anteriormente (WITHOUT HISTORY).
  - La información sobre esperas de bloqueo que se va a recopilar. Por omisión, no se recopila información sobre esperas de bloqueo si el bloqueo no se ha adquirido en el tiempo de espera especificado.
  - La información de unidad de trabajo para cada transacción asociada con esta carga de trabajo que se enviará al supervisor de sucesos de unidad de trabajo si está activo, cuando finaliza una unidad de trabajo. Por omisión, no se envía información de unidad de trabajo.
  - Las plantillas de histograma que debe utilizar la carga de trabajo como plantillas para sus histogramas. Las plantillas de histograma especificadas están reflejadas en la vista SYSCAT.HISTOGRAMTEMPLATEUSE. Para obtener más información acerca de histogramas y plantillas de histograma, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.
2. Confirme los cambios. Cuando confirma los cambios, la carga de trabajo se añade a la vista SYSCAT.WORKLOADS. Al confirmar el cambio se desencadena la reevaluación de la carga de trabajo al principio de la siguiente unidad de trabajo de cada aplicación. En función de la carga de trabajo que se elija, es posible que la aplicación se reasigne a una carga de trabajo diferente.

## Qué hacer a continuación

Después de crear una carga de trabajo, puede necesitar otorgar el privilegio USAGE a uno o más usuarios de la sesión. (Los usuarios de sesión con autorización WLMADM o DBADM tienen un privilegio implícito para utilizar cualquier carga de trabajo.) Aunque una conexión proporciona una coincidencia exacta con los atributos de conexión de la carga de trabajo, si el usuario de la sesión no tiene el privilegio USAGE sobre la carga de trabajo, el servidor de datos no tomará en consideración la carga de trabajo cuando realice una evaluación de cargas de trabajo. Para obtener más información, consulte “Concesión del privilegio USAGE en una carga de trabajo” en la página 39.

## Alteración de una carga de trabajo

Una sentencia ALTER WORKLOAD cambia una carga de trabajo en los catálogos.

## Antes de empezar

Para alterar una carga de trabajo, se requiere la autorización SQLADM, WLMADM o DBADM. Para especificar una cláusula que no sea COLLECT, el id de autorización debe incluir la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

## Procedimiento

Para alterar una carga de trabajo:

1. Especifique una o más de las propiedades siguientes para la carga de trabajo utilizando la sentencia ALTER WORKLOAD:
  - Los atributos de conexión. Puede añadir atributos de conexión a la definición de carga de trabajo, o bien descartarlos de la misma, a menos que se trate de la carga de trabajo SYSDEFAULTUSERWORKLOAD o SYSDEFAULTADMWORKLOAD. Para que se produzca una coincidencia, la conexión de entrada debe proporcionar atributos de conexión que coincidan con los que especificó para la carga de trabajo. Para obtener más información, consulte “Identificación del trabajo por origen con cargas de trabajo” en la página 19. Para ver los atributos de conexión de una carga de trabajo, realice una consulta de la vista SYSCAT.WORKLOADCONNATTR.
  - Un valor que indica si se permite una aparición de esta carga de trabajo para acceder a la base de datos. Por omisión, se permite una aparición de esta carga de trabajo para acceder a la base de datos. No puede eliminar el acceso a la base de datos desde la carga de trabajo SYSDEFAULTADMWORKLOAD.
  - Un valor que indica si la carga de trabajo está habilitada o inhabilitada. Por omisión, la carga de trabajo está habilitada. No puede inhabilitar la carga de trabajo SYSDEFAULTUSERWORKLOAD o SYSDEFAULTADMWORKLOAD.
  - La clase de servicio bajo la que se ejecutarán las apariciones de esta carga de trabajo. La superclase de servicio SYSDEFAULTUSERCLASS es el valor por omisión. Si especifica una superclase de servicio definida por el usuario, puede especificar una subclase de servicio bajo la superclase de servicio. No puede especificar la subclase SYSDEFAULTSUBCLASS bajo cualquier superclase de servicio, incluida la superclase de servicio SYSDEFAULTUSERCLASS. Además, no puede especificar la superclase de servicio SYSDEFAULTSYSTEMCLASS o SYSDEFAULTMAINTENANCECLASS.
  - La posición de la carga de trabajo relativa a otras cargas de trabajo, que determina el orden en el que la carga de trabajo se evalúa durante la asignación de la carga de trabajo. No puede especificar la posición de la carga de trabajo SYSDEFAULTUSERWORKLOAD o SYSDEFAULTADMWORKLOAD. Para obtener más información, consulte “Asignación de cargas de trabajo” en la página 24.
  - El tipo de información de actividad que se va a recopilar. Por omisión, no se envía información sobre las actividades asociadas con la carga de trabajo al supervisor de sucesos de actividades.
  - El nivel de recopilación de la supervisión de métricas de actividad para las actividades enviadas por conexiones asociadas con esta carga de trabajo. Tenga en cuenta que la configuración efectiva de la recopilación de actividades correspondiente a las actividades es la combinación del nivel de recopilación de las métricas de actividad de la carga de trabajo y el parámetro de configuración **mon\_act\_metrics** de la base de datos.

- La información de actividad agregada que se va a recopilar. La información de actividad agregada que se utiliza para la carga de trabajo solo cambia después de confirmar la operación ALTER WORKLOAD.
  - La información de sucesos del tiempo de espera excedido de bloqueo que se enviará al supervisor de sucesos de bloqueo, si está activo, cuando se produce un suceso de bloqueo.
  - La información de los puntos muertos que se enviará al supervisor de sucesos de bloqueo, si está activo, cuando se produzca un suceso de punto muerto.
  - La información sobre esperas de bloqueo que se va a recopilar.
  - La información de unidad de trabajo para cada transacción asociada con esta carga de trabajo que se enviará al supervisor de sucesos de unidad de trabajo si está activo, cuando finaliza una unidad de trabajo.
  - Las plantillas de histograma que debe utilizar la carga de trabajo como plantillas para sus histogramas. Las plantillas de histograma especificadas están reflejadas en la vista SYSCAT.HISTOGRAMTEMPLATEUSE. Para obtener más información acerca de histogramas y plantillas de histograma, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.
2. Confirme los cambios. Cuando confirma los cambios, la carga de trabajo se actualiza en la vista SYSCAT.WORKLOADS. El cambio confirmado desencadena la reevaluación de la carga de trabajo al principio de la siguiente unidad de trabajo de cada aplicación. En función de la carga de trabajo que se elija, es posible que la aplicación se reasigne a una carga de trabajo diferente.

## Qué hacer a continuación

Puede necesitar otorgar el privilegio USAGE a uno o más usuarios de la sesión. (Los usuarios de sesión con autorización DBADM tienen un privilegio implícito para utilizar cualquier carga de trabajo.) Aunque una conexión proporciona una coincidencia exacta con los atributos de conexión de la carga de trabajo, si el usuario de la sesión no tiene el privilegio USAGE sobre la carga de trabajo, el servidor de datos no asocia la conexión con la carga de trabajo para crear una aparición de la carga de trabajo. Para obtener más información, consulte “Concesión del privilegio USAGE en una carga de trabajo” en la página 39.

## Habilitación de una carga de trabajo

El servidor de datos de DB2 comprueba los atributos de conexión especificados para una carga de trabajo frente a los atributos de conexión de la sesión actual. El servidor de datos no toma en consideración una carga de trabajo inhabilitada cuando busca una carga de trabajo coincidente.

### Antes de empezar

Para alterar una carga de trabajo, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### Acerca de esta tarea

Cuando se crea una carga de trabajo queda habilitada por omisión. Si crea una carga de trabajo y la inhabilita, deberá habilitarla para que el servidor de datos la tome en cuenta cuando realice una evaluación de cargas de trabajo.

## Procedimiento

Para habilitar una carga de trabajo:

1. Identifique la carga de trabajo que desea habilitar. Puede mostrar el conjunto de cargas de trabajo inhabilitadas consultando la vista SYSCAT.WORKLOADS, tal como se muestra en el ejemplo siguiente:

```
SELECT * FROM SYSCAT.WORKLOADS WHERE ENABLED='N'
```

2. Utilice la sentencia ALTER WORKLOAD para habilitar la carga de trabajo inhabilita:

```
ALTER WORKLOAD...ENABLE
```

Si la sentencia ALTER WORKLOAD es satisfactoria, la definición de la carga de trabajo se graba en el catálogo de la base de datos.

3. Confirme los cambios. Cuando confirma los cambios, la carga de trabajo se actualiza en la vista SYSCAT.WORKLOADS.

## Resultados

La habilitación de una carga de trabajo surte efecto al principio de la siguiente unidad de trabajo. En ese punto, se vuelve a evaluar la carga de trabajo, y el servidor de datos toma en consideración la carga de trabajo recién habilitada cuando vuelve a realizar una evaluación de cargas de trabajo.

## Inhabilitación de una carga de trabajo

Utilice esta tarea para impedir que cargas de trabajo específicas se tomen en cuenta durante la asignación de cargas de trabajo. Si inhabilita una carga de trabajo, el servidor de datos no la tendrá en consideración cuando busque una carga de trabajo coincidente. En su lugar, el servidor de datos asignará la unidad de trabajo a la siguiente carga de trabajo coincidente. Si no se encuentra ninguna carga de trabajo definida por el cliente que coincida, el trabajo se asignará a la carga de trabajo por omisión.

## Antes de empezar

Para crear o alterar una carga de trabajo, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

## Procedimiento

Para inhabilitar una carga de trabajo:

1. Utilice la opción DISABLE de la sentencia ALTER WORKLOAD para inhabilitar la carga de trabajo:

```
ALTER WORKLOAD...DISABLE
```

2. Confirme los cambios. Cuando confirma los cambios, la carga de trabajo se actualiza en la vista SYSCAT.WORKLOADS.

## Resultados

La inhabilitación de una carga de trabajo surte efecto al principio de la siguiente unidad de trabajo. En ese punto, se vuelve a evaluar la carga de trabajo, y se asigna la conexión a la siguiente carga de trabajo habilitada que coincida con los

atributos de conexión y para la que tenga autorización.

## Descarte de una carga de trabajo

Descartar una carga de trabajo la elimina del catálogo de bases de datos.

### Antes de empezar

Para descartar una carga de trabajo, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### Procedimiento

Para descartar una carga de trabajo:

1. Inhabilite la carga de trabajo especificando la sentencia ALTER WORKLOAD. Consulte “Inhabilitación de una carga de trabajo” en la página 36 para obtener más información. Al inhabilitar la carga de trabajo se impide la ejecución en la base de datos de nuevas apariciones de la carga de trabajo.
2. Asegúrese de que no exista ninguna aparición de esta carga de trabajo en ejecución mediante la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES. Para obtener más información, consulte la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES.  
La función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES devuelve los descriptores de contexto de aplicación que corresponden a las apariciones de carga de trabajo activas. Puede utilizar el mandato **FORCE APPLICATION** para finalizar las aplicaciones utilizando los descriptores de contexto de la aplicación.
3. Descarte la carga de trabajo especificando la sentencia DROP WORKLOAD. Por ejemplo, para descartar la carga de trabajo ACCTNG, especifique la sentencia siguiente:  

```
DROP WORKLOAD ACCTNG
```
4. Confirme los cambios. Cuando confirme los cambios, la carga de trabajo se eliminará de la vista SYSCAT.WORKLOADS. Además, la información de autorización de la carga de trabajo se elimina de la vista SYSCAT.WORKLOADAUTH.

## Cómo permitir apariciones de una carga de trabajo para acceder a la base de datos

Si tiene una carga de trabajo que no dispone de permiso para acceder a la base de datos pero ahora desea permitir que se ejecuten apariciones de esa carga de trabajo, modifique la carga de trabajo de manera que tenga permiso para acceder a la base de datos. Por omisión, cuando se crea una carga de trabajo, se le permite el acceso a la base de datos.

### Antes de empezar

Para modificar una carga de trabajo de manera que pueda acceder a una base de datos, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### **Acerca de esta tarea**

Cuando se impide que una carga de trabajo acceda a la base de datos, el servidor de datos sigue examinando dicha carga de trabajo a la hora de realizar asignaciones de cargas de trabajo. Sin embargo, todas las apariciones de esa carga de trabajo se rechazan con un error.

### **Procedimiento**

Para permitir que una carga de trabajo acceda a la base de datos:

1. Utilice la opción `ALLOW DB ACCESS` de la sentencia `ALTER WORKLOAD` para permitir que la carga de trabajo acceda a la base de datos. Por ejemplo, para permitir que una carga de trabajo llamada `WL1` acceda a la base de datos, especifique la sentencia siguiente:  
`ALTER WORKLOAD WL1 ALLOW DB ACCESS`
2. Confirme los cambios. Cuando confirma los cambios, la carga de trabajo se actualiza en la vista `SYSCAT.WORKLOADS`.

### **Resultados**

La modificación de una carga de trabajo para que sus apariciones puedan acceder a la base de datos surte efecto cuando el servidor de datos analiza la siguiente unidad de trabajo para dicha carga de trabajo. Por ejemplo, si ha especificado `DISALLOW DB ACCESS` para la carga de trabajo `A` y modifica la carga de trabajo especificando `ALLOW DB ACCESS`, se permitirá la ejecución de las nuevas apariciones de la carga de trabajo `A`. Previamente, cualquier aparición de la carga de trabajo `A` se habrá rechazado con un error.

## **Cómo impedir que las apariciones de una carga de trabajo accedan a la base de datos**

Utilice esta tarea para controlar qué cargas de trabajo pueden acceder a la base de datos. Antes de que una aparición de carga de trabajo empiece a ejecutarse, el servidor de datos comprueba si la carga de trabajo tiene permiso para acceder a la base de datos. Si deniega el permiso a la aparición de la carga de trabajo de acceder a la base de datos, se devuelve un error que indica que la aparición de la carga de trabajo se ha rechazado.

### **Antes de empezar**

Para impedir que una carga de trabajo acceda a la base de datos, se necesita la autorización `WLMADM` o `DBADM`.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### **Acerca de esta tarea**

Impedir a una aparición de carga de trabajo difiere de inhabilitar una carga de trabajo. Cuando se inhabilita una carga de trabajo, la definición de carga de trabajo no se guarda en la memoria caché y por lo tanto no se toma en consideración para la asignación de la carga de trabajo.

## Procedimiento

Para impedir que una carga de trabajo acceda a una base de datos:

1. Utilice la opción `DISALLOW DB ACCESS` de la sentencia `ALTER WORKLOAD`, tal como se muestra en el ejemplo siguiente:

```
ALTER WORKLOAD nombre_carga_trabajo
DISALLOW DB ACCESS
...
```

2. Confirme los cambios. Cuando confirma los cambios, la carga de trabajo se actualiza en la vista `SYSCAT.WORKLOADS`.

## Resultados

La alteración de una carga de trabajo para impedir que sus apariciones accedan a una base de datos surte efecto al principio de la siguiente unidad de trabajo para aquellas apariciones de carga de trabajo que ya se están ejecutando. Por ejemplo, si especifica `ALLOW DB ACCESS` para la carga de trabajo A y altera la carga de trabajo especificando `DISALLOW DB ACCESS`, las apariciones de la carga de trabajo A que ya se están ejecutando reciben un error de SQL al principio de la siguiente unidad de trabajo. Las nuevas apariciones de la carga de trabajo A se rechazarán.

## Concesión del privilegio USAGE en una carga de trabajo

Para asociar una carga de trabajo con una conexión, el usuario de la sesión debe tener el privilegio `USAGE` en dicha carga de trabajo. Los usuarios que tienen la autorización `ACCESSCTRL`, `DATAACCESS`, `DBADM`, `SECADM` o `WLMADM` disponen implícitamente del privilegio `USAGE` en todas las cargas de trabajo.

### Antes de empezar

Para utilizar la sentencia `GRANT USAGE ON WORKLOAD`, necesita la autorización `ACCESSCTRL`, `SECADM` o `WLMADM`.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### Acerca de esta tarea

Cuando el servidor de datos encuentra una carga de trabajo que coincide con los atributos de una conexión de entrada, dicho servidor comprueba si el usuario de la sesión tiene el privilegio `USAGE` en esa carga de trabajo. Si el usuario de la sesión no tiene el privilegio `USAGE` en esa carga de trabajo, el servidor de datos busca la siguiente carga de trabajo coincidente. (En otras palabras, las cargas de trabajo para las que el usuario de la sesión no dispone del privilegio `USAGE` se consideran como si no existieran.) Por lo tanto, el privilegio `USAGE` de carga de trabajo le ofrece la facultad de controlar a qué carga de trabajo, de entre las cargas de trabajo coincidentes, debe asignarse un usuario, grupo o rol. Por ejemplo, puede definir más de una carga de trabajo con los mismos atributos de conexión y conceder el privilegio `USAGE` en cada una de esas cargas de trabajo únicamente para determinados usuarios, grupos o roles. Para obtener más información, consulte “Asignación de cargas de trabajo” en la página 24.

El cliente puede definir el ID de usuario de cliente, el nombre de la aplicación de cliente, el nombre de la estación de trabajo de cliente y la serie de contabilidad de cliente (que son algunos de los atributos de conexión que se utilizan para asignar

una conexión a una carga de trabajo) sin autorización. Por lo tanto, el privilegio USAGE de la carga de trabajo también permite controlar qué usuario de sesión tiene autorización para utilizar una carga de trabajo.

Puede ver la información del privilegio USAGE consultando la vista SYSCAT.WORKLOADAUTH.

Si crea una base de datos con la opción RESTRICTIVE, el privilegio USAGE de la carga de trabajo SYSDEFAULTUSERWORKLOAD no se otorga a PUBLIC durante la creación de la base de datos. Debe conceder explícitamente el privilegio USAGE en esta carga de trabajo a usuarios que no sean WLMADM ni DBADM. Si el usuario de la sesión no tiene el privilegio USAGE en ninguna de las cargas de trabajo, incluida SYSDEFAULTUSERWORKLOAD, se devuelve SQL4707N cuando el servidor de datos intenta asociar una carga de trabajo con la conexión de la base de datos.

## Procedimiento

Para otorgar el privilegio USAGE en una carga de trabajo:

1. Utilice la sentencia GRANT USAGE ON WORKLOAD. Puede otorgar el privilegio USAGE a usuarios, grupos o roles específicos, o a PUBLIC. Por ejemplo, para otorgar el privilegio USAGE en la carga de trabajo ACCOUNTS al grupo CPA, debe emitir la sentencia siguiente:  

```
GRANT USAGE ON WORKLOAD ACCOUNTS TO GROUP CPA
```

No puede otorgar el privilegio USAGE en la carga de trabajo SYSDEFAULTADMWORKLOAD. La carga de trabajo SYSDEFAULTADMWORKLOAD solo la pueden utilizar usuarios ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM que emitan el mandato SET WORKLOAD TO SYSDEFAULTADMWORKLOAD.
2. Confirme los cambios. Cuando confirme los cambios, se actualizará la vista SYSCAT.WORKLOADAUTH. Hasta que no se confirme la sentencia GRANT, el servidor de datos no tendrá en consideración la carga de trabajo a la hora de realizar asignaciones de carga de trabajo para los recién autorizados usuarios, grupos o roles.

## Revocación del privilegio USAGE en una carga de trabajo

Utilice la sentencia REVOKE USAGE ON WORKLOAD para revocar el privilegio USAGE en una carga de trabajo.

### Antes de empezar

Para utilizar la sentencia REVOKE USAGE ON WORKLOAD, necesita la autorización ACCESSCTRL, SECADM o WLMADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### Acerca de esta tarea

No puede revocar explícitamente el privilegio USAGE en la carga de trabajo SYSDEFAULTADMWORKLOAD. Esta carga de trabajo solo pueden utilizarla los usuarios ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM que emitan el mandato SET WORKLOAD TO SYSDEFAULTADMWORKLOAD. Por lo tanto, las sentencias REVOKE USAGE ON WORKLOAD no funcionan para SYSDEFAULTADMWORKLOAD.

## Procedimiento

Para revocar el privilegio USAGE en una carga de trabajo:

1. Utilice la sentencia REVOKE USAGE ON WORKLOAD. Puede revocar el privilegio USAGE de usuarios, grupos o roles específicos, o a PUBLIC. Por ejemplo, para revocar el privilegio USAGE en la carga de trabajo ACCOUNTS de PUBLIC, debe especificar la sentencia siguiente:

```
REVOKE USAGE ON WORKLOAD ACCOUNTS FROM PUBLIC
```

2. Confirme los cambios. Cuando confirme los cambios, se actualizará la vista SYSCAT.WORKLOADAUTH. Hasta que no se confirme la sentencia REVOKE, el servidor de datos tendrá en consideración la carga de trabajo a la hora de realizar asignaciones de carga de trabajo.

## Ejemplo: Asignación de cargas de trabajo

Al principio de la primera unidad de trabajo después de que se establezca la conexión con la base de datos, el servidor de datos asigna la conexión a una carga de trabajo evaluando los atributos de conexión de cada carga de trabajo que está habilitada.

La reevaluación de la carga de trabajo se produce al principio de cada unidad de trabajo si el valor de un atributo de conexión o la propia definición de la carga de trabajo cambia en el transcurso de la unidad de trabajo.

En la figura siguiente se muestra una asignación de carga de trabajo. Los usuarios del grupo Marketing, que envían consultas a través de AppA, se asignan a la carga de trabajo APPAQUERIES. No se asignan a la carga de trabajo PAYROLL, aunque PAYROLL se encuentre antes que APPAQUERIES, porque la definición de la carga de trabajo PAYROLL especifica la palabra clave SESSION\_USER GROUP Finance. Los usuarios del grupo Finance, que envían consultas a través de AppA, se asignan a la carga de trabajo FINANCE. No se asignan a la carga de trabajo PAYROLL, aunque sea más específica y especifique tanto AppA como Finance en su definición, porque la carga de trabajo FINANCE se encuentra antes que la carga de trabajo PAYROLL. Los usuarios del grupo Marketing, que envían consultas a través de AppB, se asignan a la carga de trabajo SYSDEFAULTUSERWORKLOAD, porque ninguno de los atributos de conexión especificados en las definiciones de carga de trabajo FINANCE, PAYROLL o APPAQUERIES coinciden con la aplicación AppB o el grupo Marketing.

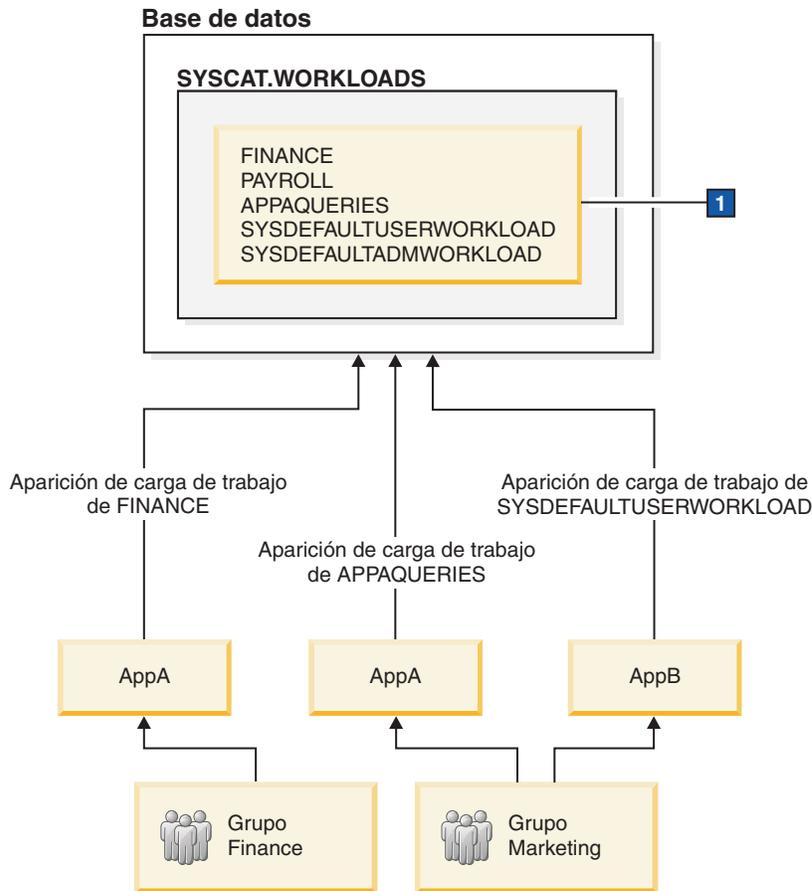


Figura 5. Ejemplo de asignación de cargas de trabajo

**1** En la figura anterior, las sentencias CREATE WORKLOAD son las siguientes:

```
CREATE WORKLOAD PAYROLL APPLNAME ('AppA') SESSION_USER GROUP ('FINANCE')
SERVICE CLASS SC1
```

```
CREATE WORKLOAD APPAQUERIES APPLNAME('AppA') POSITION LAST
SERVICE CLASS SC2
```

```
CREATE WORKLOAD FINANCE SESSION_USER GROUP ('FINANCE') SERVICE CLASS SC1
POSITION BEFORE PAYROLL
```

Por omisión, muchos servidores de aplicaciones configuran conexiones con la misma información y pasan la misma información de cliente, si la hay, para todas las peticiones de clientes que puedan manejar. Algunos productos, como WebSphere y Cognos, ofrecen la posibilidad de aplicar la información exclusiva sobre cada petición a través de los campos de información del cliente. Esta información identifica de manera exclusiva la petición del usuario final en DB2. La mayoría del resto de los productos ofrecen un método para que el servidor de aplicaciones pueda personalizarse de manera que la información exclusiva del cliente pueda enviarse a DB2 antes de iniciar el proceso de una petición de usuario final.

Mediante la especificación de atributos de cliente exclusivos desde el servidor de aplicaciones, se permite el tratamiento especializado de peticiones en DB2 y la asignación de peticiones de diferentes clientes a diferentes cargas de trabajo (y a diferentes clases de servicio).

En la figura siguiente se muestra un ejemplo de un entorno de tres niveles en el que diferentes aplicaciones de usuario envían consultas (marketing.exe, auditing.exe y reporting.exe), a través de un servidor de aplicaciones que establece una conexión con la base de datos utilizando el usuario de sesión APPUSER. Se definen tres cargas de trabajo: una para las consultas enviadas por marketing.exe, una para las consultas enviadas por reporting.exe y otra para las demás consultas. Como se muestra en la figura, para asignar consultas enviadas por marketing.exe a la carga de trabajo MARKETING, el servidor de aplicaciones llama a la API sqlesei para establecer el valor del registro especial CURRENT CLIENT\_APPLNAME en marketing.exe. De modo parecido, para asignar consultas enviadas por reporting.exe a la carga de trabajo REPORTING, el servidor llama a sqlesei para establecer el valor del registro especial CURRENT CLIENT\_APPLNAME en reporting.exe. Tenga en cuenta que, en la figura, cuando el servidor llama a sqlesei para establecer el registro especial CURRENT CLIENT\_USERID en Lidia (sin que cambie nada más, es decir, el nombre de la aplicación cliente todavía es reporting.exe), no se produce ninguna reasignación de carga de trabajo, porque no se ha definido específicamente ninguna carga de trabajo con CURRENT CLIENT\_USERID establecido en Lidia.

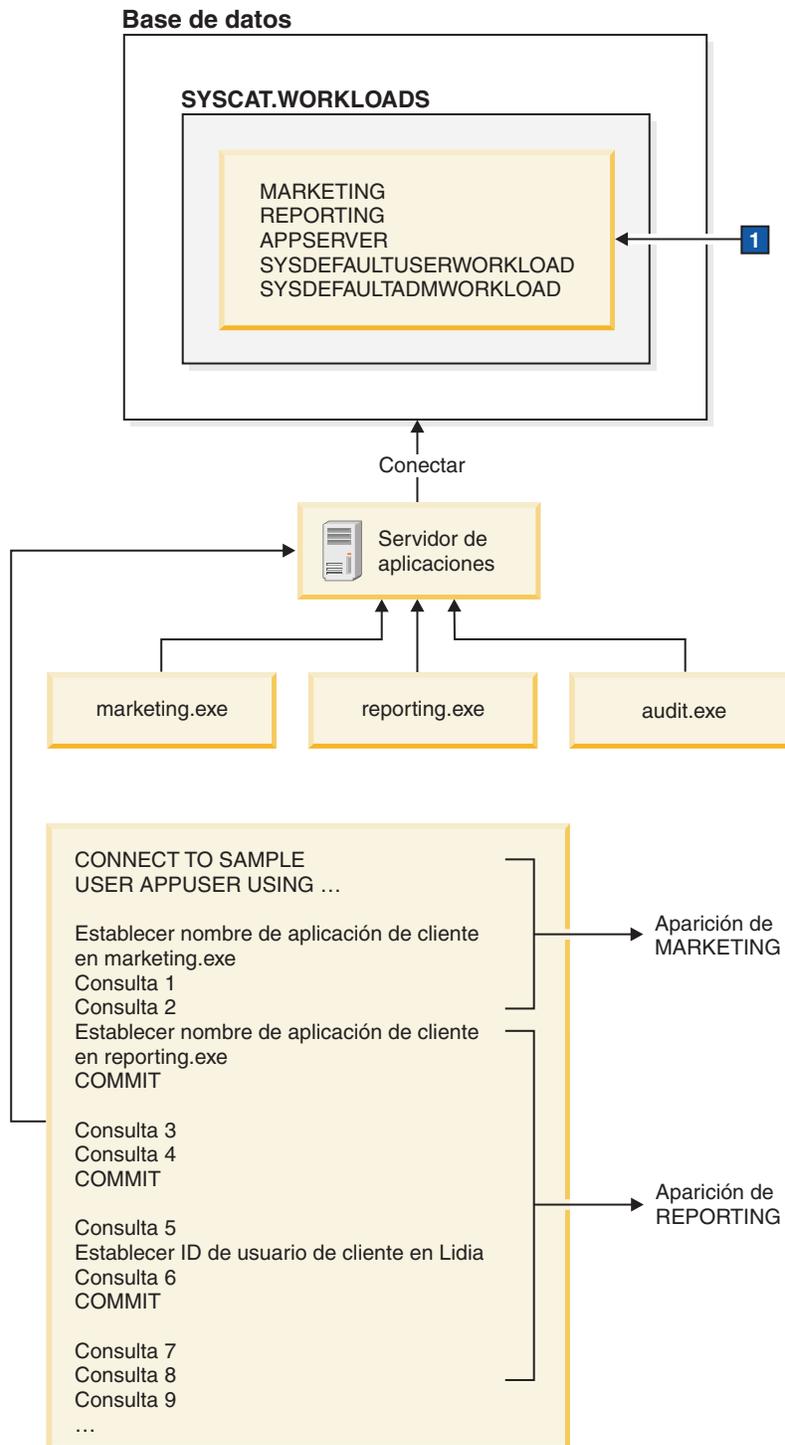


Figura 6. Ejemplo de asignación de cargas de trabajo en un entorno de tres niveles

Las sentencias siguientes se utilizan para definir las cargas de trabajo especificadas en el recuadro **1** de la figura anterior:

```
CREATE WORKLOAD MARKETING SESSION_USER ('APPUSER')
CURRENT CLIENT_APPLNAME ('marketing.exe') SERVICE CLASS SC2
POSITION AT 1
```

```
CREATE WORKLOAD REPORTING SESSION_USER ('APPUSER')
CURRENT CLIENT_APPLNAME ('reporting.exe') SERVICE CLASS SC4
```

POSITION AFTER MARKETING

```
CREATE WORKLOAD APPSERV SESSION_USER ('APPUSER')
SERVICE CLASS SC1
```

## Ejemplo: Asignación de cargas de trabajo cuando los atributos de la carga de trabajo sólo tienen un valor

En el ejemplo de este tema se muestra cómo realiza el servidor de datos una asignación de cargas de trabajo. En este ejemplo, sólo se especifica un valor por cada atributo de conexión de la carga de trabajo.

Supongamos que existen las siguientes cargas de trabajo en el catálogo:

Tabla 5. Cargas de trabajo en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	ADDRESS	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER_GROUP	SESSION_USER_ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
1	REPORTS		AppA								
2	INVENTORY REPORT		AppB	LYNN		ACCOUNTING	TELEMKTR				
3	SALES REPORT		AppC	KATE	KATE		SALESREP				
4	AUDIT REPORT		AppB			ACCOUNTING	FINANALYST				
5	EXPENSE REPORT		AppA	TIM			EXPENSE APPROVER				
6	AUDIT RESULT				LYNN			LYNN			Grupo de auditoría

Supongamos que se establece una conexión de base de datos con los siguientes atributos:

Tabla 6. Atributos de conexión de base de datos

ADDRESS	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER_GROUP	SESSION_USER_ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
9.26.53.111	AppA	TIM	TIM	FINANCE	FINANALYST, EXPENSE APPROVER	NULL	NULL	NULL	Cuenta empresarial

Cuando se envía la primera unidad de trabajo, el servidor de datos comprueba cada carga de trabajo del catálogo, empezando por la primera de la lista, y procesa las cargas de trabajo en orden ascendente hasta que encuentra una carga de trabajo con atributos coincidentes. Cuando se encuentra una carga de trabajo coincidente, la unidad de trabajo se ejecuta bajo una aparición de esa carga de trabajo. A la hora de determinar a qué carga de trabajo debe asignarse la conexión, el servidor de datos compara los atributos de conexión en orden determinista.

El servidor de datos comprueba primero si hay coincidencias en la carga de trabajo REPORTS. La carga de trabajo REPORTS es la primera de la lista.

Tabla 7. Carga de trabajo REPORTS en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	ADDRESS	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER_GROUP	SESSION_USER_ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
1	REPORTS		AppA								

El servidor de datos comprueba los atributos de conexión en el siguiente orden determinista:

1. APPLNAME. El valor de APPLNAME, AppA, para la conexión de base de datos coincide con el valor de APPLNAME para la carga de trabajo REPORTS.
2. SYSTEM\_USER, que no está establecido en la definición de la carga de trabajo. Cualquier valor (incluso un valor nulo) se considera una coincidencia.

3. SESSION\_USER, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.
4. SESSION\_USER GROUP, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.
5. SESSION\_USER ROLE, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.
6. CURRENT CLIENT\_USERID, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.
7. CURRENT CLIENT\_APPLNAME, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.
8. CURRENT CLIENT\_WRKSTNNAME, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.
9. CURRENT CLIENT\_ACCTNG, que no está establecido en la definición de la carga de trabajo. Cualquier valor se considera una coincidencia.

En esta situación, debido a las coincidencias explícitas e implícitas entre los atributos de conexión de la carga de trabajo REPORTS y la información proporcionada en la conexión, el servidor de datos selecciona la carga de trabajo REPORT como posible coincidencia. Después de seleccionar una carga de trabajo, el servidor de datos comprueba si el usuario de la sesión tiene el privilegio USAGE sobre la carga de trabajo. Suponiendo que el usuario de la sesión TIM tenga el privilegio USAGE sobre la carga de trabajo REPORTS, dicha carga de trabajo se utilizará para la conexión. Sin embargo, si TIM no posee el privilegio USAGE sobre la carga de trabajo REPORTS, el servidor de datos procederá a comprobar si existe alguna coincidencia en la carga de trabajo INVENTORYREPORT.

Supongamos que desea asignar a TIM la carga de trabajo EXPENSEREPORT, porque esa carga de trabajo tiene especificados atributos de conexión adicionales. En este caso, modificaría el orden de evaluación de las cargas de trabajo para colocar EXPENSEREPORT antes que REPORTS en la lista de cargas de trabajo:

```
ALTER WORKLOAD EXPENSEREPORT POSITION AT 1
```

También podría utilizar la siguiente sentencia de SQL para obtener el mismo resultado:

```
ALTER WORKLOAD EXPENSEREPORT BEFORE REPORTS
```

Para asegurarse de que la sentencia ALTER WORKLOAD se ejecute, debe emitir una sentencia COMMIT inmediatamente después de la sentencia ALTER WORKLOAD. El efecto de la sentencia ALTER WORKLOAD en el catálogo es el siguiente:

*Tabla 8. Cargas de trabajo en el catálogo después de recolocar la carga de trabajo EXPENSEREPORT*

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT CLIENT_USERID	CURRENT CLIENT_APPLNAME	CURRENT CLIENT_WRKSTNNAME	CURRENT CLIENT_ACCTNG
1	EXPENSE REPORT	AppA	TIM			EXPENSE APPROVER				
2	REPORTS	AppA								
3	INVENTORY REPORT	AppB	LYNN		ACCOUNTING	TELEMKTR				
4	SALES REPORT	AppC	KATE	KATE		SALESREP				
5	AUDIT REPORT	AppB			ACCOUNTING	FINANALYST				
6	AUDIT RESULT			LYNN			LYNN			Grupo de auditoría

Si TIM no tiene el privilegio USAGE sobre la carga de trabajo EXPENSEREPORT, debe emitir las sentencias siguientes (la sentencia COMMIT garantiza que la sentencia GRANT se ejecuta):

GRANT USAGE ON WORKLOAD EXPENSEREPORT TO USER TIM  
COMMIT

Al principio de la siguiente unidad de trabajo se produce la reasignación de la carga de trabajo, y el servidor de datos asigna la conexión de TIM a la carga de trabajo EXPENSEREPORT. Además, las nuevas unidades de trabajo que envíen otras conexiones que tengan los mismos atributos también se asociarán a la carga de trabajo EXPENSEREPORT.

## Ejemplo: Asignación de una carga de trabajo para una unidad de trabajo cuando existen varias cargas de trabajo

En el ejemplo de este tema se muestra cómo el servidor de datos realiza una evaluación de la carga de trabajo para asignar la conexión a una carga de trabajo existente.

Supongamos que se han definido las siguientes cargas de trabajo en el catálogo:

Tabla 9. Cargas de trabajo en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
1	EXPENSE REPORT	AppB	TIM			EXPENSE APPROVER				
2	REPORTS	AppB								
3	INVENTORYREPORT	AppA	LYNN		ACCOUNTING	TELEMKTR				
4	SALES REPORT	AppC	KATE	KATE		SALESREP				
5	AUDIT REPORT	AppA			ACCOUNTING	FINANALYST				
6	AUDIT RESULT			LYNN			LYNN			Grupo de auditoría

Supongamos que se establece una conexión de base de datos con los siguientes atributos:

Tabla 10. Atributos de conexión de base de datos

APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
AppA	LYNN	LYNN	ACCOUNTING	FINANALYST, SALESREP	LYNN	NULL	wrkstn2	Grupo de auditoría

Cuando se envía la primera unidad de trabajo, el servidor de datos comprueba cada carga de trabajo del catálogo en orden de evaluación ascendente, y se detiene cuando encuentra una carga de trabajo cuyos atributos de conexión coinciden con los suministrados por la conexión. Al comprobar las cargas de trabajo, el servidor de datos compara los atributos de conexión en orden determinista.

Primero, el servidor de datos comprueba la carga de trabajo EXPENSEREPORT:

Tabla 11. Carga de trabajo EXPENSEREPORT en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
1	EXPENSEREPORT	AppB	TIM			EXPENSE APPROVER				

Puesto que el atributo APPLNAME en la definición de carga de trabajo es AppB pero el atributo APPLNAME proporcionado por la conexión es AppA, la coincidencia no es posible. El servidor de datos pasa a la carga de trabajo REPORTS, que es la segunda de la lista:

Tabla 12. Carga de trabajo REPORTS en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACTNG
2	REPORTS	AppB								

Nuevamente, el atributo APPLNAME en la definición de carga de trabajo es AppB, lo que no coincide con AppA. El servidor de datos continúa con la tercera carga de trabajo de la lista, INVENTORYREPORT:

Tabla 13. Carga de trabajo INVENTORYREPORT en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACTNG
3	INVENTORYREPORT	AppA	LYNN		ACCOUNTING	TELEMKTR				

El servidor de datos comprueba si hay una coincidencia entre los atributos de conexión enviados y la carga de trabajo INVENTORYREPORT. Los atributos se comprueban en el orden siguiente:

1. APPLNAME. Tanto la definición de carga de trabajo como la conexión tienen el valor AppA, de forma que existe una coincidencia.
2. SYSTEM\_USER. Tanto la definición de carga de trabajo como la conexión tienen el valor LYNN, de forma que existe una coincidencia.
3. SESSION\_USER. La conexión ha proporcionado un valor de LYNN. Puesto que no se ha establecido el atributo SESSION\_USER para la carga de trabajo, cualquier valor, incluso un valor nulo, que proporcione la conexión dará como resultado una coincidencia.
4. SESSION\_USER GROUP. Tanto la definición de carga de trabajo como la conexión tienen el valor ACCOUNTING, de forma que existe una coincidencia.
5. SESSION\_USER ROLE. La definición de carga de trabajo especifica el valor de TELEMKTR, pero la conexión ha proporcionado los valores de FINANALYST y SALESREP. No se producen coincidencias para este atributo.

El servidor de datos detiene el intento de hacer coincidir la carga de trabajo INVENTORYREPORT con los atributos de la conexión y continúa con la cuarta carga de trabajo de la lista, SALESREPORT:

Tabla 14. Carga de trabajo SALESREPORT en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACTNG
4	SALESREPORT	AppC	KATE	KATE		SALESREP				

Puesto que el APPLNAME en la definición de la carga de trabajo SALESREPORT es AppC, no existe coincidencia con la conexión (que ha proporcionado el valor AppA para APPLNAME). A continuación, el servidor de datos continúa con la quinta carga de trabajo de la lista, AUDITREPORT:

Tabla 15. Carga de trabajo AUDITREPORT en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACTNG
5	AUDITREPORT	AppA			ACCOUNTING	FINANALYST				

El servidor de datos compara los atributos de la carga de trabajo AUDITREPORT y la conexión en orden determinista:

1. APPLNAME. Tanto la definición de carga de trabajo como la conexión tienen el valor AppA, de forma que existe una coincidencia.

2. SYSTEM\_USER. La conexión ha proporcionado un valor de LYNN. Puesto que no se ha establecido el atributo SYSTEM\_USER para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
3. SESSION\_USER. La conexión ha proporcionado un valor de LYNN. Puesto que no se ha establecido el atributo SESSION\_USER para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
4. SESSION\_USER GROUP. Tanto la carga de trabajo como la conexión tienen el valor ACCOUNTING para este atributo, de forma que existe una coincidencia.
5. SESSION\_USER ROLE. Tanto la carga de trabajo como la conexión tienen el valor FINANALYST para este atributo, de forma que existe una coincidencia.
6. CURRENT\_CLIENT\_USERID. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_USERID para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
7. CURRENT\_CLIENT\_APPLNAME. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_APPLNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
8. CURRENT\_CLIENT\_WRKSTNNAME. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_WRKSTNNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
9. CURRENT\_CLIENT\_ACCTNG. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_ACCTNG para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.

Después de procesar todos los atributos de conexión y de encontrar una carga de trabajo coincidente, el servidor de datos comprueba si el usuario de la sesión tiene privilegios USAGE sobre la carga de trabajo. Supongamos que LYNN no tiene privilegios USAGE sobre la carga de trabajo AUDITREPORT. En este caso, aunque todos los atributos de la conexión coinciden, esta carga de trabajo no se asocia a la conexión. El servidor de datos continúa con la sexta carga de trabajo de la lista de evaluación, AUDITRESULT:

Tabla 16. Carga de trabajo AUDITRESULT en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
6	AUDITRESULT			LYNN			LYNN			Grupo de auditoría

El servidor de datos compara los atributos de la carga de trabajo AUDITRESULT y la conexión en orden determinista:

1. APPLNAME. Puesto que no se ha establecido el atributo APPLNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
2. SYSTEM\_USER. Puesto que no se ha establecido el atributo SYSTEM\_USER para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
3. SESSION\_USER. Tanto la carga de trabajo como la conexión tienen el valor LYNN para este atributo, de forma que existe una coincidencia.
4. SESSION\_USER GROUP. Puesto que no se ha establecido el atributo SESSION\_USER GROUP para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.

5. SESSION\_USER ROLE. Puesto que no se ha establecido el atributo SESSION\_USER ROLE para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
6. CURRENT\_CLIENT\_USERID. Tanto la carga de trabajo como la conexión tienen el valor LYNN para este atributo, de forma que existe una coincidencia.
7. CURRENT\_CLIENT\_APPLNAME. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_APPLNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
8. CURRENT\_CLIENT\_WRKSTNNAME. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_WRKSTNNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
9. CURRENT\_CLIENT\_ACCTNG. Tanto la carga de trabajo como la conexión tienen el valor Grupo de auditoría para este atributo, de forma que existe una coincidencia.

Después de procesar todos los atributos de conexión y de encontrar una carga de trabajo coincidente, el servidor de datos comprueba si el usuario de la sesión tiene privilegios USAGE sobre la carga de trabajo. En este caso, supongamos que el usuario de la sesión LYNN tiene privilegios USAGE sobre la carga de trabajo AUDITRESULT. Puesto que todos los atributos de la conexión coinciden y el usuario de la sesión tiene privilegios USAGE, la conexión se asigna a la carga de trabajo AUDITRESULT.

## Ejemplo: Asignación de una carga de trabajo cuando los atributos de la carga de trabajo tienen varios valores

En el ejemplo de este tema se muestra cómo realiza el servidor de datos una asignación de cargas de trabajo. En este ejemplo, algunas de las definiciones de la carga de trabajo permiten más de un valor para un atributo de conexión.

Supongamos que se han definido las siguientes cargas de trabajo en el catálogo:

Tabla 17. Cargas de trabajo en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
1	ITEMINQ		KYLE, GEORGE		RETAIL, SALES					
2	DAILY TRANS REPORT	AppC		KYLE, CAROL	SALES, ACCOUNTING					
3	SALES SUMMARY	AppA, AppB				ACCOUNTANT, FINANALYST				

Supongamos que se establece una conexión de base de datos con los siguientes atributos:

Tabla 18. Atributos de conexión de base de datos

APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
AppC	LINDA	KYLE	SALES	ACCOUNTANT	LINDA	NULL	NULL	Cuenta empresarial

Cuando se envía la primera unidad de trabajo, el servidor de datos comprueba cada carga de trabajo del catálogo en orden de evaluación ascendente, y se detiene cuando encuentra una carga de trabajo cuyos atributos de conexión coinciden con los suministrados por la conexión. Al comprobar las cargas de trabajo, el servidor de datos compara los atributos de conexión en orden determinista.

Primero, el servidor de datos comprueba la carga de trabajo ITEMINQ:

Tabla 19. Carga de trabajo ITEMINQ en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
1	ITEMINQ		KYLE, GEORGE		RETAIL, SALES					

El servidor de datos comprueba si hay una coincidencia entre los atributos de conexión enviados y la carga de trabajo ITEMINQ. Los atributos se comprueban en el orden siguiente:

1. APPLNAME. Puesto que no se ha establecido el atributo APPLNAME para la carga de trabajo, cualquier valor, incluso un valor nulo, que proporcione la conexión dará como resultado una coincidencia.
2. SYSTEM\_USER. La conexión ha proporcionado un valor de LINDA. Sin embargo, los valores de la carga de trabajo ITEMNO son KYLE y GEORGE. No se producen coincidencias para este atributo.

El servidor de datos detiene el intento de hacer coincidir la carga de trabajo ITEMNO con la conexión y continúa en la segunda carga de trabajo de la lista, DAILYTRANSREPORT:

Tabla 20. Carga de trabajo DAILYTRANSREPORT en el catálogo

Orden de evaluación	Nombre de la carga de trabajo	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
2	DAILYTRANSREPORT	AppC		KYLE, CAROL	SALES, ACCOUNTING					

El servidor de datos compara los atributos de la carga de trabajo DAILYTRANSREPORT y la conexión en orden determinista:

1. APPLNAME. Tanto la definición de carga de trabajo como la conexión tienen el valor AppC, de forma que existe una coincidencia.
2. SYSTEM\_USER. Puesto que no se ha establecido el atributo SYSTEM\_USER para la carga de trabajo, cualquier valor, incluso un valor nulo, que proporcione la conexión dará como resultado una coincidencia.
3. SESSION\_USER. El valor SESSION\_USER proporcionado en la conexión es KYLE, que coincide con uno de los valores SESSION\_USER de la carga de trabajo. Si la conexión hubiera proporcionado CAROL, también se daría una coincidencia, porque se ha especificado tanto KYLE como CAROL como parte de la definición de la carga de trabajo DAILYTRANSREPORT.
4. SESSION\_USER GROUP. El valor de SESSION\_USER GROUP proporcionado en la conexión es SALES, que coincide con el valor de SALES especificado para el atributo SESSION\_USER GROUP de la carga de trabajo. Si la conexión hubiera proporcionado ACCOUNTING, también se daría una coincidencia, puesto que se ha especificado tanto SALES como ACCOUNTING en la definición de la carga de trabajo.
5. SESSION\_USER ROLE. Puesto que no se ha establecido el atributo SESSION\_USER ROLE para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
6. CURRENT\_CLIENT\_USERID. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_USERID para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
7. CURRENT\_CLIENT\_APPLNAME. Puesto que no se ha establecido el atributo CURRENT\_CLIENT\_APPLNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.

8. CURRENT CLIENT\_WRKSTNNAME. Puesto que no se ha establecido el atributo CURRENT CLIENT\_WRKSTNNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.
9. CURRENT CLIENT\_ACCTNG. Puesto que no se ha establecido el atributo CURRENT CLIENT\_WRKSTNNAME para la carga de trabajo, cualquier valor que proporcione la conexión dará como resultado una coincidencia.

Después de procesar todos los atributos de conexión y de encontrar una carga de trabajo coincidente para la conexión, el servidor de datos comprueba si el usuario de la sesión tiene privilegios USAGE sobre la carga de trabajo. En este caso, supongamos que el usuario de la sesión KYLE tiene privilegios USAGE sobre la carga de trabajo DAILYTRANSREPORT. Puesto que todos los atributos de la conexión coinciden y el usuario de la sesión tiene privilegios USAGE, la conexión se asigna a la carga de trabajo DAILYTRANSREPORT.

---

## Identificación de trabajos por tipo, por coste o por medio de los datos a los que se accede con clases de trabajo

Además de utilizar atributos de conexión que se centran en el origen de las actividades con cargas de trabajo, puede identificar las actividades basándose en el tipo, el coste o los datos a los que podría accederse mediante la creación de un conjunto de clases de trabajo que contenga una clase de trabajo.

Una clase de trabajo es un método para clasificar las diferentes actividades de la base de datos en función de sus atributos. Si una clase de trabajo tiene definida una acción de trabajo, la acción de trabajo se aplicará a la clase de trabajo y determinará cómo han de gestionarse las actividades de la clase de trabajo. Para obtener más información, consulte “Aplicación de controles a tipos de actividades con conjuntos de acciones de trabajo” en la página 102.

En la siguiente tabla se muestran las palabras clave de tipo que están disponibles para las clases de trabajo y las sentencias de SQL que corresponden a las diferentes palabras clave. A excepción del programa de utilidad de carga, todas las sentencias de la tabla siguiente se interceptan inmediatamente antes de la ejecución en el proceso de una petición EXECUTE, EXECUTE IMMEDIATE u OPEN. El programa de utilidad de carga, cuando se emite desde un cliente, podría emitir peticiones antes de iniciar la operación de carga real en el servidor de datos.

Tabla 21. Tipos de trabajo

Palabra clave de tipo de trabajo	Sentencias de SQL que se aplican
READ, incluidas las sentencias SET con SQL READ incorporado	<ul style="list-style-type: none"> <li>• Todas las sentencias SELECT (select into, values into, full select)</li> <li><b>Excepción:</b> No se incluyen las sentencias SELECT que contienen una sentencia DELETE, INSERT o UPDATE.</li> <li>• Todas las sentencias de XQuery</li> </ul>

Tabla 21. Tipos de trabajo (continuación)

Palabra clave de tipo de trabajo	Sentencias de SQL que se aplican
WRITE, incluidas las sentencias SET con SQL WRITE incorporado	<ul style="list-style-type: none"> <li>• Todas las sentencias UPDATE (buscadas, posicionadas)</li> <li>• Todas las sentencias DELETE (buscadas, posicionadas)</li> <li>• Todas las sentencias INSERT (valores, subselección)</li> <li>• Todas las sentencias MERGE</li> <li>• Todas las sentencias SELECT que contienen una sentencia DELETE, INSERT o UPDATE</li> </ul>
CALL	<p>CALL, sentencia</p> <p>La sentencia CALL sólo se clasifica bajo los tipos de clase de trabajo CALL y ALL.  <b>Nota:</b> Los bloques anónimos y las rutinas anónimas se clasifican como sentencias CALL.</p>
DML, incluidas las sentencias SET con SQL READ o WRITE incorporado	<p>Todas las sentencias que se clasifican bajo los tipos de clase de trabajo READ y WRITE.</p>
DDL	<ul style="list-style-type: none"> <li>• Todas las sentencias ALTER</li> <li>• Todas las sentencias CREATE</li> <li>• Sentencia COMMENT</li> <li>• Sentencia DECLARE GLOBAL TEMPORARY TABLE</li> <li>• Sentencia DROP</li> <li>• Sentencia FLUSH PACKAGE CACHE</li> <li>• Todas las sentencias GRANT</li> <li>• REFRESH TABLE</li> <li>• Todas las sentencias RENAME</li> <li>• Todas las sentencias REVOKE</li> <li>• Sentencia SET INTEGRITY</li> </ul>
LOAD	<p>Programa de utilidad de carga</p> <p>El programa de utilidad de carga sólo se clasifica bajo los tipos de clase de trabajo LOAD y ALL.</p>
ALL	<p>Tipos de trabajo que están representados por todas las palabras clave anteriores.</p> <p><b>Nota:</b> Si la acción es un umbral, la actividad de base de datos a la que se aplica el umbral depende del tipo de umbral. Por ejemplo, si el tipo de umbral es ESTIMATEDSQLCOST, el umbral sólo afectará a la actividad de DML con un coste estimado (en timerons).</p> <p>Para obtener más información, consulte “Ejemplo: Cómo trabajar con una clase de trabajo definida con la palabra clave ALL” en la página 68.</p>

En la siguiente figura es muestra una vista jerárquica de las palabras clave de tipo de trabajo:

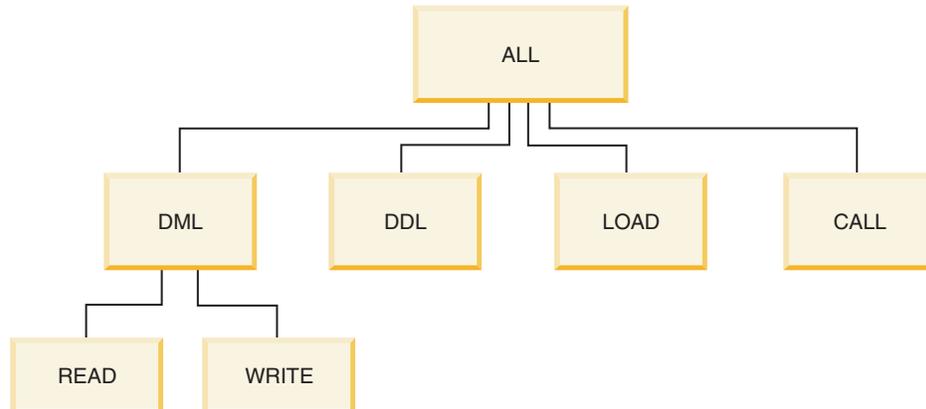


Figura 7. Palabras clave de tipo de trabajo

Las sentencias de SQL que no aparecen bajo ninguna de las palabras clave disponibles no están clasificadas y se comportan como si no existiera ninguna clase de trabajo y ningún conjunto de clases de trabajo. Por ejemplo, si la sentencia es SET SCHEMA y la única clase de trabajo del conjunto de clases de trabajo tiene un tipo de trabajo DML, esa sentencia no se clasifica y no puede aplicarse ninguna acción de trabajo a ésta. Por lo tanto, si la acción es MAP, la actividad SET SCHEMA se ejecuta en la subclase de servicio por omisión (SYSDEFAULTSUBCLASS). Si la acción es un umbral, no se aplica ningún umbral a la actividad.

## Identificación adicional

Las clases de trabajo también le permiten utilizar elementos predictivos en la identificación para el trabajo DML (o sentencias READ y WRITE). Los elementos predictivos son útiles porque proporcionan información acerca de las actividades de base de datos y pueden utilizarse para realizar una acción antes de que estas actividades empiecen a consumir recursos en el servidor de datos. En la siguiente tabla se proporciona información acerca de los elementos predictivos que las clases de servicio soportan:

Tabla 22. Características para la identificación predictiva

Elemento predictivo	Descripción
Coste estimado	Utiliza el coste estimado disponible del compilador de DB2 para incluir DML dentro de un rango de timerons determinado (por ejemplo, crear una clase de trabajo para todas las consultas grandes con un coste estimado de más de 1.000.000 timerons).
Cardinalidad estimada	Utiliza las filas estimadas devueltas (cardinalidad) del compilador de DB2 para incluir DML dentro de un rango de filas devueltas determinado (por ejemplo, crear una clase de trabajo para consultas grandes que se estima que devolverán más de 500.000 filas).

Tabla 22. Características para la identificación predictiva (continuación)

Elemento predictivo	Descripción
Estimación de los datos a los que se accede	<p>Utiliza la estimación de los datos a los que se accede desde el compilador de DB2 para incluir el DML que accede a los datos asignados a un código de datos específico (por ejemplo, crear una clase de trabajo para consultas que se estima que accederán a los datos de los espacios de tablas que se han codificado con un valor de código de datos 3).</p> <p><b>Nota:</b> El acceso al conjunto de espacios de tablas que el compilador de DB2 predecirá se basa en la sentencia de SQL optimizado, que podría diferir de la sentencia de SQL especificada por el usuario. Por ejemplo, en algunos casos, al insertar tablas particionadas por rango, el número de espacios de tablas cuyo acceso se predice es superior a los espacios de tablas esperados.</p>

También puede identificar actividades utilizando el nombre de esquema del procedimiento que una sentencia CALL llama.

Basándose en los atributos de carga de trabajo y los tipos de clase de trabajo, puede identificar trabajo y preparar éste para la siguiente etapa, la gestión del trabajo.

Para obtener más información acerca de cómo trabajar con las clases de trabajo y los conjuntos de clases de trabajo, consulte los temas siguientes:

## Clases de trabajo y conjuntos de clases de trabajo

Una clase de trabajo es un método para clasificar las diferentes actividades de la base de datos en función de sus atributos. Las clases de trabajo se agrupan en conjuntos de clases de trabajo, que diferentes conjuntos de acciones de trabajo pueden compartir.

Algunos ejemplos de atributos de actividades de base de datos que pueden determinar la clase de trabajo a la que se asocia una actividad son: tipo de actividad (DDL, DML, LOAD), coste estimado (si está disponible), cardinalidad estimada (si está disponible) y esquema (si está disponible).

### Clases de trabajo

Una clase de trabajo tiene los atributos siguientes:

- El nombre de la clase de trabajo, que debe ser exclusivo en el conjunto de clases de trabajo.
- Los atributos de actividad de la base de datos, que contienen la información siguiente:
  - El tipo de actividad de la base de datos que coincide con esta clase de trabajo. Mediante el uso de palabras clave predefinidas (por ejemplo, CALL, READ, WRITE, DML, DDL, LOAD o ALL), puede clasificar las peticiones de la base de datos en diferentes categorías. Los diferentes tipos de actividades de la base de datos pueden asociarse a una clase de trabajo en función de su tipo de trabajo. Por ejemplo, la palabra clave WRITE incluye actualizaciones, supresiones, inserciones, fusiones y selecciones se contienen una supresión, inserción o actualización. Para obtener más información, consulte

“Identificación de trabajos por tipo, por coste o por medio de los datos a los que se accede con clases de trabajo” en la página 52.

- Información de rango que clasifica con más detalle los tipos de actividad de base de datos DML o XQuery.
  - El tipo de rango que debe especificarse (coste en timerons o cardinalidad). La especificación de un rango de valores es opcional. Por ejemplo, cuando especifica un rango para una clase de trabajo, puede especificar que todas las consultas con un coste estimado de menos de 100 timerons puedan procesarse de forma diferente de las demás consultas.
  - La parte inferior del rango.
  - La parte superior del rango.
- El esquema de la rutina a la que debe llamarse. La especificación del esquema es opcional. Cuando defina una clase de trabajo, puede utilizar el atributo de esquema para clasificar con más detalle las sentencias CALL de acuerdo con el esquema del procedimiento al que se llama. Por ejemplo, si especifica SCHEMA1 como esquema de una clase de trabajo y el tipo de trabajo es CALL, todas las sentencias CALL que llamen al procedimiento SCHEMA1 se clasificarán en esa clase de trabajo. Si especifica el esquema para un tipo de clase de trabajo distinto de CALL o ALL, se devolverá el error SQL0628N.
- El código de identificación proporcionado a los datos con los que la actividad podría estar relacionada. Por ejemplo, cuando especifica un código de datos 3 para una clase de trabajo, las actividades que están relacionadas con los datos del espacio de tablas o del grupo de almacenamiento que tiene un código de datos 3 pueden aislarse y gestionarse de forma distinta.
- El orden de evaluación de la clase de trabajo (o posición de la clase de trabajo en el conjunto de clases de trabajo). Para obtener más información, consulte “Orden de evaluación de clases de trabajo en un conjunto de clases de trabajo” en la página 63.
- Un identificador de clase generado automáticamente que identifica de forma exclusiva la clase de trabajo.

Puede crear clases de trabajo de dos formas:

- Cree un nuevo conjunto de clases de trabajo que contendrá la nueva clase de trabajo mediante la palabra clave WORK CLASS de la sentencia CREATE WORK CLASS SET.
- Añada la nueva clase de trabajo a un conjunto de clases de trabajo existente mediante la palabra clave ADD de la sentencia ALTER WORK.

Puede modificar clases de trabajo mediante la palabra clave ALTER WORK CLASS de la sentencia ALTER WORK CLASS SET.

Puede descartar clases de trabajo de un conjunto de clases de trabajo mediante la palabra clave DROP WORK CLASS de la sentencia ALTER WORK CLASS SET o mediante la sentencia DROP WORK CLASS SET para descartar el conjunto de clases de trabajo.

Para visualizar sus clases de trabajo, puede consultar la vista SYSCAT.WORKCLASSES.

## Conjuntos de clases de trabajo

Puede utilizar conjuntos de clases de trabajo para agrupar una o varias clases de trabajo. Un conjunto de clases de trabajo se compone de los siguientes atributos:

- Un nombre descriptivo exclusivo para el conjunto de clases de trabajo
- Los comentarios que desee proporcionar al conjunto de clases de trabajo
- Cero o más clases de trabajo (si bien una clase de trabajo sólo puede existir en un conjunto de clases de trabajo, un conjunto de clases de trabajo no tiene que contener necesariamente una clase de trabajo).
- Un ID generado automáticamente que identifica de forma exclusiva el conjunto de clases de trabajo.

Puede crear un nuevo conjunto de clases de trabajo utilizando la sentencia `CREATE WORK CLASS SET`. Puede crear un conjunto de clases de trabajo vacío y añadirle clases de trabajo más tarde, o puede crear un conjunto de clases de trabajo que contenga una o varias clases de trabajo.

Puede cambiar un conjunto de clases de trabajo existente de los modos siguientes utilizando la sentencia `ALTER WORK CLASS SET`:

- Añada clases de trabajo al conjunto de clases de trabajo.
- Cambie atributos de clase de trabajo para las clases de trabajo del conjunto de clases de trabajo.
- Descarte clases de trabajo del conjunto de clases de trabajo.

No puede cambiar ningún atributo del conjunto de clases de trabajo.

Para descartar un conjunto de clases de trabajo, utilice la sentencia `DROP WORK CLASS SET`.

Para visualizar sus conjuntos de clases de trabajo, consulte la vista de catálogo `SYSCAT.WORKCLASSETS`.

En la figura siguiente se muestra un ejemplo de clases de trabajo en un conjunto de clases de trabajo.

#### Conjunto de clases de trabajo: Actividades grandes

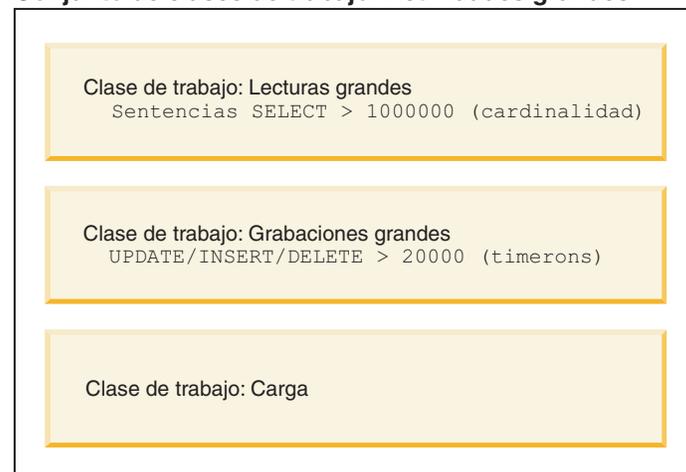


Figura 8. Ejemplo de clases de trabajo y un conjunto de clases de trabajo

Para que un conjunto de clases de trabajo sea efectivo en el sistema, debe definir un conjunto de acciones de trabajo y asociarlo al conjunto de clases de trabajo. Al utilizar un conjunto de acciones de trabajo, puede asociar un conjunto de clases de trabajo a una superclase de servicio, una carga de trabajo o una base de datos, para indicar la acción que debería aplicarse a las actividades de la base de datos que

coincidan con la clasificación. Si no crea un conjunto de acciones de trabajo para el conjunto de clases de trabajo, el servidor de datos pasará por alto el conjunto de clases de trabajo.

## Creación de una clase de trabajo

Para crear una clase de trabajo, utilice la sentencia `CREATE WORK CLASS SET` o la sentencia `ALTER WORK CLASS SET`.

### Antes de empezar

Para crear una clase de trabajo, necesita la autorización `WLMADM` o `DBADM`.

Si necesita conocer los requisitos adicionales, consulte los temas siguientes:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Normas de denominación

### Procedimiento

Para crear una clase de trabajo:

1. Cree una clase de trabajo en el mismo momento en que crea un nuevo conjunto de clases de trabajo o añada la nueva clase de trabajo a un conjunto de clases de trabajo que ya exista:
  - Para crear una nueva clase de trabajo que se añadirá a un nuevo conjunto de clases de trabajo, utilice la palabra clave `WORK CLASS` de la sentencia `CREATE WORK CLASS SET`.
  - Para crear una nueva clase de trabajo que se añadirá a un conjunto de clases de trabajo ya existente, utilice la palabra clave `ADD WORK CLASS` de la sentencia `ALTER WORK CLASS SET`.

Especifique una o más de las propiedades siguientes para la nueva clase de trabajo:

- Un nombre para la clase de trabajo. Este nombre debe ser exclusivo en el conjunto de clases de trabajo.
- Atributos para la clase de trabajo. Estos atributos se utilizan para asociar una actividad con la clase de trabajo:
  - El tipo de trabajo para el que la clase de trabajo se va a utilizar. Utilice el parámetro `WORK TYPE` para especificar esta característica.
  - `READ`, que representa actividades `SELECT` que no se actualizan, y todas las actividades `XQuery`. Cuando especifica la palabra clave `READ`, también puede especificar un argumento opcional de cláusula-para-desde-a o cláusula-código-datos.
- Utilice el argumento cláusula-para-desde-a opcional para especificar un rango para el coste de la sentencia en timerons o bien para su cardinalidad (es decir, el número de filas que se devuelven). Debe especificar un valor numérico para el primer valor. Para el segundo, puede especificar un valor numérico, o el valor `UNBOUNDED` para indicar que no desea imponer un límite superior ni en el coste ni en la cardinalidad de la actividad. También puede especificar este argumento para las palabras clave `WRITE`, `DML` y `ALL`.

Por ejemplo, para asociar las actividades `SELECT` que tienen un coste de 5000 timerons o más con esta clase de trabajo, debería especificar:

```
WORK TYPE READ FOR TIMERONCOST FROM 5000 TO UNBOUNDED
```

- Utilice el argumento cláusula-código-datos opcional para especificar un código de datos que identifique los datos a los que se estima que la actividad accederá. Puede especificar un valor comprendido entre el 1 y el 9. Si el argumento cláusula-código-datos no se especifica, no se impondrá ninguna restricción relacionada con el tipo de datos a los que una consulta accede. También puede especificar este argumento para las palabras clave WRITE, DML y ALL.

Por ejemplo, para asociar actividades SELECT a esta clase de trabajo, que se estima que accederán a los datos de los espacios de tablas a los que se ha asignado el valor de código de datos 1, especificaría:

```
WORK TYPE READ DATA TAG LIST CONTAINS 1
```

- WRITE, que representa las actividades SQL que actualizan los datos en la base de datos. Cuando especifica la palabra clave WRITE, también puede especificar un argumento opcional de cláusula-para-desde-a o cláusula-código-datos.

Por ejemplo, para asociar todas las actividades de grabación de datos a esta clase de trabajo que, en tiempo de compilación, se estima que estarán relacionadas con los datos codificados con el valor 5, especificaría:

```
WORK TYPE WRITE FOR CARDINALITY FROM 50 TO 100
```

- CALL, que representa las actividades CALL.

Cuando se especifica la palabra clave CALL, también puede especificar la palabra clave ROUTINES IN SCHEMA para indicar que sólo se asociarán las actividades CALL a las rutinas de un esquema específico con esta clase de trabajo. Por ejemplo, si sólo desea asociar llamadas a rutinas en el esquema ACCOUNTS con esta clase de trabajo, deberá especificar:

```
WORK TYPE CALL ROUTINES IN SCHEMA ACCOUNTS
```

- DML, que representa las actividades SQL que cubren las palabras clave READ y WRITE.

Por ejemplo, para asociar todas las actividades de DML que tienen un coste estimado en timerons de 500 a 1000 y que se estima que, en tiempo de compilación, estarán relacionadas con los datos codificados con el valor 8, especificaría:

```
WORK TYPE DML FOR TIMERONCOST FROM 500 TO 1000 DATA TAG LIST CONTAINS 8
```

- DDL, que representa las actividades siguientes:

- ALTER
- CREATE
- COMMENT
- DECLARE GLOBAL TEMPORARY TABLE
- DROP
- FLUSH PACKAGE CACHE
- GRANT
- REFRESH TABLE
- RENAME
- REVOKE
- SET INTEGRITY

Por ejemplo, para asociar todas las actividades DDL con esta clase de trabajo, debería especificar:

```
WORK TYPE DDL
```

- LOAD, que representa una actividad LOAD.

Por ejemplo, para asociar las actividades LOAD con esta clase de trabajo, debería especificar:

```
WORK TYPE LOAD
```

- ALL, que representa todos los tipos de trabajo indicados por todas las palabras clave anteriores.

Cuando se especifica ALL para un tipo de clase de trabajo, también puede especificar la palabra clave ROUTINES IN SCHEMA para indicar que sólo se asociarán las actividades CALL a las rutinas de un esquema específico con esta clase de trabajo. También puede especificar el argumento cláusula-para-desde-a para indicar que todas las actividades de DML que tienen un coste estimado en timerons o una cardinalidad específica se engloban en esta clase. Por ejemplo, para asociar las actividades de DML que tienen una cardinalidad de 300 a 1500 filas y las rutinas que se llaman desde el esquema NEWHIRES para esta clase de trabajo, especificaría la sentencia siguiente. También puede especificar el argumento de cláusula-código-datos para indicar todas las actividades de DML que acceden a los datos de los espacios de tablas que se han codificado con un valor de código de datos. Puesto que esta clase de trabajo tiene un tipo ALL, también se aplicará a otras actividades que no tienen un esquema o cardinalidad, como las actividades LOAD y las actividades DDL.

```
WORK TYPE ALL FOR CARDINALITY FROM 300 TO 1500 ROUTINES
IN SCHEMA NEWHIRES
```

- Opcional. La posición de la clase de trabajo en el conjunto de clases de trabajo. La posición de la clase de trabajo en el conjunto de clases de trabajo determina el orden en el que la clase de trabajo se evalúa cuando se clasifica una actividad en una clase de trabajo. Cuando se produce una asignación de clase de trabajo, en primer lugar el servidor de datos determina el conjunto de clases de trabajo asociado con el objeto (ya sea una superclase de servicio o la base de datos), y luego selecciona la primera clase de trabajo coincidente en el conjunto de clases de trabajo que tenga una acción de trabajo asociada. Utilice la palabra clave POSITION para especificar una de estas opciones:

- LAST. La clase de trabajo se coloca al final de la lista de clases de trabajo en el conjunto de clases de trabajo. Por ejemplo:

```
WORK TYPE ... POSITION LAST
```

- BEFORE *nombre\_clase\_trabajo*. La clase de trabajo se crea en el conjunto de clases de trabajo y se coloca delante de la clase de trabajo especificada. Por ejemplo:

```
WORK TYPE ... POSITION BEFORE LARGEDDL
```

- AFTER *nombre\_clase\_trabajo*. La clase de trabajo se crea en el conjunto de clases de trabajo y se coloca detrás de la clase de trabajo especificada. Por ejemplo:

```
WORK TYPE ... POSITION AFTER LARGEDDL
```

- AT *entero*. La clase de trabajo se crea en el conjunto de clases de trabajo y se coloca en la posición especificada por el valor del entero. Por ejemplo:

```
WORK TYPE ... POSITION AT 3
```

2. Confirme los cambios. Cuando confirma los cambios, la clase de trabajo se añade a la vista SYSCAT.WORKCLASSES.

## Alteración de una clase de trabajo

Si necesita alterar una clase de trabajo, utilice la sentencia ALTER WORK CLASS SET.

### Antes de empezar

Para alterar una clase de trabajo, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para conocer los requisitos adicionales.

### Procedimiento

Para alterar una clase de trabajo:

1. Utilice la palabra clave ALTER de la sentencia ALTER WORK CLASS SET para cambiar una o más de las propiedades siguientes. Consulte “Creación de una clase de trabajo” en la página 58 para obtener una explicación de los valores que se admiten para estas propiedades.
  - La palabra clave FOR. Por ejemplo, puede cambiar el valor especificado para la palabra clave FOR de CARDINALITY a TIMERONCOST.
  - El argumento FROM *desde\_valor* TO *a\_valor*. Por ejemplo, puede cambiar el argumento de FROM 50 TO 100 a FROM 500 TO 1500.
  - Las palabras clave ROUTINES IN SCHEMA o ROUTINES IN ALL para las actividades CALL. Por ejemplo, si la clase de trabajo no especifica actualmente un esquema, puede añadir uno. También puede especificar la palabra clave ALL, de manera que la clase de trabajo se aplique a todas las sentencias CALL independientemente del esquema de la rutina. ALL es el valor por omisión.
  - La palabra clave DATA TAG LIST CONTAINS. Por ejemplo, puede cambiar el valor especificado para esta palabra clave y, en lugar de ANY, especificar 8.
  - La palabra clave POSITION, seguida por las palabras clave LAST, BEFORE, AFTER o AT. Si especifica POSITION BEFORE o POSITION AFTER, también deberá especificar la clase de trabajo que desea utilizar para colocar la clase de trabajo alterada. Si especifica POSITION AT, deberá incluir el número de posición. Por ejemplo, puede mover una clase de trabajo desde la última posición a cualquier otra posición utilizando la palabra clave AT, o desde cualquier posición a la última posición utilizando la palabra clave LAST.
2. Confirme los cambios. Cuando confirma los cambios, la clase de trabajo se actualiza en la vista SYSCAT.WORKCLASSES.

## Descarte de una clase de trabajo

Si ya no requiere una clase de trabajo, puede descartarla del conjunto de clases de trabajo.

### Antes de empezar

Para descartar una clase de trabajo, se requiere autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para conocer los requisitos adicionales.

## Procedimiento

Para descartar una clase de trabajo:

1. Utilice la palabra clave DROP de la sentencia ALTER WORK CLASS SET. No puede descartar una clase de trabajo si alguna acción de trabajo en algún conjunto de acciones de trabajo está asociada a este conjunto de clases de trabajo y tiene alguna dependencia con la clase de trabajo que desea descartar. En este caso, primero debe descartar todas las acciones de trabajo dependientes antes de descartar la clase de trabajo.
2. Confirme los cambios. Cuando confirme los cambios, la clase de trabajo se eliminará de la vista SYSCAT.WORKCLASSES.

## Creación de un conjunto de clases de trabajo

Para crear un conjunto de clases de trabajo, utilice la sentencia CREATE WORK CLASS SET.

### Antes de empezar

Para crear un conjunto de clases de trabajo, necesita la autorización WLMADM o DBADM.

Si necesita conocer los requisitos adicionales, consulte los temas siguientes:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18

## Procedimiento

Para crear un conjunto de clases de trabajo:

1. Especifique las propiedades siguientes para el conjunto de clases de trabajo utilizando la sentencia CREATE WORK CLASS SET:
  - Un nombre para el conjunto de clases de trabajo. El nombre que especifique debe ser exclusivo en la base de datos.
  - Opcional: una o más clases de trabajo para el conjunto de clases de trabajo. Para obtener más información, consulte “Creación de una clase de trabajo” en la página 58.
2. Confirme los cambios. Cuando confirma los cambios, el conjunto de clases de trabajo se añade a la vista SYSCAT.WORKCLASSSETS.

## Alteración de un conjunto de clases de trabajo

No puede cambiar los atributos del conjunto de clases de trabajo una vez que haya creado dicho conjunto. Sin embargo, puede añadir, alterar y descartar clases de trabajo del conjunto de clases de trabajo mediante la sentencia ALTER WORK CLASS SET.

### Antes de empezar

Para alterar un conjunto de clases de trabajo, necesita la autorización WLMADM o DBADM.

Si necesita conocer los requisitos adicionales, consulte los temas siguientes:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Normas de denominación

## Procedimiento

1. Si desea añadir una clase de trabajo al conjunto de clases de trabajo, utilice la palabra clave ADD. Para obtener información sobre las palabras clave que puede especificar en el momento de añadir una clase de trabajo, consulte "Creación de una clase de trabajo" en la página 58.
2. Si desea alterar una clase de trabajo, utilice la palabra clave ALTER. Para obtener información sobre cómo alterar una clase de trabajo, consulte "Alteración de una clase de trabajo" en la página 61.
3. Si desea descartar una clase de trabajo, utilice la palabra clave DROP. Para obtener información sobre cómo descartar una clase de trabajo de un conjunto de clases de trabajo, consulte "Descarte de una clase de trabajo" en la página 61. Si desea descartar todas las clases de trabajo del conjunto de clases de trabajo, puede descartar el propio conjunto de clases de trabajo. Para obtener más información, consulte "Descarte de un conjunto de clases de trabajo".
4. Confirme los cambios. Cuando confirma los cambios, se actualizará la vista SYSCAT.WORKCLASSES y se mostrarán las clases de trabajo añadidas, alteradas o descartadas.

## Descarte de un conjunto de clases de trabajo

Utilice la sentencia DROP WORK CLASS SET para descartar un conjunto de clases de trabajo.

## Antes de empezar

Para descartar un conjunto de clases de trabajo, necesita la autorización WLMADM o DBADM.

## Acerca de esta tarea

Un conjunto de clases de trabajo solo puede descartarse si no tiene ningún conjunto de acciones de trabajo asociado. Si desea descartar el conjunto de clases de trabajo, primero debe descartar los conjuntos de acciones de trabajo que dependen de él.

## Procedimiento

Para descartar un conjunto de clases de trabajo:

1. Utilice la sentencia DROP WORK CLASS SET.
2. Confirme los cambios. Cuando confirma los cambios, el conjunto de clases de trabajo se elimina de la vista SYSCAT.WORKCLASSSETS. Además, todas las clases de trabajo que formaban parte del conjunto de clases de trabajo también se eliminan de la vista SYSCAT.WORKCLASSES.

## Orden de evaluación de clases de trabajo en un conjunto de clases de trabajo

Un conjunto de clases de trabajo puede tener varias clases de trabajo que coincidan con una actividad de base de datos. Para seleccionar la clase de trabajo a la que corresponde una actividad en un conjunto de clases de trabajo, el servidor de datos revisa las clases de trabajo de acuerdo con el orden de evaluación y se detiene en la primera clase de trabajo que coincide con la actividad.

Si no existe ninguna clase de trabajo coincidente, la actividad de la base de datos no pertenece a ninguna clase de trabajo y no se aplica ninguna acción de trabajo a esa actividad.

Puede alterar el orden de evaluación de las clases de trabajo en un conjunto de clases de trabajo cuando crea o modifica un conjunto de clases de trabajo. Cuando crea o modifica un conjunto de clases de trabajo, determina la posición en la que se ubica una clase de trabajo en el conjunto de clases de trabajo mediante uno de los tres métodos siguientes:

- Especificar la posición absoluta de la clase de trabajo en la lista.  
Por ejemplo, POSITION AT 2. En este caso, la clase de trabajo se ubica en la segunda posición en el conjunto de clases de trabajo, mientras que la clase de trabajo que estaba en segunda posición está ahora en tercera, la tercera clase de trabajo está ahora en la cuarta, y así sucesivamente. Si la posición especificada para la clase de trabajo mediante la sentencia CREATE WORK CLASS SET o ALTER WORK CLASS SET es mayor que el número total de clases de trabajo en el conjunto de clases de trabajo, la clase de trabajo se coloca la última de la lista.
- Utilizar la palabra clave POSITION BEFORE o POSITION AFTER para especificar la posición de la clase de trabajo en relación con las clases de trabajo que ya existan en el conjunto de clases de trabajo.
- Omitir la posición cuando se cree una clase de trabajo.  
En este caso, la nueva clase de trabajo se coloca al final de la lista. La posición que especifique para la clase de trabajo en la lista del conjunto de clases de trabajo no es necesariamente el valor real de la columna EVALUATIONORDER en la vista SYSCAT.WORKCLASSES. El servidor de datos asigna automáticamente el valor de orden para evitar espacios vacíos.

Las clases de trabajo se procesan en el orden en que se reciben, lo que puede afectar al orden de evaluación. Por ejemplo, supongamos que emite la sentencia siguiente:

```
ALTER WORK CLASS SET WCS ALTER WORK CLASS C1 POSITION AT 1  
ALTER WORK CLASS C2 POSITION AT 1
```

Como resultado, la clase de trabajo C1 tiene un orden de evaluación 2 y la clase de trabajo C2 tiene un orden de evaluación 1, porque C2 ha sido la última clase de trabajo procesada.

## Asignación de actividades a clases de trabajo

Si un conjunto de clases de trabajo, a través de un conjunto de acciones de trabajo, se asocia a una base de datos, una carga de trabajo o una superclase de servicio, entonces, justo antes de emprender un proceso de ejecución, ejecución inmediata o petición de apertura, o justo antes de la ejecución del programa de utilidad de carga, se comprueba la actividad de la base de datos para determinar si coincide alguno de los criterios especificados en las clases de trabajo dentro del conjunto de clases de trabajo.

Las clases de trabajo se clasifican dentro del conjunto de clases de trabajo según su orden de evaluación. En función de este orden de evaluación, se comprueba la actividad de la base de datos en relación con cada clase de trabajo de acuerdo con los atributos de la actividad de la base de datos (como el tipo de actividad y la cardinalidad) hasta que se produce una coincidencia o se agota la lista de clases de trabajo en el conjunto de clases de trabajo.

Supongamos que en un conjunto de clases de trabajo existen las siguientes clases de trabajo:

- Orden de evaluación: 1; nombre de clase de trabajo: MyLoad; tipo de clase de trabajo: LOAD
- Orden de evaluación: 2; nombre de clase de trabajo: SmallRead; tipo de clase de trabajo: READ; otros atributos: coste estimado < 300 timerons
- Orden de evaluación: 3; nombre de clase de trabajo: AllDML; tipo de clase de trabajo: DML
- Orden de evaluación: 4; nombre de clase de trabajo: LargeRead; tipo de clase de trabajo: READ; otros atributos: coste estimado > 301 timerons
- Orden de evaluación: 5; nombre de clase de trabajo: MyDDL; tipo de clase de trabajo: DDL

Si se recibe una sentencia SELECT con un coste estimado de 200 timerons, se asigna a la clase de trabajo SmallRead. Si se recibe una actividad DDL (como CREATE TABLE), se le asigna la clase de trabajo MyDDL. Si se recibe una sentencia SELECT con un coste estimado de 500 timerons, se asigna a la clase de trabajo AllDML, porque AllDML se encuentra antes que la clase de trabajo LargeRead. Para obtener más información, consulte “Ejemplo: Cómo trabajar con una clase de trabajo definida con la palabra clave ALL” en la página 68.

## Clasificaciones de trabajo soportadas por los umbrales

Aunque cualquiera de los tipos de umbral que puede utilizarse en las acciones de trabajo pueden asociarse con cualquier clase de trabajo, no se da soporte a todos los tipos de actividades de base de datos en todos esos tipos de umbral.

Por ejemplo, si crea una clase de trabajo para DDL, y luego asocia esa clase de trabajo con una acción de trabajo de umbral ESTIMATEDSQLCOST, ese umbral no se aplicará a ninguna de las peticiones que están clasificadas bajo DDL porque las sentencias DDL no tienen un coste estimado. Si crea una clase de trabajo para ALL, y luego asocia esa clase de trabajo con una acción de trabajo de umbral ESTIMATEDSQLCOST, aunque todas las actividades de base de datos pertenecen a la clase de trabajo ALL, el umbral solo se aplicará las actividades de base de datos que tengan un coste estimado.

En las tablas siguientes se muestran las categorías de clase de trabajo que están soportadas por cada tipo de umbral:

Tabla 23. Clasificaciones de trabajo soportadas por los umbrales

	“Umbral ACTIVITYTOTALTIME” en la página 144	“Umbral ESTIMATEDSQLCOST” en la página 148	“Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154	“Umbral CPUTIME” en la página 145
READ, incluidas las sentencias SET con SQL READ incorporado	Sí	Sí	Sí <sup>1</sup>	Sí
WRITE, incluidas las sentencias SET con SQL WRITE incorporado	Sí	Sí	Sí <sup>1</sup>	Sí
CALL	Sí	No	No	Sí
DML, incluidas las sentencias SET con SQL READ o WRITE incorporado	Sí	Sí	Sí <sup>1</sup>	Sí
DDL	Sí	No	Sí <sup>1</sup>	No

Tabla 23. Clasificaciones de trabajo soportadas por los umbrales (continuación)

	"Umbral ACTIVITYTOTALTIME" en la página 144	"Umbral ESTIMATEDSQLCOST" en la página 148	"Umbral CONCURRENTDBCOORDACTIVITIES" en la página 154	"Umbral CPUTIME" en la página 145
LOAD	Sí	No	Sí <sup>1</sup>	No
ALL	Sí	Algunos	Sí <sup>1</sup>	Algunos

**Nota:**

- Las actividades que se ejecutan en funciones definidas por el usuario (UDF) y que contienen estas clasificaciones de trabajo no se ven afectadas por el umbral CONCURRENTDBCOORDACTIVITIES.

Tabla 24. Clasificaciones de trabajo soportadas por los umbrales (continuación)

	"Umbral SQLROWSREAD" en la página 149	"Umbral SQLROWSRETURNED" en la página 151	"Umbral SQLTEMPSPACE" en la página 152
READ, incluidas las sentencias SET con SQL READ incorporado	Sí	Sí	Sí
WRITE, incluidas las sentencias SET con SQL WRITE incorporado	Sí	Sí	Sí
CALL	No	No (ver la nota)	No
DML, incluidas las sentencias SET con SQL READ o WRITE incorporado	Sí	Sí	Sí
DDL	No	No	No
LOAD	No	No	No
ALL	Algunos	Algunos	Algunos

**Nota:**

- Aunque las sentencias del procedimiento invocado puede devolver filas, porque las filas no se devuelven como resultado de la sentencia CALL, no están controladas por el umbral SQLROWSRETURNED.

### Ejemplo: Análisis de cargas de trabajo por tipo de actividad

Puede utilizar funciones de tabla de la gestión de carga de trabajo de DB2 para examinar las cargas de trabajo de su entorno de acuerdo con los tipos de actividades que se ejecutan.

En algunos casos, puede interesarle el comportamiento de un determinado tipo de actividades, como las actividades LOAD. Por ejemplo, puede observar cuántas actividades LOAD hay actualmente en el sistema del modo siguiente:

```
SELECT COUNT(*)
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(CAST(NULL AS BIGINT), -2))
AS ACTS
WHERE ACTIVITY_TYPE = 'LOAD'
```

Puede obtener un recuento de cuántas actividades de un tipo específico se han enviado desde el último restablecimiento de las estadísticas de la gestión de carga de trabajo de DB2 utilizando la función de tabla WLM\_GET\_WORK\_ACTION\_SET\_STATS, como se muestra en el ejemplo siguiente. Supongamos que existen las clases de trabajo READCLASS y LOADCLASS para las actividades de tipo READ y las actividades de tipo LOAD. El carácter\* representa todas las actividades que no corresponden a la clase de trabajo READCLASS o LOADCLASS.

```
SELECT SUBSTR(WORK_ACTION_SET_NAME,1,18) AS WORK_ACTION_SET_NAME,
SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
SUBSTR(WORK_CLASS_NAME,1,15) AS WORK_CLASS_NAME,
```

```

        LAST_RESET,
        SUBSTR(CHAR(ACT_TOTAL),1,14) AS TOTAL_ACTS
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS('', -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME, PART

```

WORK_ACTION_SET_NAME	PART	WORK_CLASS_NAME	LAST_RESET	TOTAL_ACTS
AdminActionSet	0	ReadClass	2005-11-25-18.52.49.343000	8
AdminActionSet	1	ReadClass	2005-11-25-18.52.50.478000	0
AdminActionSet	0	LoadClass	2005-11-25-18.52.49.343000	2
AdminActionSet	1	LoadClass	2005-11-25-18.52.50.478000	0
AdminActionSet	0	*	2005-11-25-18.52.50.478000	0
AdminActionSet	1	*	2005-11-25-18.52.50.478000	0

Para visualizar el promedio de vida útil de las actividades LOAD, puede crear un conjunto de acciones de trabajo para correlacionar las actividades LOAD con una subclase de servicio específica. Por ejemplo, supongamos que correlaciona actividades LOAD con la superclase de servicio LOADSERVICECLASS bajo la superclase de servicio MYSUPERCLASS. A continuación, puede consultar la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS.

```

SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       CAST(COORD_ACT_LIFETIME_AVG / 1000 AS DECIMAL(9,3)) AS AVGLIFETIME
FROM TABLE
      (WLM_GET_SERVICE_SUBCLASS_STATS('MYSUPERCLASS', 'LOADSERVICECLASS', -2))
AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, PART

```

SUPERCLASS_NAME	SUBCLASS_NAME	PART	AVGLIFETIME
SYSDEFAULTUSERCLASS	LOADSERVICECLASS	0	4691.242
SYSDEFAULTUSERCLASS	LOADSERVICECLASS	1	4644.740
SYSDEFAULTUSERCLASS	LOADSERVICECLASS	2	4612.431
SYSDEFAULTUSERCLASS	LOADSERVICECLASS	3	4593.451

## Ejemplo: Utilización de un conjunto de clases de trabajo para gestionar tipos específicos de actividades

En el ejemplo siguiente se muestra cómo utilizar un conjunto de clases de trabajo para gestionar actividades de DML.

Supongamos que a diario tiene un gran número de aplicaciones en ejecución en su base de datos NONAME y que últimamente se están produciendo algunos problemas de rendimiento. Para hacer frente a algunos de estos problemas, decide que necesita poder controlar el número de consultas grandes (es decir, cualquier consulta que tenga un coste estimado de más de 9999 timerons o una cardinalidad de más de 9999 filas) que pueden ejecutarse simultáneamente en la base de datos.

Para controlar el número de consultas grandes que pueden ejecutarse en la base de datos, realizaría lo siguiente:

1. Crear un conjunto de clases de trabajo MYWORKCLASSSET que contenga las clases de trabajo: una para las consultas con un coste estimado grande y otra para las consultas con una cardinalidad estimada grande. Por ejemplo:

```

CREATE WORK CLASS SET MYWORKCLASSSET
  (WORK CLASS LARGEESTIMATEDCOST WORK TYPE DML
  FOR TIMERONCOST FROM 10000 TO UNBOUNDED,
  WORK CLASS LARGECARDINALITY WORK TYPE DML
  FOR CARDINALITY FROM 10000 TO UNBOUNDED)

```

2. Crear un conjunto de acciones de trabajo DATABASEACTIONS que contenga las acciones de trabajo que deban aplicarse a las clases de trabajo en el conjunto de clases de trabajo MYWORKCLASSET en el nivel de la base de datos.

```
CREATE WORK ACTION SET DATABASEACTIONS FOR DATABASE
USING WORK CLASS SET LARGEQUERIES
(WORK ACTION ONECONCURRENTQUERY ON WORK CLASS LARGEESTIMATEDCOST
WHEN CONCURRENTDBCOORDACTIVITIES > 1 AND QUEUEDACTIVITIES > 1 STOP EXECUTION,
WORK ACTION TWOCONCURRENTQUERIES ON WORK CLASS LARGECARDINALITY
WHEN CONCURRENTDBCOORDACTIVITIES > 2 AND QUEUEDACTIVITIES > 3 STOP EXECUTION)
```

Además, varias aplicaciones administrativas grandes se ejecutan diariamente en la base de datos, y desea que estas aplicaciones se ejecuten en una agrupación de recursos. Para cumplir este objetivo, crearía una superclase de servicio denominada ADMINAPPS para estas aplicaciones. Por cada aplicación, crearía una carga de trabajo para correlacionarla con la superclase de servicio ADMINAPPS.

Puesto que es importante que las consultas (sentencias SELECT) se ejecuten rápidamente, decide crear una subclase de servicio denominada SELECTS en la superclase de servicio ADMINAPPS para estas consultas.

Para correlacionar las sentencias SELECT con la subclase de servicio SELECTS:

1. Cree un conjunto de clases de trabajo SELECTDML que contenga una clase de trabajo para todas las sentencias SELECT que no actualicen la base de datos:
2. Cree un conjunto de acciones de trabajo ADMINAPPSACTIONS. Este conjunto de acciones de trabajo contiene una acción de trabajo que debe aplicarse a la clase de trabajo en el conjunto de clases de trabajo SELECTDML en el nivel de la superclase de servicio.

```
CREATE WORK CLASS SET SELECTDML (WORK CLASS SELECTCLASS WORK TYPE READ)
CREATE WORK ACTION SET ADMINAPPSACTIONS FOR SERVICE CLASS ADMINAPPS
USING WORK CLASS SET SELECTDML
(WORK ACTION MAPSELECTS ON WORK CLASS SELECTCLASS MAP ACTIVITY TO SELECTS)
```

## Ejemplo: Cómo trabajar con una clase de trabajo definida con la palabra clave ALL

En este ejemplo se muestra cómo trabajar con una clase de trabajo definida como ALL, que potencialmente cubre todas las actividades reconocidas en la base de datos.

La infraestructura de gestión de carga de trabajo de DB2 reconoce el siguiente subconjunto específico a partir del dominio de todas las actividades posibles de la base de datos: operaciones de carga, la sentencia CALL, todas las sentencias DDL y todas las sentencias DML. La gestión de carga de trabajo de DB2 proporciona soporte completo para supervisar y controlar estas *actividades reconocidas*.

Cuando se utiliza una clase de trabajo con el tipo ALL con una acción de trabajo de correlación, toda la actividad de la base de datos reconocida se correlaciona con la subclase de servicio especificada en la acción de trabajo. Si se utiliza una clase de trabajo con el tipo de trabajo ALL con una acción de trabajo de umbral, el tipo de umbral determina las actividades de la base de datos a las que se aplica el umbral. Considere el ejemplo siguiente.

Supongamos que crea un conjunto de clases de trabajo llamado Example con las siguientes clases de trabajo. El orden de evaluación de la clase de trabajo será el siguiente:

1. SMALLDML, que es para todo el código SQL de tipo DML con un coste estimado de menos de 1.000 timerons.
2. LOADUTIL, que es para el programa de utilidad de carga.
3. ALLACTIVITY, que es para toda la actividad de la base de datos.

ALLACTIVITY es la última clase de trabajo que se evalúa y abarca actividades de la base de datos que no se corresponden con las primeras tres clases de trabajo.

El DDL para crear este conjunto de clases de trabajo es:

```
CREATE WORK CLASS SET EXAMPLE
(WORK CLASS SMALLDML WORK TYPE DML FOR TIMERONCOST FROM 0 TO 999,
WORK CLASS LOADUTIL WORK TYPE LOAD,
WORK CLASS ALLACTIVITY WORK TYPE ALL)
```

Supongamos que tiene una superclase de servicio denominada EXAMPLESERVICECLASS, y que tiene dos subclases de servicio denominadas SMALLACTIVITY y OTHERACTIVITY. Desea configurar el sistema de modo que todas las actividades pequeñas de la base de datos se ejecuten en la subclase de servicio SMALLACTIVITY, mientras que todas las demás actividades reconocidas de la base de datos, salvo las del programa de utilidad de carga, se ejecuten en la subclase de servicio OTHERACTIVITY. No desea volver a correlacionar el programa de utilidad de carga con ninguna otra subclase de servicio, sino que, en su lugar, desea ejecutarlo en la subclase de servicio por omisión.

Para llevar a cabo estos objetivos, configuraría un conjunto de acciones de trabajo, SERVICECLASSACTIONS, para la superclase de servicio EXAMPLESERVICECLASS. El conjunto de acciones de trabajo SERVICECLASSACTIONS contendría las siguientes acciones de trabajo.

*Tabla 25. Conjunto de acciones de trabajo SERVICECLASSACTIONS*

Acción de trabajo	Clase de trabajo a la que se aplica	Acción
MAPDML	SMALLDML	Se correlaciona con la subclase de servicio SMALLACTIVITY
COUNTLOAD	LOADUTIL	Cuenta el número de actividades LOAD
MAPOTHER	ALLACTIVITY	Se correlaciona con la subclase de servicio OTHERACTIVITY

EL DDL para crear este conjunto de acciones de trabajo es:

```
CREATE WORK ACTION SET SERVICECLASSACTIONS FOR SERVICE CLASS EXAMPLESERVICECLASS
USING WORK CLASS SET EXAMPLE
(WORK ACTION MAPDML ON WORK CLASS SMALLDML MAP ACTIVITY TO SMALLACTIVITY,
WORK ACTION COUNTLOAD ON WORK CLASS LOADUTIL COUNT ACTIVITY,
WORK ACTION MAPOTHER ON WORK CLASS ALLACTIVITY MAP ACTIVITY TO OTHERACTIVITY)
```

Utilizando esta configuración, todas las actividades de DML pequeñas se ejecutan bajo la subclase de servicio SMALLACTIVITY. La acción de trabajo COUNTLOAD se aplica a la clase de trabajo LOADUTIL, que se ejecuta bajo la subclase de servicio por omisión. Todas las demás actividades reconocidas de la base de datos se ejecutan bajo la subclase de servicio OTHERACTIVITY.

**Nota:** Si la clase de trabajo ALLACTIVITY estuviera en primer lugar en el orden de evaluación, todas las actividades reconocidas se correlacionarían con la subclase de servicio OTHERACTIVITY.

Supongamos ahora que desea definir un conjunto de acciones de trabajo para la base de datos y aplicar umbrales que controlen lo que se permite ejecutar simultáneamente en el sistema. Podría crear un conjunto de acciones de trabajo denominado DATABASEACTIONS que contuviera las siguientes acciones de trabajo. El DML para crear este conjunto de acciones de trabajo es:

```
CREATE WORK ACTION SET DATABASEACTIONS FOR DATABASE USING WORK CLASS SET EXAMPLE
(WORK ACTION CONCURRENTSMALLDML ON WORK CLASS SMALLDML
WHEN CONCURRENTDBCOORDACTIVITIES > 1000 AND QUEUEDACTIVITIES > 10000
COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION CONCURRENTLOAD ON WORK CLASS LOADUTIL
WHEN CONCURRENTDBCOORDACTIVITIES > 2 AND QUEUEDACTIVITIES > 10
COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION CONCURRENTOTHER ON WORK CLASS ALLACTIVITY
WHEN CONCURRENTDBCOORDACTIVITIES > 100 AND QUEUEDACTIVITIES > 100
COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION MAXCOSTALLOWED ON WORK CLASS ALLACTIVITY
WHEN ESTIMATEDSQLCOST > 1000000 COLLECT ACTIVITY DATA STOP EXECUTION)
```

Tabla 26. Conjunto de acciones de trabajo DATABASEACTIONS

Acción de trabajo	Clase de trabajo a la que se aplica	Tipo y valor de umbral	Acción
CONCURRENTSMALLDML	SMALLDML	Simultaneidad de hasta 1.000 sentencias; cola de hasta 10.000 sentencias	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
CONCURRENTLOAD	LOADUTIL	Simultaneidad de hasta 2 apariciones; cola de hasta 10 apariciones	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
CONCURRENTOTHER	ALLACTIVITY	Simultaneidad de hasta 100 actividades; cola de hasta 100 actividades	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
MAXCOSTALLOWED	ALLACTIVITY	Coste estimado de SQL de hasta 1.000.000 timerons	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>

Cuando se aplican estas acciones de trabajo, pueden ejecutarse hasta 1.000 sentencias de SQL de tipo DML pequeñas (debido a la clase de trabajo SMALLDML) al mismo tiempo, y pueden colocarse en cola hasta 10.000 de estas sentencias. Sólo pueden ejecutarse dos apariciones del programa de utilidad de carga al mismo tiempo y colocarse en cola hasta diez apariciones. Sólo se permite ejecutar 100 actividades que no sean LOAD y que no sean DML pequeño al mismo tiempo, y sólo pueden colocarse en cola 100 de estas actividades al mismo tiempo. En todos los casos, si se infringe el umbral de cola, no se permitirá ejecutar la actividad de la base de datos y se devolverá un mensaje de error.

Además, la acción de trabajo MAXCOSTALLOWED se aplica a la clase de trabajo ALLACTIVITY. Esto significa que no se permitirá la ejecución de una actividad de la base de datos con un coste estimado (es decir, sentencias DML y XQueries) de más de 1.000.000 de timerons. Aunque la acción de trabajo MAXCOSTALLOWED se aplica a la clase de trabajo ALLACTIVITY, esta acción de trabajo sólo afectará a

las actividades de la base de datos que tengan un coste estimado superior a 1.000.000 de timerons. Esta acción de trabajo no afecta a las actividades que no tengan coste estimado, como DDL.



---

## Capítulo 3. Gestión de actividades

Después de haber identificado el trabajo que se ejecuta en el servidor de datos, está preparado para gestionar activamente este trabajo asignando recursos e imponiendo controles.

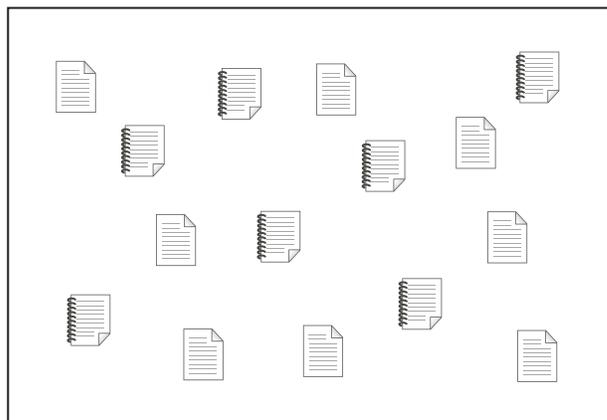
---

### Asignación de recursos con clases de servicio

Una clase de servicio define un entorno de ejecución en el que el trabajo puede ejecutarse. Este entorno de ejecución asigna los recursos disponibles y puede incluir umbrales que determinen cómo está permitido que se ejecute el trabajo.

Todo el trabajo se ejecuta en una clase de servicio, y puede utilizar cargas de trabajo para asignar trabajo a superclases de servicio o asignar trabajo a subclases de servicio en una superclase de servicio utilizando cargas de trabajo, la acción de umbral REMAP ACTIVITY o la acción de trabajo MAP ACTIVITY. Cuando define una carga de trabajo, indica la clase de servicio en la que se ejecuta el trabajo asociado a esa carga de trabajo. Por omisión, también existe una carga de trabajo de usuario por omisión (SYSDEFAULTUSERWORKLOAD) que correlaciona el trabajo con la clase de servicio de usuario por omisión (SYSDEFAULTUSERCLASS), por lo tanto, cualquier trabajo no correlacionado explícitamente con una clase de servicio definida por el usuario mediante la utilización de una carga de trabajo definida por el usuario se ejecutará en la clase de servicio de usuario por omisión.

Sin clases de servicio, las peticiones no pueden organizarse en agrupaciones lógicas que puedan volver a organizarse, como se muestra en la figura siguiente.



*Figura 9. Trabajo no organizado*

Puede crear distintas superclases de servicio para proporcionar el entorno de ejecución para distintos tipos de trabajo y, a continuación, asignar las peticiones aplicables a las superclases de servicio. Supongamos que tiene aplicaciones de dos líneas de gestión empresarial por separado, una de finanzas y otra de inventario. Cada línea de gestión empresarial tendría sus propias aplicaciones para realizar las tareas bajo su responsabilidad en la organización. Puede organizar las peticiones en categorías que tengan sentido para sus objetivos de gestión de carga de trabajo. En la figura siguiente, se han asignado distintas superclases de servicio a distintas

líneas de gestión empresarial.

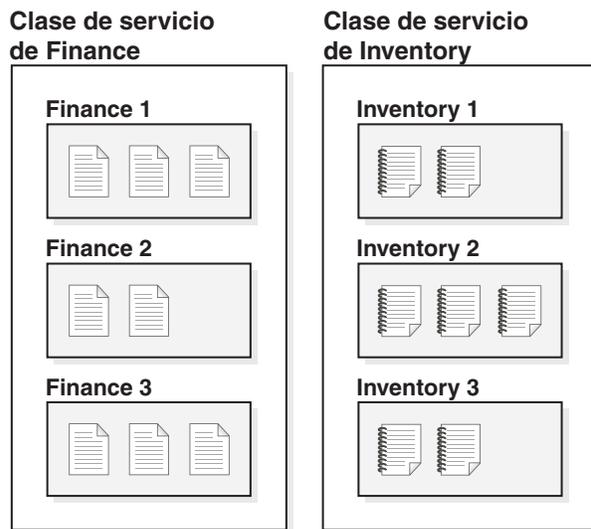


Figura 10. Trabajo organizado por clases de servicio

En la figura anterior, las actividades de ambas superclases de servicio se han subdividido aún más. La clase de servicio proporciona una jerarquía de dos niveles: una superclase de servicio y subclases de servicio bajo ésta. Esta jerarquía permite una división más compleja del entorno de ejecución y emula mejor un modelo real. A menos que se especifique algo distinto, las subclases de servicio heredan características de la superclase de servicio. Utilice las subclases de servicio para subdividir aún más el trabajo en la superclase de servicio.

## Priorización y control de recursos

Cuando crea o altera un objeto de clase de servicio, puede definir diversos controles de recursos:

Tabla 27. Control de recursos que las clases de servicio admiten

Control	Descripción
Prioridad de agente	Este control establece un nivel de prioridad de procesador para las hebras de agente que se ejecutan en una clase de servicio. Esta prioridad fluye por el sistema operativo como una prioridad relativa (delta) respecto a otras hebras y procesos que se ejecutan en el servidor de datos. <b>Nota:</b> Este control no puede establecerse cuando el correlacionador de salida está en uso.
Prioridad de captación previa	Este control asigna una prioridad a las peticiones de captación previa, lo que afecta al orden en que el servidor de datos las direcciona.
Prioridad de agrupación de almacenamientos intermedios	Este control asigna una prioridad de agrupación de almacenamientos intermedios a las clases de servicio que afecta a la probabilidad de intercambio de las páginas que las actividades de una clase de servicio captan previamente.

Tabla 27. Control de recursos que las clases de servicio admiten (continuación)

Control	Descripción
Correlacionador de salida	<p>Este control permite que parte de los recursos de una carga de trabajo la controle un gestor de carga de trabajo del sistema operativo, como el gestor de carga de trabajo de AIX o la gestión de carga de trabajo de Linux. El código fluye a través del agente hasta el gestor de carga de trabajo externo y se correlaciona con un grupo de recursos definido con el gestor.</p> <p>Cuando la gestión de carga de trabajo de DB2 se utiliza junto con un gestor de carga de trabajo del sistema operativo, existen controles adicionales. Con el gestor de carga de trabajo de AIX, puede controlar la cantidad de recursos de procesador que han de asignarse a cada clase de servicio estableciendo una cuota mínima, máxima o relativa de recursos de procesador para cada clase de servicio. Con la gestión de carga de trabajo de Linux, puede controlar la cantidad de recursos de CPU estableciendo cuotas para cada clase de servicio relativas respecto a la clase de servicio por omisión de Linux.</p> <p><b>Nota:</b> Este control no puede establecerse cuando la prioridad de agente está en uso.</p>

## Subclases de servicio

Aunque la superclase de servicio es el nivel más alto para el trabajo, las actividades sólo se ejecutan en subclases de servicio. Cada superclase de servicio tiene definida una subclase de servicio por omisión para ejecutar las actividades que no se han asignado a una subclase definida explícitamente. Esta subclase por omisión se crea al crearse la superclase de servicio. Puede crear subclases adicionales en una clase de servicio a medida que las necesite para aislar aún más el trabajo. A excepción de los histogramas y de las opciones COLLECT ACTIVITY DATA, COLLECT AGGREGATE ACTIVITY DATA y COLLECT AGGREGATE REQUEST DATA, una subclase de servicio hereda los atributos de su superclase de servicio, a menos que se especifique algo distinto. Los recursos de la superclase los comparten todas las subclases que ésta contiene.

Sólo puede definir un único nivel de subclases (es decir, no puede definir una subclase bajo otra subclase, sólo bajo una superclase de servicio).

La figura siguiente es un ejemplo de una configuración de gestión de carga de trabajo de DB2 en la que se utilizan cargas de trabajo y clases de servicio:

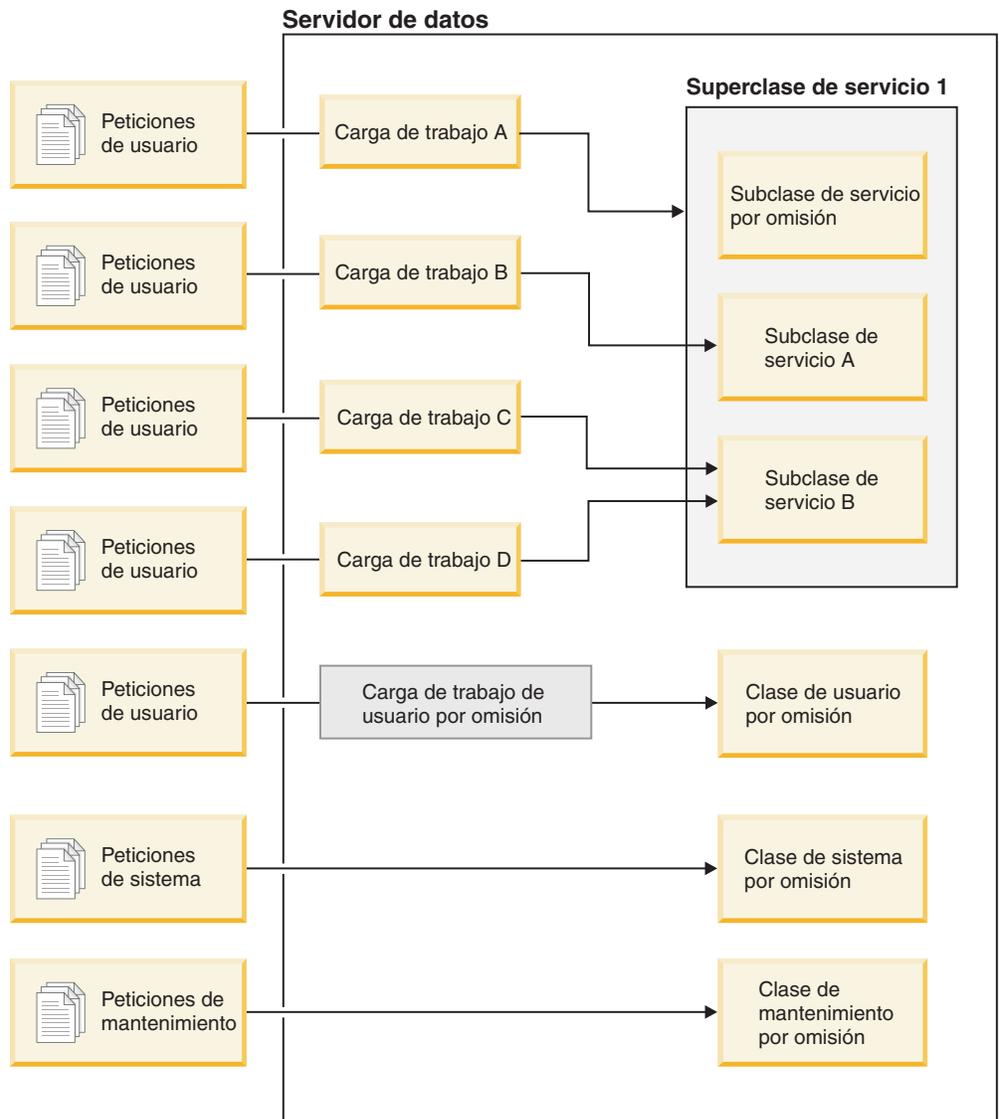


Figura 11. Configuración de gestión de carga de trabajo de DB2 personalizada en la que se utilizan cargas de trabajo y clases de servicio

A medida que las peticiones de usuario entran en el servidor de datos, éstas se identifican como pertenecientes a una carga de trabajo determinada y se asignan a una superclase o subclase de servicio. También existen peticiones del sistema (por ejemplo, las captaciones previas) que se ejecutan bajo una clase de servicio del sistema por omisión especial (SYSDEFAULTSYSTEMCLASS) y peticiones de mantenimiento gestionadas por DB2 (como la sentencia automática RUNSTATS del supervisor de salud) que se ejecutan bajo una clase de servicio de mantenimiento por omisión (SYSDEFAULTMAINTENANCECLASS).

Puede ver las clases de servicio consultando la vista SYSCAT.SERVICECLASSES.

## Superclases y subclases de servicio por omisión

Cada base de datos nueva o actualizada tiene predefinidas tres superclases de servicio por omisión: la clase de usuario por omisión, la clase de mantenimiento por omisión y la clase de sistema por omisión.

No puede inhabilitar ni descartar ninguna de las superclases de servicio por omisión.

Todas las superclases de servicio por omisión se crean con una subclase de servicio por omisión. No puede crear subclases de servicio adicionales para las superclases de servicio por omisión. La subclase de servicio por omisión se crea siempre con el nombre `SYSDEFAULTSUBCLASS`, como sigue:

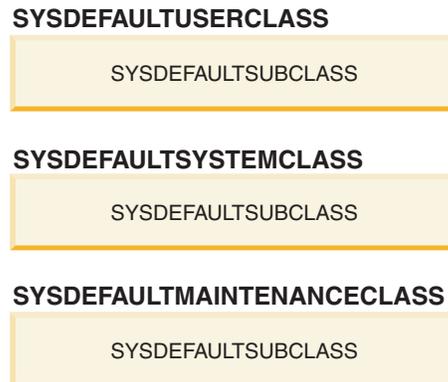


Figura 12. Jerarquía de clase de servicio de los niveles

Todo el trabajo que emiten las conexiones a una superclase de servicio por omisión se procesará en la subclase de servicio por omisión de dicha superclase de servicio.

Las superclases de servicio por omisión y sus subclases de servicio por omisión se descartan únicamente cuando se descarta la base de datos. No pueden descartarse utilizando la sentencia `DROP SERVICE CLASS`.

#### **Superclase de servicio de usuario por omisión (SYSDEFAULTUSERCLASS)**

Por omisión, todas las actividades de usuario se ejecutan en `SYSDEFAULTUSERCLASS`.

#### **Superclase de servicio de mantenimiento por omisión (SYSDEFAULTMAINTENANCECLASS)**

La superclase de servicio de mantenimiento por omisión realiza el seguimiento de las conexiones internas de DB2 que realizan tareas de mantenimiento y administración de la base de datos. Las conexiones de los agentes de proceso en segundo plano asíncronos de DB2 (ABP) se correlacionan con esta superclase de servicio. Los agentes ABP son agentes internos que realizan tareas de mantenimiento de la base de datos. La limpieza de índices asíncrona (AIC) es un ejemplo de una tarea dirigida por ABP. Los agentes ABP reducen automáticamente su consumo de recursos y el número de subagentes cuando el número de conexiones de usuario aumenta en el servidor de datos. Los programas de utilidad que emiten las conexiones de usuario se correlacionan utilizando las clases de servicio normales. No puede implementar umbrales de clase de servicio en `SYSDEFAULTMAINTENANCECLASS`.

Las conexiones internas cuyo seguimiento realiza la superclase de servicio de mantenimiento por omisión incluyen:

- Conexiones ABP (incluida AIC)
- Copia de seguridad iniciada por el supervisor de salud
- RUNSTATS iniciado por el supervisor de salud

- REORG iniciado por el supervisor de salud

### **Superclase de servicio de sistema por omisión (SYSDEFAULTSYSTEMCLASS)**

La superclase de servicio de sistema por omisión realiza el seguimiento de las hebras y las conexiones internas de DB2 que realizan tareas en el nivel del sistema. No puede definir subclases de servicio para esta superclase de servicio, ni asociarle cargas de trabajo ni acciones de trabajo. Además, no puede implementar umbrales de clase de servicio en SYSDEFAULTSYSTEMCLASS. Las hebras y conexiones de DB2 cuyo seguimiento realiza la superclase de servicio de sistema por omisión incluyen:

- Daemon ABP
- Gestor de memoria de ajuste automático (STMM)
- Unidades asignables de motor (EDU) del captador previo (db2pfchr)
- EDU del limpiador de página (db2pclnr)
- EDU del lector de anotaciones cronológicas (db2loggr)
- EDU del grabador de anotaciones cronológicas (db2loggw)
- EDU del lector de archivos de anotaciones cronológicas (db2lfr)
- EDU del detector de puntos muertos (db2dlock)
- Supervisores de sucesos (db2evm)
- Grabadores rápidos del supervisor de sucesos (db2fw)
- Conexiones que realizan tareas de nivel del sistema

## **Correlación de actividades con clases de servicio**

A todas las conexiones de base de datos se les asigna una carga de trabajo al principio de la primera unidad de trabajo. Cuando se inicia una aparición de la carga de trabajo, todas las actividades que se ejecutan bajo esa aparición de la carga de trabajo se correlacionan con las clases de servicio de acuerdo con el nombre de la clase de servicio especificado en la definición de carga de trabajo.

El servidor de datos asigna una conexión a una definición de carga de trabajo en una definición de carga de trabajo si la conexión cumple los criterios definidos para esa definición de carga de trabajo. Por ejemplo, puede preparar una configuración de gestión de carga de trabajo de modo que todas las conexiones de la aplicación A pertenezcan a la definición de carga de trabajo Alfa, mientras que todas las conexiones de la aplicación B pertenezcan a la definición de carga de trabajo Beta.

Si se asigna una superclase de servicio a la aparición de la carga de trabajo, las actividades enviadas para esa aparición de la carga de trabajo pueden reasignarse a una subclase de servicio definida por el usuario en esa superclase de servicio utilizando un conjunto de acciones de trabajo.

Puede utilizar la carga de trabajo para correlacionar actividades de una conexión con una superclase de servicio especificando la palabra clave SERVICE CLASS de la sentencia CREATE WORKLOAD. Suponiendo que ninguna clase de trabajo o acción de trabajo se aplique a la actividad, la actividad se ejecuta en la subclase de servicio por omisión de la superclase de servicio. También puede utilizar una carga de trabajo para correlacionar actividades de una conexión con una subclase de servicio en la superclase de servicio especificando la palabra clave UNDER para la palabra clave SERVICE CLASS de la sentencia CREATE WORKLOAD. En este caso, la conexión todavía pertenece a la superclase de servicio, pero todas las

actividades emitidas desde esa conexión se correlacionan automáticamente con la subclase de servicio especificada en la definición de la carga de trabajo.

Después de que se correlacione una actividad con una subclase de servicio y de que haya empezado la ejecución, permanece en esa subclase de servicio, a menos que la correlacione con otra subclase de servicio (con un umbral). Volver a correlacionar es el proceso por el cual puede cambiar las asignaciones de recursos para una actividad mediante una correlación diferente de la actividad con la subclase de servicio. Tanto la subclase de servicio fuente como la de destino deben existir bajo de la misma superclase. Después de volver a correlacionar la actividad, continúa ejecutando la nueva subclase de servicio.

En la figura siguiente se muestra la relación entre conexiones, cargas de trabajo y superclases de servicio. Las conexiones que cumplen la definición de la carga de trabajo A se correlacionan con la superclase de servicio 1; las conexiones que cumplen la definición de las cargas de trabajo B o C se correlacionan con la superclase de servicio 2; las conexiones que cumplen la definición de la carga de trabajo D se correlacionan con la superclase de servicio SYSDEFAULTUSERCLASS.

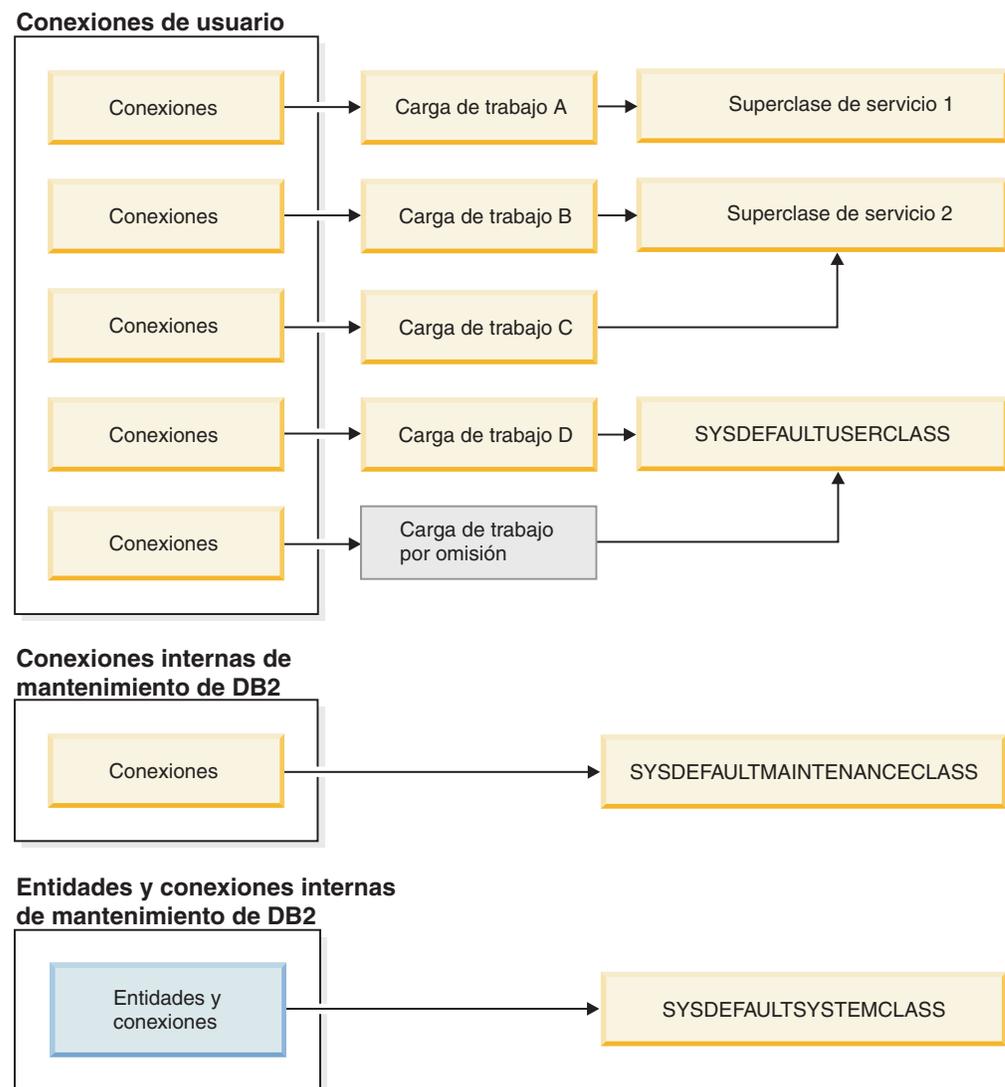


Figura 13. Correlación de conexiones de base de datos con una superclase de servicio

## Diferenciación más detallada de las actividades

Si tiene una configuración más compleja de gestor de carga de trabajo de DB2, es posible que desee gestionar las actividades de forma diferente en función del tipo de actividad o algún otro atributo de actividad. Por ejemplo, es posible que desee realizar una de las acciones siguientes:

- Colocar DML en una subclase de servicio diferente que DDL.
- Colocar todas las consultas de tipo lectura con un coste estimado de menos de 100 timerons (activaciones de temporizador) en una subclase de servicio diferente que las demás consultas de tipo lectura.

En una configuración más compleja puede configurar la carga de trabajo para correlacionar actividades de la conexión con la superclase de servicio. A continuación, mediante acciones de trabajo (contenidas en un conjunto de acciones de trabajo aplicado a la superclase de servicio), puede volver a correlacionar actividades en función de su tipo o atributo para subclases de servicio en una superclase de servicio.

Específicamente, puede aplicar un conjunto de acciones de trabajo que contenga una acción de trabajo MAP ACTIVITY con la superclase de servicio. Todas las actividades que están correlacionadas con la superclase de servicio y que coinciden con una clase de trabajo a la que está asociada la acción de trabajo ACTIVITY, se correlacionan a una subclase de servicio especificada mediante la acción de trabajo.

Si una definición de carga de trabajo se asocia a una subclase de servicio, las actividades que se envían a través de dicha carga de trabajo no se ven afectadas por ningún conjunto de acciones de trabajo que esté asignado a la superclase de servicio principal.

- Una actividad puede correlacionarse a una subclase de servicio en una superclase de servicio mediante una carga de trabajo.
- Una acción de trabajo que correlaciona la actividad con una subclase de servicio diferente en la misma superclase de servicio también se aplica a la actividad.

Si una actividad no se correlaciona con una subclase de servicio mediante una carga de trabajo o una acción de trabajo, la actividad se correlaciona con la subclase por omisión (SYSDEFAULTSUBCLASS) de la superclase de servicio para esa actividad.

Cuando las actividades de la base de datos se han correlacionado con sus superclases de servicio y subclases de servicio respectivas, puede implementar controles en todas las actividades en una determinada clase de servicio. Hay disponibles estadísticas en el nivel de clase de servicio que puede utilizar para supervisar actividades de la base de datos en esa clase de servicio.

En la figura siguiente se muestran peticiones a la base de datos que se está correlacionando con una superclase de servicio o una subclase de servicio mediante cargas de trabajo. Para obtener más información sobre cómo se utilizan las acciones de trabajo para correlacionar actividades con una subclase de servicio, consulte “Acciones de trabajo y conjuntos de acciones de trabajo” en la página 105

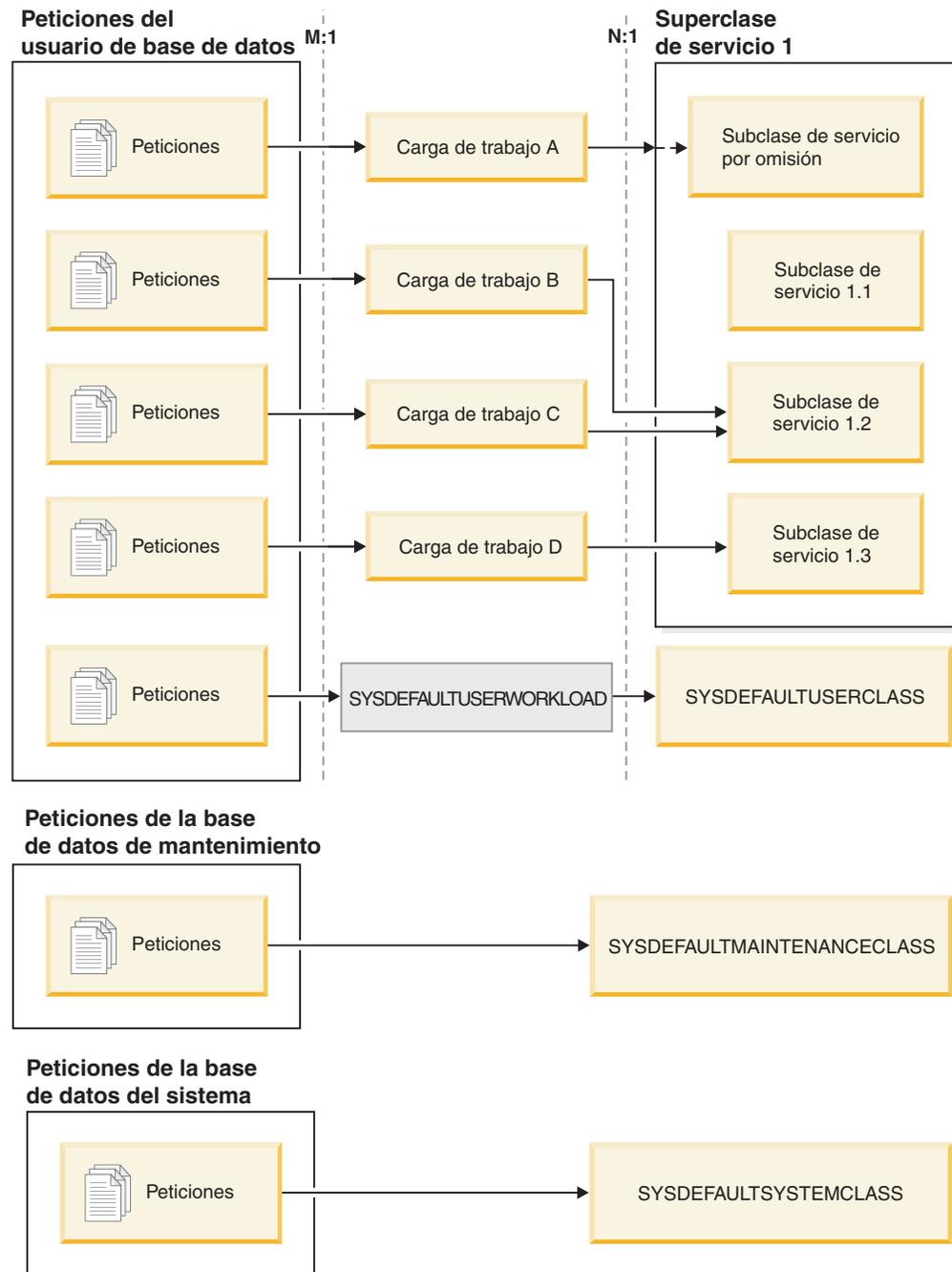


Figura 14. Conexiones de base de datos que se correlacionan con una superclase de servicio

Las conexiones que no se correlacionan con una definición de carga de trabajo definida por el usuario se correlacionan con la definición de carga de trabajo del usuario por omisión, `SYSDEFAULTUSERWORKLOAD`. Por omisión, las conexiones de la definición de carga de trabajo por omisión (`SYSDEFAULTUSERWORKLOAD`) se correlacionan con la superclase de servicio `SYSDEFAULTUSERCLASS`, que es la superclase de servicio por omisión para peticiones de usuario. Puede alterar la carga de trabajo `SYSDEFAULTUSERWORKLOAD` de modo que se correlacione con una clase de servicio diferente. Las conexiones de mantenimiento internas de DB2 se correlacionan con `SYSDEFAULTMAINTENANCECLASS`, que es la superclase de servicio por omisión para peticiones de mantenimiento. Las entidades y conexiones

internas del sistema se correlacionan con SYSDEFAULTSYSTEMCLASS, que es la superclase de servicio por omisión para conexiones y hebras de DB2 que realizan tareas de nivel del sistema.

## Prioridad de agente de clases de servicio

Puede asociar cada clase de servicio de DB2 con una prioridad de agente relativa, que controla la prioridad del procesador del servidor de datos. Esta prioridad se establece para todos los agentes que trabajan en una clase de servicio y es relativa a la prioridad de agente de todos los agentes de DB2 restantes.

Si no especifica el valor de prioridad de agente para una clase de servicio, todos los agentes de esa clase de servicio tendrán la misma prioridad que el resto de los agentes de DB2.

El establecimiento de la prioridad de agente para una clase de servicio de DB2 ajusta la prioridad de los agentes únicamente para el trabajo nuevo que entre en la clase de servicio. Las hebras que no son de agente y que se ejecutan en la clase de servicio no utilizan el valor de prioridad de agente que ha especificado. La gestión de carga de trabajo de DB2 no asigna la prioridad de agente de clase de servicio al trabajo que se lleva a cabo en un proceso de modalidad delimitada (FMP). Los procedimientos delimitados no ejecutan su propia lógica en una clase de servicio. Dichos procedimientos delimitados se ejecutan en el FMP de DB2 y este trabajo no lo realizan los agentes de DB2. Como recordatorio, WLM de DB2 controla los agentes de DB2.

Si va a integrar clases de servicio de DB2 con un gestor de carga de trabajo del sistema operativo como el gestor de carga de trabajo de AIX o la gestión de carga de trabajo de Linux, puede utilizar el gestor de carga de trabajo del sistema operativo para especificar la prioridad de procesador que se utilizará para la clase del sistema operativo (como cuotas de procesador), y luego permitir que la clase de servicio de DB2 herede este valor a través del valor OUTBOUND CORRELATOR de la clase de servicio de DB2. La prioridad del procesador que especifique mediante el gestor de carga de trabajo del sistema operativo controla la prioridad de los agentes que se ejecutan en la clase de servicio de DB2, y se pasa por alto cualquier otro valor de prioridad de agente de la clase de servicio.

**Nota:** No puede utilizar la prioridad de agente para limitar la cantidad de CPU utilizada por un subconjunto de trabajo en el sistema. La prioridad de agente favorece el uso de los recursos de CPU por parte de un trabajo cuando hay cargas de trabajo en curso compitiendo por los recursos de CPU, pero dicha prioridad no asigna ni gestiona de manera explícita los recursos de CPU entre esas cargas de trabajo. Por lo tanto, bajo aquellas circunstancias en que las cargas de trabajo en curso no requieren por igual recursos de CPU, la prioridad de agente parece no ser eficaz en aquellos casos en los que el trabajo de menos prioridad realiza más peticiones de recursos de CPU que el trabajo de más prioridad. No obstante, la prioridad de agente es eficaz si tiene cargas de trabajo en curso que soliciten por igual recursos de CPU y solo desee favorecer el consumo de CPU por parte de una carga de trabajo en lugar de regularlo o controlarlo. Para aplicar un control más explícito sobre el consumo de CPU, los controles de simultaneidad o de integración de Linux WLM son las opciones más eficaces.

**Importante:** No utilice el parámetro en desuso **agentpri** de configuración del gestor de la base de datos con la gestión de carga de trabajo de DB2. Este parámetro de configuración puede utilizarse para definir con un valor fijo la prioridad absoluta del procesador de todos los agentes en una instancia de DB2.

Sin embargo, si establece la prioridad absoluta de un agente utilizando **agentpri**, no podrá modificar la prioridad relativa del agente estableciendo la prioridad de agente de clase de servicio de DB2 ni utilizando el gestor de carga de trabajo del sistema operativo. Si establece **agentpri**, la prioridad de agente de clase de servicio y el gestor de carga de trabajo del sistema operativo no tendrán ningún efecto en la prioridad de los agentes.

En los sistemas operativos UNIX y Linux, los valores válidos son DEFAULT y -20 a 20 (SQLSTATE 42615). Los valores negativos indican una prioridad relativa más alta. Los valores positivos indican una prioridad relativa más baja.

En los sistemas operativos Windows, los valores válidos son DEFAULT y -6 a 6 (SQLSTATE 42615). Los valores negativos indican una prioridad relativa más baja. Los valores positivos indican una prioridad relativa más alta.

En los sistemas operativos AIX, el propietario de la instancia debe tener funciones CAP\_NUMA\_ATTACH y CAP\_PROPAGATE para definir una prioridad relativa más alta para los agentes de una clase de servicio utilizando AGENT PRIORITY. Para otorgar estas funciones, debe iniciarse la sesión como root y ejecutar el mandato siguiente:

```
chuser capabilities=CAP_NUMA_ATTACH,CAP_PROPAGATE
```

En Solaris 10 o superior, el propietario de la instancia debe tener el privilegio proc\_priocntl para establecer una prioridad relativa más alta para agentes en una clase de servicio utilizando AGENT PRIORITY. Para otorgar este privilegio, debe iniciarse la sesión como root y ejecutar el mandato siguiente:

```
usermod -K defaultpriv=basic,proc_priocntl db2user
```

En este ejemplo, se añade proc\_priocntl al conjunto de privilegios por omisión del usuario db2user.

Además, cuando DB2 se ejecuta en una zona de Solaris no global, el privilegio proc\_priocntl debe añadirse al conjunto de privilegios de límite de la zona. Para otorgar este privilegio a la zona, debe iniciarse la sesión como root y ejecutar el mandato siguiente:

```
global# zonecfg -z db2zone  
zonecfg:db2zone> set limitpriv="default,proc_priocntl"
```

En este ejemplo, se añade proc\_priocntl al conjunto de privilegios de límite de la zona db2zone.

En Solaris 9, no hay ningún recurso para que DB2 eleve la prioridad relativa de los agentes. Actualice a Solaris 10 o superior si desea utilizar la prioridad de agente de clase de servicio.

## Prioridad de captación previa de clases de servicio

Las captaciones previas recuperan datos del disco y los almacenan en agrupaciones de almacenamientos intermedios para que los agentes puedan acceder a ellos rápidamente. A cada superclase y subclase de servicio se le puede asignar una prioridad de captación previa diferente.

Los agentes envían peticiones de lectura anticipada a la cola de captación previa de la base de datos. Las captaciones previas toman estas peticiones de lectura anticipada de la cola, y luego recuperan los datos en las agrupaciones de almacenamientos intermedios. Cuando un agente requiere datos específicos,

primero comprueba si en las agrupaciones de almacenamientos intermedios los datos están disponibles. Si no lo están, el agente recupera los datos del disco. Las captaciones previas llevan a cabo grandes operaciones de E/S de disco, que libera a los agentes de llevar a cabo un trabajo de cálculo en paralelo.

Las peticiones de captación previa de cualquier conexión dirigida a una clase de servicio se procesan según la prioridad de captación previa asignada para la clase de servicio. Cada clase de servicio puede estar asociada a una de estas tres prioridades de captación previa: alta, media o baja. La prioridad de captación previa de una clase de servicio se especifica con la palabra clave PREFETCH PRIORITY en la sentencia CREATE o ALTER SERVICE CLASS.

Si especifica DEFAULT para una superclase de servicio, se establece una prioridad media de captación previa para la superclase de servicio. Puede especificar una prioridad de captación previa diferente para cualquier subclase de servicio en la superclase de servicio, pero si utiliza la prioridad de captación previa por omisión para la subclase de servicio, ésta heredará su valor de prioridad de captación previa de la superclase de servicio correspondiente.

Las peticiones de captación previa de alta prioridad se procesan antes que las peticiones de captación previa de prioridad media, las cuales, a su vez, se procesan antes que las peticiones de captación previa de prioridad baja. La prioridad de captación previa afecta al orden en el que se procesan las peticiones de captación previa, pero no afecta a la velocidad a la que se procesan.

## **Prioridad de agrupación de almacenamientos intermedios de clases de servicio**

El establecimiento de la prioridad de agrupación de almacenamientos intermedios de las clases de servicio le permite influir en la proporción de páginas de la agrupación de almacenamientos intermedios que las actividades pueden ocupar en una clase de servicio determinada, lo que puede mejorar la productividad y el rendimiento de las actividades en esa clase de servicio.

Puede asociar cada clase de servicio de DB2 a una prioridad de agrupación de almacenamientos intermedios relativa, que controla la probabilidad de intercambio de las páginas que las actividades de una clase de servicio captan previamente en la agrupación de almacenamientos intermedios. El incremento de la prioridad de agrupación de almacenamientos intermedios aumenta potencialmente la proporción de páginas que los agentes de una clase de servicio en particular utilizan.

Si no especifica una prioridad de agrupación de almacenamientos intermedios, o si especifica BUFFERPOOL PRIORITY DEFAULT, la prioridad de agrupación de almacenamientos intermedios que se asigna a la clase de servicio es DEFAULT. Para las superclases de servicio, DEFAULT se correlaciona con el valor LOW; para las subclases de servicio, DEFAULT se correlaciona con el valor de la prioridad de agrupación de almacenamientos intermedios de la superclase de servicio padre. Todas las subclases por omisión tienen la prioridad de agrupación de almacenamientos intermedios DEFAULT, que no puede cambiarse.

Al actualizar desde una versión anterior de DB2, la prioridad de agrupación de almacenamientos intermedios de las clases de servicio existentes se establece en DEFAULT.

## Reconocimiento de los beneficios de establecer una prioridad de agrupación de almacenamientos intermedios

Es más probable que pueda reconocer las ventajas en el rendimiento que supone establecer la prioridad de agrupación de almacenamientos intermedios para una clase de servicio si existe una cantidad de contención razonable en la agrupación de almacenamientos intermedios. Una contención de agrupación de almacenamientos intermedios que muestre una tasa de aciertos global del 85% o menos posiblemente obtendrá el mayor número de beneficios. Si la tasa de aciertos global excede del 90%, probablemente no existe una contención de agrupación de almacenamientos intermedios sustancial con la que empezar, y el establecimiento de la prioridad de agrupación de almacenamientos intermedios ofrecerá menos o pocos beneficios en la mayoría de los casos. Los beneficios que podrá reconocer dependen del tipo de carga de trabajo que el servidor de datos ejecuta.

Para algunas cargas de trabajo, el establecimiento de la prioridad de agrupación de almacenamientos intermedios es más efectivo si también activa la limpieza de páginas proactiva. Esto es así porque los valores de prioridad de agrupación de almacenamientos intermedios sólo son efectivos para las páginas no sucias y la limpieza de páginas proactiva es más agresiva en cuanto a la grabación de páginas sucias en disco. Tenga en cuenta que sólo debe activar la limpieza de páginas proactiva si ésta ofrece beneficios para el rendimiento.

Si utiliza la limpieza de páginas asíncrona (también denominada limpieza de páginas clásica), el establecimiento del parámetro de configuración de base de datos `chngpgs_thresh` en un valor más bajo posiblemente ofrecerá el mismo efecto que incrementar la efectividad de los valores de prioridad de agrupación de almacenamientos intermedios porque un valor bajo para este parámetro también garantiza que existirán suficientes páginas limpias en la agrupación de almacenamientos intermedios.

Es posible que los efectos positivos del establecimiento de la prioridad de agrupación de almacenamientos intermedios se vean eclipsados por los efectos de la captación previa, con o sin establecimiento de prioridad de captación previa, si existe una cantidad razonable de captación previa en ejecución. Por ejemplo, si define una clase de servicio con una prioridad de agrupación de almacenamientos intermedios alta en la que existe poca captación previa, la ventaja efectiva de este valor de prioridad de agrupación de almacenamientos intermedios podría ser menor en comparación con una clase de servicio que tiene una prioridad de agrupación de almacenamientos intermedios baja pero donde las actividades realizan una cantidad significativa de captación previa. Debido a los beneficios de la captación previa, las actividades de la clase de servicio con una prioridad de agrupación de almacenamientos intermedios baja incluso podrían superar el rendimiento de las actividades de la clase de servicio de prioridad de agrupación de almacenamientos intermedios alta. Sin embargo, el establecimiento de la prioridad de agrupación de almacenamientos intermedios todavía es capaz de complementar la estrategia de gestión de carga de trabajo en determinadas circunstancias, y debería utilizarla.

## Estados de las conexiones y de las actividades en una clase de servicio

Las clases de servicio recopilan las estadísticas de conexiones de cada clase de servicio. Puede comprobar qué conexiones y qué actividades hay en una clase de servicio, y el estado de la conexión o de la actividad.

## Estados de una conexión

A continuación se muestran los estados posibles de una conexión en una clase de servicio:

### CONNECTED

La conexión se ha conectado correctamente a la base de datos, pero todavía no se ha asociado a su carga de trabajo ni a la superclase de servicio.

### DECOUPLED

La conexión no tiene un agente coordinador asociado (caso de concentrador).

### DISCONNECTPEND

La conexión se está desconectando de la base de datos.

### FORCED

La conexión se ha forzado.

### INTERRUPTED

La conexión se ha interrumpido.

### MAPPED

La conexión se ha correlacionado con una carga de trabajo y se ha unido a una superclase de servicio. La conexión ahora puede emitir actividades para su ejecución.

### QUEUED

Una conexión de DB2 o un umbral de actividad ha puesto en cola al agente coordinador de la conexión que ofrece colocación en cola, como el umbral CONCURRENTDBCOORDACTIVITIES. En un entorno de base de datos de varios miembros, este estado puede indicar que el agente coordinador ha convertido un RPC en otro miembro a fin de obtener tíquets de umbral y aún no ha recibido una respuesta.

### TRANSIENT

La conexión está intentando unirse a una clase de servicio que ha alcanzado su umbral de conexiones. La conexión se ha puesto en cola para unirse a la clase de servicio. Si la clase de servicio no infringe su umbral de conexiones, la conexión se une a la clase de servicio. Una conexión en estado transitorio no puede emitir actividades para su ejecución.

### TERMINATING

La conexión ha recibido una restauración de la conexión desde el cliente, o está finalizando debido a una condición de error o a una acción de forzar.

### UOWEXEC

La conexión está procesando una petición.

### UOWWAIT

La conexión está esperando una petición del cliente.

## Estados de una actividad

A continuación se muestran los estados posibles de una actividad en una clase de servicio:

### CANCEL\_PENDING

Si cancela una actividad que no tiene ningún agente trabajando activamente en una petición para la actividad, ésta se coloca en estado CANCEL\_PENDING y se cancela cuando se reciba la siguiente petición.

**EXECUTING**

La actividad se está ejecutando.

**IDLE** No hay ningún agente que procese activamente una petición para la actividad.

**INITIALIZING**

Se ha creado una actividad y se está preparando para su ejecución.

**QUEUED**

La actividad no puede ejecutarse debido a un umbral de simultaneidad en la base de datos o en la clase de servicio. La actividad estará en cola hasta que se permita su ejecución.

**TERMINATING**

La actividad está finalizando.

**UNKNOWN**

El estado de la actividad es desconocido.

## Entidades de nivel del sistema cuya gestión no realizan las clases de servicio

Las clases de servicio se utilizan para supervisar y controlar objetos en el nivel de la base de datos. Sin embargo, no todas las entidades de DB2 trabajan directamente en una base de datos.

Puesto que las clases de servicio trabajan en una base de datos y se almacenan en las tablas de catálogo de la base de datos, las clases de servicio no pueden realizar la gestión de las entidades que no trabajan en una base de datos. Las entidades de nivel del sistema, como el controlador del sistema y los daemons de supervisión de salud, trabajan en el nivel de la instancia y no están directamente asociadas con ninguna base de datos. Las clases de servicio tampoco realizan la gestión de los agentes que realizan conexiones de instancias y conexiones de pasarela. Puesto que los agentes de conexión de instancias y pasarelas no trabajan en una base de datos, las clases de servicio no realizan la gestión.

La lista siguiente es una lista parcial de entidades que no trabajan en una base de datos y cuya gestión no realizan las clases de servicio:

- Controladores del sistema de DB2 (db2sysc)
- Receptores IPC (db2ipccm)
- Receptores TCP (db2tccpm)
- Daemons FCM (db2fcms, db2fcmr)
- Agentes de resincronización de DB2 (db2resync)
- Agentes desocupados (agentes sin asociación de base de datos)
- Agentes de conexión de instancias
- Agentes de pasarelas
- Las demás EDU de nivel de instancia

## Creación de una clase de servicio

Para crear superclases de servicio y las subclases de servicio correspondientes, utilice la sentencia de DDL CREATE SERVICE CLASS.

## Antes de empezar

Para crear una clase de servicio, necesita la autorización WLMADM o DBADM.

Consulte también los temas siguientes para otros requisitos:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Apéndice A, “Normas de denominación generales”, en la página 545

## Procedimiento

Para crear una clase de servicio:

1. Especifique una o más de las propiedades siguientes para la clase de servicio en la sentencia CREATE SERVICE CLASS:

- Especifique el nombre de la clase de servicio:

**Nota:** Una vez establecido, el nombre de la clase de servicio no puede cambiarse.

- Si está creando una superclase de servicio, el nombre debe ser exclusivo entre todas las superclases de servicio de la base de datos.

Cuando se crea una superclase de servicio, se crea automáticamente su subclase de servicio asociada por omisión. Solo después de crear una superclase de servicio podrá crear otras subclases de servicio bajo ella.

- Si va a crear una subclase de servicio, el nombre debe ser exclusivo entre todas las subclases de servicio de la superclase de servicio. Una subclase de servicio no puede tener el mismo nombre que su superclase de servicio.
- Si va a crear una subclase de servicio, especifique el nombre de la superclase de servicio padre. Después de crear una subclase de servicio bajo una superclase de servicio, no podrá asociarse con una superclase de servicio diferente.
- Especifique la prioridad de la agrupación de almacenamientos intermedios de la clase de servicio que afecta a la probabilidad que tienen las páginas capturadas por las actividades de la clase de servicio de ser intercambiadas. Para las superclases de servicio, el valor DEFAULT se correlaciona internamente con LOW. Las subclases de servicio establecidas como DEFAULT heredan la prioridad de la agrupación de almacenamientos intermedios de sus superclases padre.
- Especifique la prioridad de captación previa. Puede especificar la prioridad con la que los agentes de la clase de servicio pueden enviar sus peticiones de captación previa. En función del valor que especifique, las peticiones de captación previa se dirigen a las colas de captación previa de prioridad alta, media o baja. La prioridad de captación previa por omisión es media.
- Especifique la cadena del correlacionador de salida si desea asociar la clase de servicio de DB2 con una clase de AIX o una clase de Linux. Un valor nulo indica que no hay asociación con el gestor de carga de trabajo del sistema principal.

Si se define el correlacionador de salida, todas las hebras de la clase de servicio de DB2 se asocian al gestor de carga de trabajo del sistema operativo empleando el correlacionador de salida cuando comienza la siguiente actividad.

Si el correlacionador de salida se establece con el valor NONE para una subclase de servicio y también se especifica para la superclase de servicio

asociada, la subclase de servicio hereda el correlacionador de salida especificado para su superclase de servicio.

- Especifique los datos de actividad que va a recopilar. Cuando se habilita la recopilación de datos de actividad, se envía información sobre una actividad desde el miembro coordinador al supervisor de sucesos aplicables al final de la actividad. Puede grabar los datos en el supervisor de sucesos que incluya información sobre la sentencia que se ejecutó, su entorno de compilación y cualquier valor de datos de entrada aplicable. También puede especificar que no se recopile ningún dato de actividad. Por omisión, no se recopila ningún dato de actividad.
  - Información de actividad agregada recopilada. La información de la actividad agregada que se utiliza para la clase de servicio solo cambia después de confirmar la operación CREATE SERVICE CLASS.
  - El tipo de métrica de petición para recopilar peticiones enviadas por una conexión que está asociada con la superclase de servicio que ha especificado. Por omisión, la métrica básica se recopila siempre para actividades asociadas con la carga de trabajo.
  - Las plantillas de histograma que debe utilizar la subclase de servicio como plantillas para sus histogramas. Las plantillas de histograma especificadas están reflejadas en la vista SYSCAT.HISTOGRAMTEMPLATEUSE. Para obtener más información acerca de histogramas y plantillas de histograma, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.
  - Especifique si la clase de servicio está habilitada o inhabilitada.
    - Si se crea una clase de servicio como habilitada (el valor por omisión), las conexiones y las actividades podrán correlacionarse con la clase de servicio. Si se crea una clase de servicio como inhabilitada, se rechazarán las nuevas conexiones y actividades correlacionadas con ella.
    - Si crea una superclase de servicio como inhabilitada, todas las subclases de servicio que asocie con esta superclase de servicio se comportan como si estuvieran inhabilitadas, aunque se muestren como habilitadas cuando solicite la vista SYSCAT.SERVICECLASSES.
2. Confirme los cambios. Cuando confirme los cambios, la clase de servicio se añade a la vista SYSCAT.SERVICECLASSES.

## Alteración de una clase de servicio

Si desea cambiar la definición de una clase de servicio, utilice la sentencia ALTER SERVICE CLASS.

### Antes de empezar

Para alterar una clase de servicio, necesita la autorización SQLADM, WLMADM o DBADM. Para especificar una cláusula que no sea COLLECT, el ID de autorización debe incluir la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

### Acercas de esta tarea

Las actividades que ya han obtenido recursos y se están ejecutando generalmente no se ven afectadas por la sentencia ALTER. Estas actividades mantendrán sus recursos y se ejecutarán hasta su finalización. (Tenga en cuenta que los cambios

realizados en las cuotas de CPU o en los límites de CPU de una clase de servicio que se realizan a través de la sentencia ALTER SERVICE CLASS entran en vigor de forma inmediata, incluso cuando las actividades ya se están ejecutando.) Sin embargo, si se envía una petición de subagente a un miembro remoto durante la operación ALTER SERVICE CLASS, la definición de clase de servicio que observa el agente coordinador y el subagente puede diferir. Tenga en consideración este ejemplo, donde la prioridad de captación previa para la clase de servicio está inicialmente establecida en MEDIUM:

*Tabla 28. Diferencias entre las vistas de un agente coordinador y un subagente de una clase de servicio alterada*

Orden del suceso	Conexión 1	Conexión 2
1	El agente coordinador envía una petición al miembro remoto (la prioridad de captación previa de la clase de servicio se estableció anteriormente con el valor MEDIUM (media)).	
2		Se ha emitido ALTER SERVICE CLASS; se ha establecido la prioridad de captación previa como HIGH (alta)
3		Se ha emitido COMMIT (la propiedad de clase de servicio alterada se confirma en el miembro del catálogo y se carga en memoria en todos los miembros de bases de datos).
4	El subagente remoto recibe la petición. En este momento, el subagente establece la nueva prioridad de captación previa HIGH (alta) para la definición de clase de servicio.	

Esta situación que se describe en la tabla anterior es temporal, y solo afecta a las conexiones que emiten peticiones de subagente durante la operación ALTER SERVICE CLASS. Todas las conexiones nuevas observarán la definición de clase de servicio actualizada con la prioridad de captación previa HIGH (alta).

## Procedimiento

Para alterar una clase de servicio:

1. Especifique una o más de las propiedades siguientes para la clase de servicio en la sentencia ALTER SERVICE CLASS:
  - Especifique si la clase de servicio está habilitada o inhabilitada. Si cambia una clase de servicio de habilitada a inhabilitada, las conexiones o actividades existentes permanecerán con la clase de servicio y seguirán utilizando los recursos anteriormente asignados hasta que finalicen. Puede

inhabilitar una clase de servicio si el trabajo que entra en ella desborda la capacidad del sistema, o si desea rechazar todo el trabajo que entre en la clase de servicio.

Cuando se inhabilita una superclase de servicio, ocurre lo siguiente:

- a. La superclase de servicio se inhabilita.
- b. Las subclases de servicio correspondientes se inhabilitan.

Las subclases de servicio solo se inhabilitan si la superclase de servicio también se inhabilita. Cuando se habilita la superclase de servicio, las subclases de servicio recuperan sus estados anteriores tal como está definido en la tabla de catálogo.

Cuando se inhabilita una subclase de servicio, su superclase de servicio no se ve afectada, ni ninguna otra subclase de servicio asociada con la superclase de servicio.

No se puede inhabilitar explícitamente una subclase de servicio por omisión. Para evitar la ejecución de nuevas peticiones bajo una subclase de servicio por omisión, debe inhabilitar la superclase de servicio asociada.

- Especifique la prioridad de agente correspondiente a la clase de servicio. Cuando la prioridad de agente se establece con el valor DEFAULT, a los agentes de la clase de servicio se les asigna la misma prioridad que el sistema operativo asigna a todas las hebras de DB2. Si establece el parámetro AGENT PRIORITY con un valor que no sea DEFAULT, las hebras de agente se establecen con una prioridad igual a la prioridad por omisión, más el valor establecido cuando se inicia la actividad siguiente. Por ejemplo, si la prioridad por omisión es 20 y establece la prioridad de agente en -10, la prioridad de agente resultante se establece en  $20 + (-10) = 10$ .

En la vista de catálogo SYSCAT.SERVICECLASSES, una prioridad de agente de DEFAULT se representa como -32768.

En Linux y UNIX, los valores válidos son de -20 a 20 (un valor negativo indica una prioridad relativa más alta). En sistemas operativos de Windows los valores válidos son de -6 a 6 (un valor negativo indica una prioridad relativa más baja)

- Especifique la prioridad de captación previa. Puede especificar la prioridad con la que los agentes de la clase de servicio pueden enviar sus peticiones de captación previa. En función del valor que especifique, las peticiones de captación previa se dirigen a las colas de captación previa de prioridad alta, media o baja. La prioridad de captación previa por omisión es media. Si la prioridad de captación previa se altera después de enviar una petición de captación previa, la petición no cambiará su prioridad.
- Especifique la prioridad de la agrupación de almacenamientos intermedios de la clase de servicio que afecta a la probabilidad que tienen las páginas capturadas por las actividades de la clase de servicio de ser intercambiadas. Para las superclases de servicio, el valor DEFAULT se correlaciona internamente con LOW. Las subclases de servicio establecidas como DEFAULT heredan la prioridad de la agrupación de almacenamientos intermedios de sus superclases padre.
- Especifique la cadena del correlacionador de salida si desea asociar la clase de servicio de DB2 con una clase de AIX o una clase de Linux. Un valor nulo indica que no hay asociación con el gestor de carga de trabajo del sistema principal.

Si el valor no nulo del correlacionador de salida cambia a un valor nulo, todas las hebras de la clase de servicio de DB2 dejarán de estar asociadas con el gestor de carga de trabajo del sistema operativo cuando comience la siguiente actividad.

Si el correlacionador de salida se establece con el valor NONE para una subclase de servicio y también se especifica para la superclase de servicio asociada, la subclase de servicio hereda el correlacionador de salida especificado para su superclase de servicio.

Si una superclase de servicio utiliza un correlacionador de salida, la prioridad de agente de la superclase de servicio deberá establecerse con el valor por omisión.

Si una subclase de servicio utiliza un correlacionador de salida (ya sea explícitamente como parte de la definición de subclase de servicio o implícitamente a través de herencia de la superclase de servicio), la prioridad de agente de la subclase de servicio debe establecerse con el valor por omisión.

- Especifique si ha de añadirse o descartarse un espacio de tablas temporal del sistema de la lista de espacios de tablas temporales preferidos para la subclase de servicio especificada.
  - Especifique los datos de actividad que va a recopilar. Cuando se habilita la recopilación de datos de actividad, se envía información sobre una actividad desde el miembro coordinador al supervisor de sucesos aplicables al final de la actividad. Puede grabar los datos en el supervisor de sucesos que incluya información sobre la sentencia que se ejecutó, su entorno de compilación y cualquier valor de datos de entrada aplicable. También puede especificar que no se recopile ningún dato de actividad. Por omisión, no se recopila ningún dato de actividad.
  - Información de actividad agregada recopilada. La información de la actividad agregada que se utiliza para la clase de servicio solo cambia después de confirmar la operación ALTER SERVICE CLASS.
  - La supervisión del nivel de la recopilación de métrica de petición para peticiones enviadas por conexiones correlacionadas con una subclase bajo la superclase de servicio especificada. Tenga en cuenta que la configuración efectiva de la recopilación para peticiones que se ejecutan bajo una superclase de servicio es la combinación entre el nivel de recopilación de clase de servicio y el parámetro de configuración de la base de datos **mon\_req\_metrics**.
  - Si se alterarán las plantillas del histograma utilizadas por una subclase de servicio que tiene una recopilación de datos de actividades agregadas habilitada utilizando COLLECT AGGREGATE ACTIVITY DATA o si se agregará la recopilación de datos de petición utilizando COLLECT AGGREGATE REQUEST DATA. La actualización de las plantillas del histograma utilizadas por una subclase de servicio actualizará las filas correspondientes en la vista SYSCAT.HISTOGRAMTEMPLATEUSE que muestra las plantillas del histograma a las que una clase de servicio o una acción de trabajo hace referencia. Para obtener más información acerca de histogramas y plantillas de histograma, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.
2. Confirme los cambios. Cuando confirma los cambios, la clase de servicio se actualiza en la vista SYSCAT.SERVICECLASSES.

## Descarte de una clase de servicio

Para descartar clases de servicio, utilice la sentencia de DDL DROP SERVICE CLASS.

### Antes de empezar

Para descartar una clase de servicio, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

No se pueden descartar superclases de servicio por omisión (SYSDEFAULTUSERCLASS, SYSDEFAULTMAINTENANCECLASS y SYSDEFAULTSYSTEMCLASS) ni sus subclases de servicio asociadas. La única forma de descartar superclases de servicio por omisión y sus subclases de servicio asociadas es descartando la base de datos.

No puede descartar una clase de servicio que haya definido si no se cumple alguna de estas condiciones:

- Está habilitada
- Contiene subclases de servicio definidas por el usuario
- Alguna carga de trabajo, acción de trabajo o umbral hace referencia a ella
- Todavía hay una aparición de carga de trabajo que hace referencia a ella
- Hay una conexión o actividad está actualmente correlacionada a la clase de servicio
- Si la clase de servicio está establecida como destino de una acción REMAP ACTIVITY.

### Procedimiento

Para descartar una clase de servicio:

1. Cambie las cargas de trabajo de manera que no apunten a la clase de servicio, o inhabilite las definiciones de carga de trabajo. Como alternativa, utilice la sentencia DROP WORKLOAD para descartar todas las cargas de trabajo asociadas con la clase de servicio. Emita una sentencia COMMIT después de descartar cada carga de trabajo.

Las actividades que ya se están ejecutando bajo la clase de servicio lo seguirán haciendo. Puede obtener una lista de los agentes que están correlacionados actualmente con una clase de servicio mediante la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS. Si no desea que estas actividades finalicen, puede utilizar el identificador de aplicación que devuelve la función de tabla y emplear el mandato **FORCE APPLICATION** para forzar la extracción de estas aplicaciones de la base de datos.

2. Descarte todas las acciones de trabajo aplicables que estén asociadas con la clase de servicio que desea descartar:
  - Si va a descartar una superclase de servicio, y hay un conjunto de acciones de trabajo asociado con ella, inhabilite y descarte ese conjunto de acciones de trabajo. Utilice la sentencia ALTER WORK ACTION SET para inhabilitar un conjunto de acciones de trabajo, y la sentencia DROP WORK ACTION SET para descartar un conjunto de acciones de trabajo. Emita una sentencia COMMIT después de descartar el conjunto de acciones de trabajo.

- Si va a descartar una subclase de servicio y una acción de trabajo se correlaciona con esa subclase de servicio, descarte la acción de trabajo utilizando la cláusula DROP WORK ACTION de la sentencia ALTER WORK ACTION SET. Como alternativa, descarte el conjunto de acciones de trabajo que contiene la acción de trabajo que se correlaciona con la subclase de servicio a través de la sentencia DROP WORK ACTION SET. Emita una sentencia COMMIT después de descartar cada acción de trabajo, o después de descartar el conjunto de acciones de trabajo.
3. Inhabilite y descarte todos los umbrales asociados con la clase de servicio que desea descartar. Emita una sentencia COMMIT después de descartar cada umbral. Utilice la sentencia ALTER THRESHOLD para inhabilitar un umbral, y la sentencia DROP THRESHOLD para descartar el umbral.
  4. En función del objeto que esté descartando, haga lo siguiente:
    - Si va a descartar una subclase de servicio, utilice la sentencia DROP SERVICE CLASS para descartar la subclase de servicio.
    - Si va a descartar una superclase de servicio, utilice la sentencia DROP SERVICE CLASS para descartar todas las subclases de servicio asociadas con la superclase de servicio, y emita una sentencia COMMIT después de descartar cada subclase de servicio. Luego emita la sentencia DROP SERVICE CLASS para descartar la superclase de servicio.

**Nota:** La subclase de servicio por omisión de la superclase de servicio no puede descartarse manualmente. La subclase de servicio por omisión de una superclase de servicio se descarta cuando lo hace la superclase de servicio.

5. Inhabilite la clase de servicio mediante la sentencia ALTER SERVICE CLASS. Si va a descartar una superclase de servicio, esta acción inhabilita todas las subclases de servicio asociadas con la superclase de servicio. El hecho de inhabilitar una clase de servicio impide que otras actividades se asocien con ella. Una vez inhabilitada la clase de servicio, emita una sentencia COMMIT.
6. Confirme los cambios. Cuando confirme los cambios, la clase de servicio se elimina de la vista SYSCAT.SERVICECLASSES.

## Ejemplo: Utilización de clases de servicio

En el ejemplo siguiente se muestra cómo utilizar clases de servicio para controlar la carga de trabajo de la base de datos.

Este ejemplo tiene lugar en la empresa ficticia International Beer Emporium. International Beer Emporium es una empresa de tamaño medio compuesta por cinco departamentos principales: Ventas, Contabilidad, Ingeniería, Pruebas y Producción. Los cinco departamentos comparten la misma base de datos del catálogo de productos.

### Implementación inicial de una solución de la gestión de carga de trabajo de DB2

La base de datos del catálogo de productos se ejecuta correctamente la mayor parte del tiempo. Sin embargo, a veces los usuarios se quejan de que sus aplicaciones no pueden conectarse a la base de datos porque se ha superado el número máximo de conexiones. Después de actualizar DB2 a la versión 9.7, Bob, administrador de la base de datos, decide probar las clases de servicio. Bob desea conocer los patrones de la base de datos del catálogo de productos por parte de cada uno de los cinco departamentos y averiguar por qué su base de datos en ocasiones agota las conexiones. Estos son los pasos que sigue Bob para configurar las clases de servicio:

1. Primero, Bob crea superclases servicio por cada uno de los departamentos (la subclase de servicio por omisión se crea automáticamente por cada superclase de servicio):
  - Se crea SALES para el departamento de ventas:  
CREATE SERVICE CLASS SALES
  - Se crea ACCOUNTING para el departamento de contabilidad:  
CREATE SERVICE CLASS ACCOUNTING
  - Se crea ENGINEERING para el departamento de ingeniería:  
CREATE SERVICE CLASS ENGINEERING
  - Se crea TESTING para el departamento de pruebas:  
CREATE SERVICE CLASS TESTING
  - Se crea PRODUCTION para el departamento de producción:  
CREATE SERVICE CLASS PRODUCTION
2. Bob crea grupos de usuarios de sesión con ID de autorización apropiados para cada uno de los departamentos:
  - Se crea un grupo de usuarios de sesión con el ID de autorización SALESGRP. Este grupo incluye los ID de autorización de todos los usuarios del departamento de ventas.
  - Se crea un grupo de usuarios de sesión con el ID de autorización ACCTNGRP. Este grupo incluye los ID de autorización de todos los usuarios del departamento de contabilidad.
  - Se crea un grupo de usuarios de sesión con el ID de autorización ENGINGRP. Este grupo incluye los ID de autorización de todos los usuarios del departamento de ingeniería.
  - Se crea un grupo de usuarios de sesión con el ID de autorización TESTGRP. Este grupo incluye los ID de autorización de todos los usuarios del departamento de pruebas.
  - Se crea un grupo de usuarios de sesión con el ID de autorización PRODGRP. Este grupo incluye los ID de autorización de todos los usuarios del departamento de producción.
3. Bob crea cargas de trabajo para correlacionar las conexiones de cada grupo con la clase de servicio asociada:
  - Se crea la carga de trabajo WL\_SALES con su grupo de usuarios de sesión definido en SALESGRP. WL\_SALES correlaciona sus conexiones con la superclase de servicio SALES:  
CREATE WORKLOAD WL\_SALES SESSION\_USER GROUP ('SALESGRP')  
SERVICE CLASS SALES
  - Se crea la carga de trabajo WL\_ACCOUNTING con su grupo de usuarios de sesión definido en ACCTNGRP. WL\_ACCOUNTING correlaciona sus conexiones con la superclase de servicio ACCOUNTING:  
CREATE WORKLOAD WL\_ACCOUNTING SESSION\_USER GROUP ('ACCTNGRP')  
SERVICE CLASS ACCOUNTING
  - Se crea la carga de trabajo WL\_ENGINEERING con su grupo de usuarios de sesión definido en ENGINGRP. WL\_ENGINEERING correlaciona sus conexiones con la clase de servicio ENGINEERING:  
CREATE WORKLOAD WL\_ENGINEERING SESSION\_USER GROUP ('ENGINGRP')  
SERVICE CLASS ENGINEERING
  - Se crea la carga de trabajo WL\_TEST con su grupo de usuarios de sesión definido en TESTGRP. WL\_TEST correlaciona sus conexiones con la clase de servicio TESTING:

```
CREATE WORKLOAD WL_TEST SESSION_USER GROUP ('TESTGRP')
SERVICE CLASS TESTING
```

- Se crea la carga de trabajo WL\_PRODUCTION con su grupo de usuarios de sesión definido en PRODGRP. WL\_PRODUCTION correlaciona sus conexiones con la clase de servicio PRODUCTION:

```
CREATE WORKLOAD WL_PRODUCTION SESSION_USER GROUP ('PRODGRP')
SERVICE CLASS PRODUCTION
```

Bob utiliza los valores de clase de servicio y carga de trabajo por omisión. Desea observar los patrones de uso de la base de datos antes de implantar controles en las clases de servicio. Las definiciones resultantes de las superclases de servicio son las siguientes:

*Tabla 29. Definiciones de clase de servicio*

Clase de servicio
SALES
ACCOUNTING
ENGINEERING
TESTING
PRODUCTION
SYSDEFAULTUSERCLASS
SYSDEFAULTMAINTENANCECLASS
SYSDEFAULTSYSTEMCLASS

Con la implementación de una solución de la gestión de carga de trabajo de DB2 como la descrita anteriormente, el trabajo de cada departamento se dirige a su propia superclase de servicio. El trabajo de departamentos no contemplados específicamente se correlaciona con la superclase de servicio por omisión SYSDEFAULTUSERCLASS. Con esta configuración, Bob puede supervisar el trabajo en cada una de las clases de servicio para determinar el patrón de uso de la base de datos por parte de los departamentos.

## Primer refinamiento de la implementación de la gestión de carga de trabajo de DB2

Después del pico de conexiones más reciente, Bob consulta las estadísticas de superclases de servicio utilizando la función de tabla WLM\_GET\_SERVICE\_SUPERCLASS\_STATS y examina la marca de límite superior de conexiones por cada superclase de servicio. Bob descubre que la marca de límite superior de conexiones en todos los departamentos salvo el de pruebas se aproxima a 100. Sin embargo, las estadísticas del departamento de pruebas muestran que una vez que el equipo de pruebas estableció más de 800 conexiones.

Una vez al mes, el departamento de pruebas realiza su prueba intensiva de productos. En ese momento, el departamento establece hasta 1000 conexiones simultáneas. Puesto que el parámetro de configuración del gestor de bases de datos **max\_connections** está establecido en 1000, el departamento de pruebas utiliza la mayoría de las conexiones disponibles en la base de datos. Cuando el sistema tiene 1000, todas las conexiones posteriores se rechazan.

Debido a restricciones de memoria en el sistema, los valores de configuración de **max\_connections** y **maxagents** no pueden aumentarse en el servidor de datos para permitir más conexiones.

Para impedir que el departamento de pruebas utilice todas las conexiones, Bob decide limitar el número de conexiones del departamento de pruebas y garantizar que cada uno de los otros cuatro departamentos pueda obtener suficientes conexiones con la base de datos para satisfacer sus objetivos empresariales.

Los otros cuatro departamentos no suelen requerir más de 150 conexiones simultáneas cada uno. Además, Bob observa que las superclases de servicio de usuario por omisión, mantenimiento por omisión y sistema por omisión rara vez contienen conexiones, por lo que decide que 100 conexiones serán suficientes para estas superclases de servicio por omisión. Después de asignar 700 conexiones (600 para los cuatro departamentos y 100 para las clases por omisión) de la agrupación **max\_connections** de las 1000 conexiones disponibles, hay disponibles 300 conexiones para el departamento de pruebas. Al limitar el departamento de pruebas a un máximo de 300, a los usuarios de los demás departamentos no se les rechazarán sus peticiones de conexión.

Para limitar el grupo de pruebas a un máximo de 300 conexiones simultáneas, Bob crea un umbral **MAXSERVICECLASSCONNECTIONS** de 300 para la clase de servicio **TESTING**.

```
CREATE THRESHOLD MAXSERVICECLASSCONNECTIONS FOR SERVICE CLASS TESTING ACTIVITIES
ENFORCEMENT DATABASE PARTITION
WHEN TOTALSCMEMBERCONNECTIONS > 300 STOP EXECUTION
```

Después de implementar este cambio, la configuración de la gestión de carga de trabajo de DB2 es la siguiente:

*Tabla 30. Configuración después de añadir un umbral para la superclase de servicio TESTING*

Clase de servicio	Umbral MAXSERVICECLASSCONNECTIONS
SALES	N/A
ACCOUNTING	N/A
ENGINEERING	N/A
TESTING	300
PRODUCTION	N/A
SYSDEFAULTUSERCLASS	N/A
SYSDEFAULTMAINTENANCECLASS	N/A

Puesto que la clase de servicio **TESTING** puede contener un máximo de sólo 300 conexiones simultáneas, se rechazarán todas las peticiones de conexión por encima de este umbral. No se aplica un umbral **MAXSERVICECLASSCONNECTIONS** en las demás clases de servicio, por lo que estas clases de servicio comparten las 700 conexiones disponibles restantes con el servidor de datos. Como no hay contención para las conexiones entre estas clases de servicio, Bob no establece umbrales de conexión para ellas.

## Segundo refinamiento de la implementación de la gestión de carga de trabajo de DB2

Aunque las conexiones de los departamentos de ventas, contabilidad, ingeniería y producción ya no se rechazan, los usuarios de estos departamentos todavía se quejan de un bajo rendimiento cuando el departamento de pruebas realiza pruebas intensivas de productos. Bob examina las consultas que ejecuta el departamento de pruebas durante el ciclo de pruebas de los productos, y descubre que las consultas contienen uniones complicadas que implican grandes cantidades de datos. Estas consultas generan una actividad de captación previa considerable, lo que impide que se procesen las peticiones de captación previa de las conexiones de otros departamentos. Bob decide bajar la prioridad de captación previa de las conexiones del departamento de pruebas, por lo que modifica la clase de servicio TESTING para establecer su prioridad de captación previa en LOW:

```
ALTER SERVICE CLASS TESTING PREFETCH PRIORITY LOW
```

La configuración de la gestión de carga de trabajo de DB2 es la siguiente:

Tabla 31. Configuración después de cambiar la prioridad de captación previa para la superclase de servicio TESTING

Clase de servicio	Umbral MAXSERVICECLASSCONNECTIONS	Prioridad de captación previa
SALES	N/A	DEFAULT
ACCOUNTING	N/A	DEFAULT
ENGINEERING	N/A	DEFAULT
TESTING	300	LOW
PRODUCTION	N/A	DEFAULT
SYSDEFAULTUSERCLASS	N/A	DEFAULT
SYSDEFAULTMAINTENANCECLASS	N/A	DEFAULT

Establecer la prioridad de captación previa de la clase de servicio TESTING en LOW hace que las peticiones de captación previa de las conexiones emitidas desde el departamento de pruebas se atiendan sólo después de procesar todas las peticiones de captación previa de los demás departamentos. Este cambio aumenta el rendimiento de las consultas de los demás departamentos y reduce el rendimiento del departamento de pruebas durante su fase de pruebas de productos.

## Tercer refinamiento de la implementación de la gestión de carga de trabajo de DB2

Después de resolver el problema de captación previa, el departamento de ingeniería comunica a Bob que necesita unas cuantas conexiones para una aplicación experimental denominada Brewmeister. Debido a que la aplicación es experimental, Bob desea asegurarse de que no consuma demasiadas conexiones de base de datos y que las consultas de la aplicación no compitan por los captadores previos cuando el sistema que esté ocupado. Para cumplir estos objetivos, crea una nueva subclase de servicio bajo la superclase de servicio ENGINEERING para la aplicación experimental y una carga de trabajo para correlacionar conexiones de la aplicación con la nueva subclase de servicio. Bob actualiza la clase de servicio y las cargas de trabajo del modo siguiente:

- Se crea la subclase de servicio EXPERIMENT bajo la superclase de servicio ENGINEERING:  
CREATE SERVICE CLASS EXPERIMENT UNDER ENGINEERING
- Se crea un umbral MAXSERVICECLASSCONNECTIONS de 50 para la subclase de servicio EXPERIMENT:  
CREATE THRESHOLD MAXSERVICECLASSCONNECTIONS FOR SERVICE CLASS EXPERIMENT UNDER ENGINEERING ACTIVITIES ENFORCEMENT DATABASE WHEN TOTALMEMBERCONNECTIONS > 50 STOP EXECUTION
- Se crea la carga de trabajo WL\_EXPERIMENT para correlacionar las conexiones de la aplicación BREWMEISTER con la subclase de servicio EXPERIMENT:  
CREATE WORKLOAD WL\_EXPERIMENT APPLNAME ('BREWMEISTER') SERVICE CLASS EXPERIMENT UNDER ENGINEERING
- La prioridad de captación previa para la subclase de servicio EXPERIMENT se establece en LOW:  
ALTER SERVICE CLASS EXPERIMENT UNDER ENGINEERING PREFETCH PRIORITY LOW

La configuración de la gestión de carga de trabajo de DB2 es la siguiente:

Tabla 32. Configuración con la subclase de servicio EXPERIMENT

Clase de servicio	Umbral MAXSERVICECLASSCONNECTIONS	Prioridad de captación previa
SALES	N/A	DEFAULT
ACCOUNTING	N/A	DEFAULT
ENGINEERING	N/A	DEFAULT
EXPERIMENT	50	LOW
TESTING	300	LOW
PRODUCTION	N/A	DEFAULT
SYSDEFAULTUSERCLASS	N/A	DEFAULT
SYSDEFAULTMAINTENANCECLASS	N/A	DEFAULT

Con esta configuración, la aplicación BREWMEISTER sólo puede mantener 50 conexiones simultáneas con la base de datos. Además, las peticiones de captación previa de esta aplicación se envían a la cola de captación previa de prioridad baja. El departamento de ingeniería puede ahora experimentar con la aplicación con toda seguridad, a sabiendas de que no sobrecargará accidentalmente el sistema de la base de datos.

## Ejemplo: Análisis de una ralentización de sistema relacionada con la clase de servicio

Si observa una ralentización del sistema (por ejemplo, algunas aplicaciones tardan mucho más de lo esperado en completarse) y no está seguro de si el problema tiene relación con la configuración de las clases de servicio, puede utilizar los datos de función de tabla para investigar y, si es necesario, corregir el problema.

En primer lugar, obtenga una visión general de lo que está sucediendo en las clases de servicio. Esta visión general debe incluir el promedio de vida útil de las actividades, el número de actividades que se han completado con normalidad en relación con las que han finalizado anormalmente, así como la marca de límite superior de actividades de coordinador simultáneas en el sistema. Para obtener esta información, puede crear una consulta general con agregación de las diferentes clases de servicio y miembros de base de datos utilizando los datos obtenidos de la

función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS. Establezca series vacías para el primer y segundo argumento, -2 (un carácter comodín) en el tercer argumento, para indicar que los datos deben recopilarse para todas las clases de servicio en todos los miembros de base de datos. Su consulta puede parecerse a la siguiente:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(SUM(COORD_ACT_COMPLETED_TOTAL)),1,13) AS ACTSCOMPLETED,
       SUBSTR(CHAR(SUM(COORD_ACT_ABORTED_TOTAL)),1,11) AS ACTSABORTED,
       SUBSTR(CHAR(MAX(CONCURRENT_ACT_TOP)),1,6) AS ACTSHW,
       CAST(CASE WHEN SUM(COORD_ACT_COMPLETED_TOTAL) = 0 THEN 0
              ELSE SUM(COORD_ACT_COMPLETED_TOTAL * COORD_ACT_LIFETIME_AVG)
                / SUM(COORD_ACT_COMPLETED_TOTAL) END / 1000 AS DECIMAL(9,3))
       AS ACTAVGLIFETIME
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS ('', '', -2)) AS SCSTATS
GROUP BY SERVICE_SUPERCLASS_NAME, SERVICE_SUBCLASS_NAME
ORDER BY SERVICE_SUPERCLASS_NAME, SERVICE_SUBCLASS_NAME
```

Supongamos que, en ocasiones anteriores, la consulta proporcionaba los resultados siguientes:

SUPERCLASS_NAME	SUBCLASS_NAME	ACTSCOMPLETED	ACTSABORTED	ACTSHW	ACTAVGLIFETIME
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 8	8	0	1	3.750
BI_APPS	SYSDEFAULTSUBCLASS 4	4	0	1	5.230
BATCH	SYSDEFAULTSUBCLASS 1	1	0	1	25.600

Los datos que devuelve esta consulta pueden ser suficientes para demostrar que existe una ralentización en la clase de servicio BI\_APPS, porque su promedio de vida útil de las actividades es significativamente mayor que de costumbre. Esta situación puede indicar que se están agotando los recursos disponibles para esa clase de servicio concreta.

Si los promedios de las clases de servicio en todos los miembros de base de datos no aíslan el problema, plantéese analizar los valores promedio en cada miembro de base de datos. Acumular el promedio de cada miembro en un promedio global puede ocultar grandes discrepancias entre miembros de base de datos. En este caso, se supone que cada miembro se está utilizando como miembro coordinador. Si esta suposición es incorrecta, el promedio de vida útil calculado en los miembros no coordinadores será cero.

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEM,
       CAST(COORD_ACT_LIFETIME_AVG / 1000 AS DECIMAL(9,3)) AS AVGLIFETIME
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS('', '', -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME
```

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	AVGLIFETIME
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 0	0	3.425
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 1	1	2.752
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 2	2	8.230
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 3	3	0.593

En este ejemplo, el miembro 2 puede estar recibiendo más trabajo que de costumbre, porque el promedio de vida útil de sus actividades es mucho mayor que el de los demás miembros de base de datos.

Hay muchas situaciones diferentes que pueden provocar una ralentización del sistema. Utilice los principios siguientes para utilizar de forma óptima la información que proporcionan las funciones de tabla de la gestión de carga de trabajo de DB2:

- Solucione cantidades grandes de conflictos de bloqueo en el nivel de la lógica de la aplicación y del entorno (nivel de aislamiento, etc.).
- Si la clase de servicio se ejecuta cerca de sus niveles de umbral (número de peticiones simultáneas, etc.), es posible que deba incrementar los umbrales.
- Si los recursos asignados a una clase de servicio se están agotando y se ha establecido OUTBOUN CORRELATOR, la correlación con las clases de servicio del sistema operativo puede ser la causa del problema (es decir, la clase de servicio del sistema operativo correspondiente a la clase de servicio no obtiene suficientes recursos de procesador).
- Una cantidad de actividades mayor de la esperada puede estar ejecutándose en la clase de servicio, que puede estar consumiendo más recursos de lo normal. Compruebe el número de actividades completadas para determinar si la cantidad de trabajo que se está realizando en la clase de servicio es razonable.
- Las actividades pueden estar pasando más tiempo en cola si se envían más actividades de las esperadas y se han definido umbrales de simultaneidad. Compruebe si el promedio de tiempo en cola de las actividades ha aumentado en la misma cantidad que el promedio de vida útil. Si han aumentado en la misma cantidad, las colas se están comportando según lo previsto; sin embargo, si la vida útil es inaceptable, plantéese asignar más recursos a esta clase de servicio y reducir el umbral de simultaneidad.

## Ejemplo: Investigación del uso de agentes por parte de una clase de servicio

La gestión de carga de trabajo de DB2 proporciona la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS, que puede utilizar para determinar la distribución relativa de agentes entre clases de servicio.

Pueden surgir situaciones en las que un grupo de usuarios o una aplicación utilicen excesivamente un recurso de servidor de datos. Por ejemplo, supongamos que un grupo de usuarios utiliza casi todos los agentes disponibles y que un usuario fuera de este grupo le transmite una queja al respecto.

El primer paso que debe tomar es determinar cuántos agentes trabajan por cada clase de servicio. Puede utilizar una consulta como la siguiente:

```
SELECT SUBSTR(AGENTS.SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(AGENTS.SERVICE_SUBCLASS_NAME,1,19) AS SUBCLASS_NAME,
       COUNT(*) AS AGENT_COUNT
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS('', '', CAST(NULL AS BIGINT), -2))
AS AGENTS
WHERE AGENT_STATE = 'ACTIVE'
GROUP BY SERVICE_SUPERCLASS_NAME, SERVICE_SUBCLASS_NAME
ORDER BY SERVICE_SUPERCLASS_NAME, SERVICE_SUBCLASS_NAME
```

SUPERCLASS_NAME	SUBCLASS_NAME	AGENT_COUNT
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	7
TEST	SYSDEFAULTSUBCLASS	20

Si concluye que una determinada clase de servicio utiliza más agentes de los que debería, puede tomar medidas para restringir el número de actividades permitidas para una carga de trabajo o una clase de servicio. También puede restringir el número de conexiones para una clase de servicio.

---

## Aplicación de controles a tipos de actividades con conjuntos de acciones de trabajo

Los conjuntos de acciones de trabajo contienen acciones de trabajo que aplican controles a actividades de un determinado tipo en una superclase de servicio específica, en una carga de trabajo específica o en la base de datos en su conjunto.

Una acción de trabajo proporciona una acción que puede aplicarse a una clase de trabajo, que representa las actividades de un determinado tipo, como las actividades LOAD o READ. Las acciones de trabajo se aplican a las actividades que se producen en la clase de trabajo a la que se aplicó la acción de trabajo antes de que la actividad empezara a ejecutarse. No obstante, si la acción de trabajo es PREVENT, no se permitirá la ejecución de la actividad y no se aplicará ninguna otra acción de trabajo.

Si aplica un conjunto de acciones de trabajo a una base de datos, existen varios tipos de acciones que puede aplicar a las actividades de una clase de trabajo, como las definiciones de umbral, la evitación de la ejecución, la recopilación de datos de actividad y el recuento de actividad. La definición de un umbral para una acción de trabajo es la acción de trabajo de base de datos más potente. Por ejemplo, quizás desea evitar que el SQL lea y devuelva más de 100.000 filas. Puede definir una única clase de trabajo para un conjunto de acciones de trabajo que identifique las sentencias READ de SQL y una acción de trabajo con un umbral que detendría la ejecución si el número de filas devuelto fuera superior a 100.000. Para obtener información acerca de las acciones posibles, consulte “Acciones de trabajo y el dominio de conjunto de acciones de trabajo” en la página 117.

Si aplica un conjunto de acciones de trabajo a una carga de trabajo, los distintos tipos de acciones que puede aplicar a las actividades incluyen la definición de umbrales, la evitación de la ejecución, la recopilación de datos de actividad y de datos de actividades agregadas y el recuento de las actividades.

Si define el conjunto de acciones de trabajo para una superclase de servicio, los distintos tipos de acciones que puede aplicar a las actividades incluyen la correlación de las actividades con una subclase de servicio, la evitación de la ejecución, la recopilación de datos de actividad o de actividad agregada y el recuento de las actividades. Por lo general, la acción de trabajo correlaciona una actividad con una subclase de servicio y tiene definidos umbrales en la subclase como ayuda para gestionar la actividad.

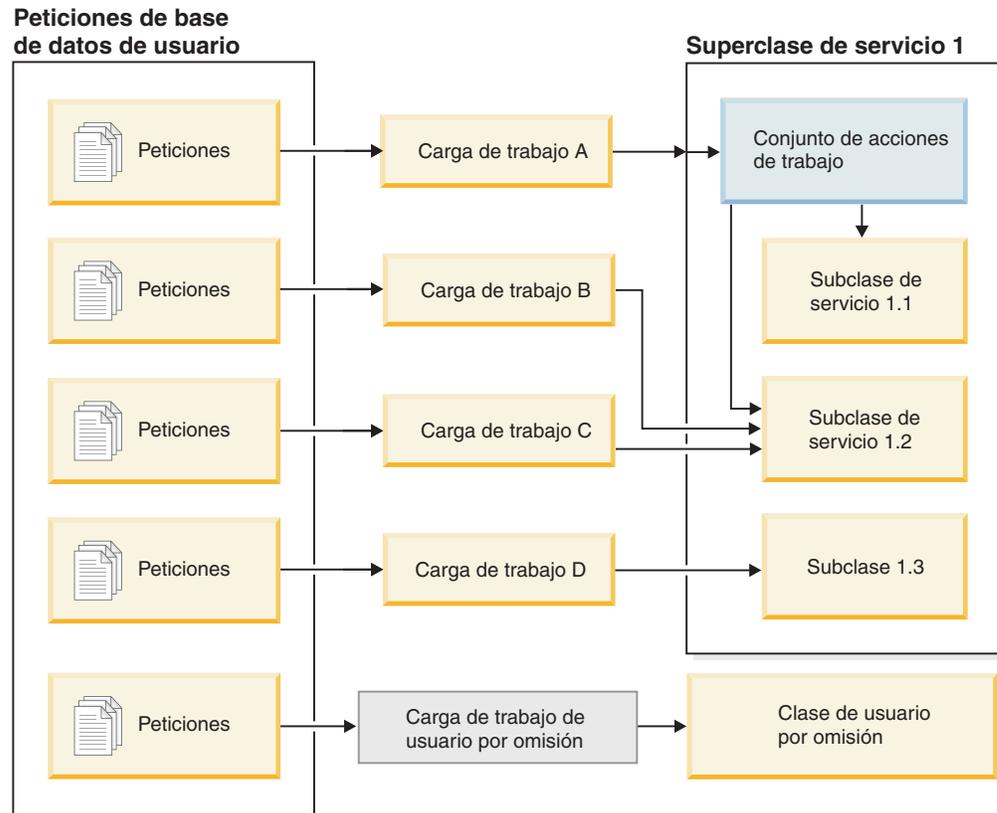


Figura 15. Correlación de un conjunto de acciones de trabajo para una superclase de servicio

## Cómo las clases de trabajo, los conjuntos de clases de trabajo, las acciones de trabajo y los conjuntos de acciones de trabajo funcionan conjuntamente y se asocian con otros objetos de DB2

Las clases de trabajo y las acciones de trabajo funcionan conjuntamente para aplicar acciones específicas a tipos de actividades específicos. El mejor modo de describir este funcionamiento es mediante un ejemplo.

En el diagrama siguiente se muestra una vista global de cómo las clases de trabajo, los conjuntos de clases de trabajo, las acciones de trabajo y los conjuntos de acciones de trabajo funcionan conjuntamente y se asocian con otros objetos de DB2.

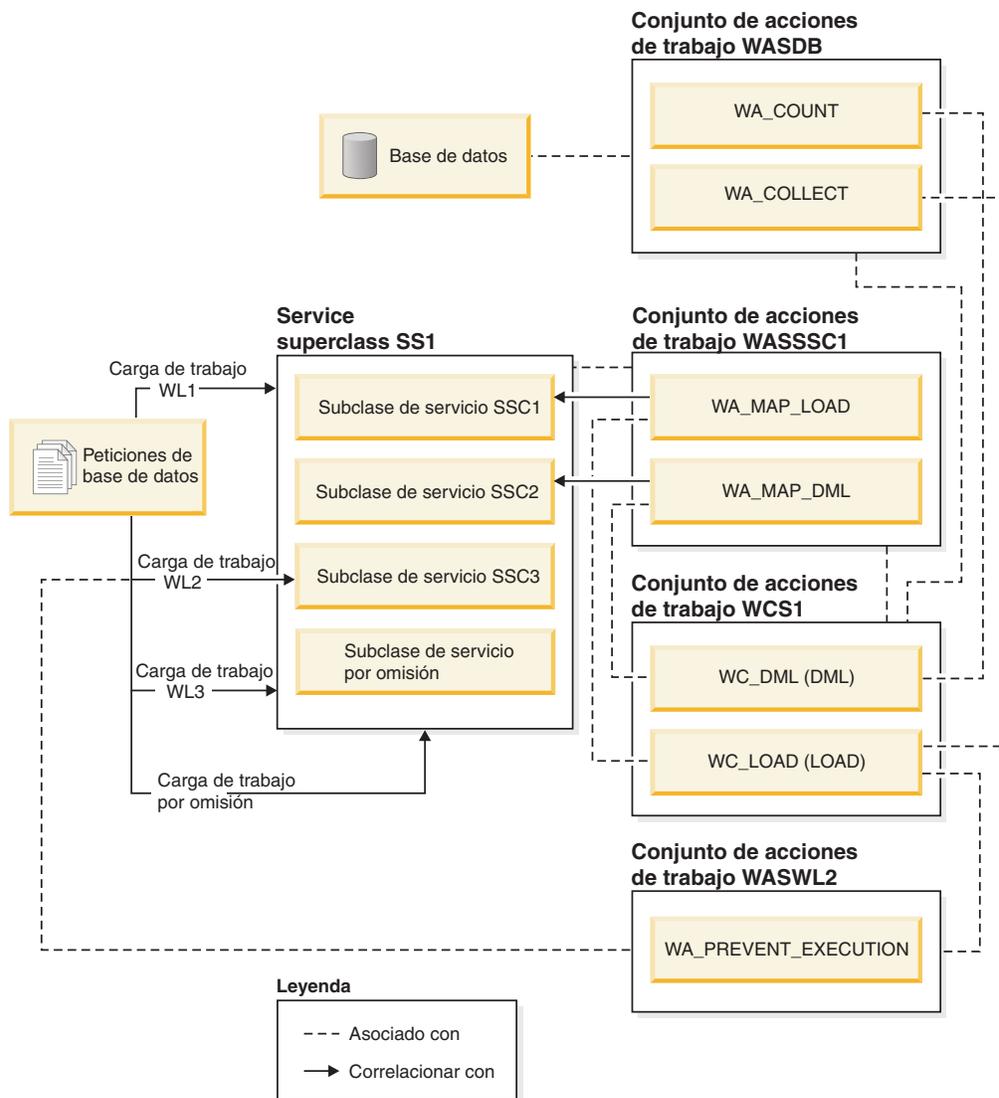


Figura 16. Visión general de conjuntos de acciones de trabajo y conjuntos de clases de trabajo

En el diagrama, algunas actividades de base de datos se correlacionan mediante la carga de trabajo WL1, la carga de trabajo WL3 y la carga de trabajo de usuario por omisión, SYSDEFAULTUSERWORKLOAD, a la superclase de servicio SS1. Puesto que el conjunto de acciones de trabajo WASDB se aplica a la base de datos, a todas las actividades que formen parte de la base de datos (independientemente de la carga de trabajo) y que se clasifiquen como clase de trabajo WC\_DML o clase de trabajo WC\_LOAD, se les aplicarán las acciones de trabajo en el conjunto de acciones de trabajo WASDB. Es decir, se contarán las actividades que tengan el tipo de trabajo DML y, para las actividades que tengan el tipo de trabajo LOAD, se recopilarán datos de actividad, que se grabarán en un supervisor de sucesos activo (si hay alguno disponible).

El conjunto de acciones de trabajo WASSSC1 se aplica a la superclase de servicio SS1. A las actividades asignadas a la carga de trabajo de usuario por omisión, la carga de trabajo WL1 o la carga de trabajo WL3 que correspondan a las clases de trabajo WC\_DML y WC\_LOAD se les aplicarán también las acciones de trabajo WA\_MAP\_DML y WA\_MAP\_LOAD, porque dichas cargas de trabajo dirigen el

trabajo a la superclase de servicio SS1. Es decir, las actividades con un tipo de trabajo LOAD se correlacionarán con la subclase de servicio SSC1 mediante la acción de trabajo WA\_MAP\_LOAD, y las actividades con un tipo de trabajo DML se correlacionarán con la subclase de servicio SSC2 mediante la acción de trabajo WA\_MAP\_DML.

Las actividades asignadas a la carga de trabajo WL2 se correlacionan directamente con una subclase de servicio (SSC3). Cuando una carga de trabajo correlaciona actividades directamente con una subclase de servicio, no se aplica a dichas actividades ninguna acción de trabajo del conjunto de acciones de trabajo WASSSC1. Sin embargo, puesto que WASWL2 se aplica a WL2, se aplicarán las acciones de trabajo del conjunto de acciones de trabajo WASWL2 a las actividades asignadas a WL2 que correspondan a WL\_LOAD. Es decir, no se permitirá ejecutar las actividades LOAD debido a la acción de trabajo PREVENT EXECUTION.

## Acciones de trabajo y conjuntos de acciones de trabajo

Una acción de trabajo, cuando se utiliza en combinación con una clase de trabajo, puede utilizarse para ayudar a controlar tipos de actividades específicos. Por ejemplo, puede aplicar diferentes acciones de trabajo a actividades LOAD de modo que se procesen de forma diferente de DML. Las acciones de trabajo se agrupan en conjuntos de acciones de trabajo.

### Acciones de trabajo

Una acción de trabajo se compone de los siguientes atributos:

- Un nombre de acción de trabajo proporcionado por el usuario, que debe ser exclusivo en el conjunto de acciones de trabajo.
- El identificador de clase de trabajo, al que debe aplicarse la acción de trabajo. Puede definir más de una acción de trabajo para una clase de trabajo, pero cada acción de trabajo debe realizar una acción diferente en esa clase de trabajo.
- La acción que debe aplicarse a la actividad de la base de datos y que coincida con la clase de trabajo. El tipo de acción válido para una acción de trabajo depende de si el conjunto de acciones de trabajo al que pertenece la acción de trabajo se aplica a una base de datos, una carga de trabajo o una superclase de servicio. Cuando el conjunto de acciones de trabajo se aplica a una base de datos, dicho conjunto de acciones de trabajo evalúa todas las actividades de trabajo que acceden a la base de datos. Cuando el conjunto de acciones de trabajo se aplica a una base de datos, dicho conjunto de acciones de trabajo evalúa todas las actividades de carga de trabajo asociadas a dicha carga de trabajo. Cuando un conjunto de acciones de trabajo se aplica a una superclase de servicio, dicho conjunto evalúa todo el trabajo que se envía para que se ejecute directamente en la superclase de servicio; es decir, el conjunto de acciones de trabajo definido en la superclase de servicio evaluará el trabajo enviado de las definiciones de carga de trabajo que designan específicamente la superclase de servicio como su clase de servicio de destino. Por ejemplo:
  - Un conjunto de acciones de trabajo que se aplica a una base de datos o a una carga de trabajo puede contener acciones de trabajo de umbral. Si una actividad se asigna a una clase de trabajo que tiene definida una acción de trabajo de umbral, el umbral se aplica a esa actividad.
  - Un conjunto de acciones de trabajo que se aplica a una superclase de servicio puede contener una acción de trabajo que correlaciona la actividad con una subclase de servicio en la superclase de servicio. Si una actividad corresponde a una clase de trabajo específica en un conjunto de clases de trabajo y el conjunto de acciones de trabajo tiene una acción de trabajo de correlación

definida para esa clase de trabajo, dicha actividad se correlacionará con la subclase de servicio que especifica la acción de trabajo.

Para ver una lista de acciones admitidas, consulte “Acciones de trabajo y el dominio de conjunto de acciones de trabajo” en la página 117.

- Un objeto, que es el destino de la acción especificada. En función de la acción, el objeto puede ser una subclase de servicio con la que se correlaciona la actividad, un umbral que especifica el umbral que debe aplicarse a la actividad, un valor nulo si la acción tiene como objeto evitar la ejecución, una de las acciones de recopilación o una actividad de recuento.
- La plantilla, que describe el histograma que recopila información estadística sobre el número de microsegundos durante los cuales deben ejecutarse las actividades asociadas a la clase de trabajo a la que se ha asignado esta acción de trabajo en un intervalo específico. Esta información sólo se recopila cuando el tipo de acción de trabajo es COLLECT AGGREGATE ACTIVITY DATA (BASE o EXTENDED). Para obtener más información sobre histogramas y plantillas de histograma, consulte “Histogramas en la gestión de carga de trabajo” en la página 278.
- Si la acción de trabajo está o no habilitada.
- Un identificador generado automáticamente que identifica la acción de trabajo.

Puede crear una acción de trabajo utilizando la palabra clave WORK ACTION en la sentencia CREATE WORK ACTION SET o la palabra clave ADD en la sentencia ALTER WORK ACTION SET. Puede modificar una acción de trabajo utilizando la palabra clave ALTER en la sentencia ALTER WORK ACTION SET. Para eliminar una acción de trabajo de un conjunto de acciones de trabajo puede utilizar la palabra clave DROP en la sentencia ALTER WORK ACTION SET o descartar todo el conjunto de acciones de trabajo.

Para visualizar sus acciones de trabajo puede consultar la vista SYSCAT.WORKACTIONS.

## Conjuntos de acciones de trabajo

Un conjunto de acciones de trabajo se compone de los siguientes atributos:

- Un nombre de conjunto de acciones de trabajo que sea exclusivo en la base de datos.
- El nombre del conjunto de clases de trabajo, que contiene una o varias clases de trabajo a las que debe aplicarse el grupo de acciones.  
Puesto que las definiciones de los conjuntos de clases de trabajo están separadas de los conjuntos de acciones de trabajo definidos para ellos, puede definir más de un conjunto de acciones de trabajo para un conjunto de clases de trabajo.
- El tipo de objeto al que está asociado el conjunto de acciones de trabajo (base de datos, superclase de servicio o carga de trabajo).
- El nombre de la superclase de servicio al que se aplica el conjunto de clases de trabajo (para conjuntos de acciones de trabajo asociados a una superclase de servicio).
- Si el conjunto de acciones de trabajo está o no habilitado.
- Comentarios del usuario.
- Una o varias acciones de trabajo (un conjunto de acciones de trabajo no tiene que contener necesariamente acciones de trabajo).
- Un ID generado automáticamente que identifica de forma exclusiva el conjunto de acciones de trabajo.

Puede crear un conjunto de acciones de trabajo utilizando la sentencia CREATE WORK ACTION SET, modificar un conjunto de acciones de trabajo utilizando la sentencia ALTER WORK ACTION SET y descartar un conjunto de acciones de trabajo utilizando la sentencia DROP WORK ACTION SET.

Para visualizar sus conjuntos de acciones de trabajo puede consultar la vista SYSCAT.WORKACTIONSETS.

Cuando crea un conjunto de acciones de trabajo, debe especificar el objeto al que debe aplicarse el conjunto de acciones de trabajo. Los tipos de objetos válidos son la base de datos, una carga de trabajo una superclase de servicio. También debe especificar el conjunto de clases de trabajo con el que debe trabajar el conjunto de acciones de trabajo. Esto le permite utilizar las clases de trabajo del conjunto de clases de trabajo para identificar los tipos de actividades a los que desea aplicar las acciones de trabajo.

Si crea un conjunto de acciones de trabajo en una superclase de servicio, hay varios aspectos importantes que conviene recordar:

1. Si configura una carga de trabajo para correlacionar sus actividades de base de datos directamente con una subclase de servicio, el conjunto de acciones de trabajo asociado a esa superclase de servicio nunca se utilizará para las actividades que emita esa carga de trabajo. En otras palabras, si una carga de trabajo correlaciona actividades directamente con una subclase de servicio, el conjunto de acciones de trabajo se pasará por alto. Ninguna de las acciones de trabajo del conjunto de acciones de trabajo se aplicará a las actividades correlacionadas directamente con la subclase de servicio.
2. Cualquier actividad, no correlacionada con una subclase de servicio por una acción de trabajo dentro del conjunto de acciones de trabajo en la superclase, se ejecuta en la SYSDEFAULTSUBCLASS de la superclase de servicio.
3. Los trabajos enviados por una carga de trabajo asociada con una superclase que no se reconoce como actividad, por ejemplo, la petición PREPARE o **RUNSTATS**, se ejecutan en la SYSDEFAULTSUBCLASS de la superclase de servicio dado que no les afecta el conjunto de acciones de trabajo.

### **Creación de una acción de trabajo**

Utilice la sentencia CREATE WORK ACTION SET o la sentencia ALTER WORK ACTION SET para crear una acción de trabajo.

### **Antes de empezar**

Para crear una acción de trabajo, necesita la autorización WLMADM o DBADM.

Si necesita conocer los requisitos adicionales, consulte los temas siguientes:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Apéndice A, “Normas de denominación generales”, en la página 545

Cuando crea una acción de trabajo:

- Asocia una acción de trabajo con una clase de trabajo. La clase de trabajo debe existir en el conjunto de clases de trabajo al que se aplica el conjunto de acciones de trabajo.
- Si la acción de trabajo es un umbral, debe definirse el conjunto de acciones de trabajo para una carga de trabajo o para la base de datos. Para conocer la lista de umbrales soportados para las acciones de trabajo, consulte “Umbrales que pueden utilizarse en acciones de trabajo” en la página 121.

- Si va a crear una acción de trabajo de correlación, debe definirse el conjunto de acciones de trabajo para una superclase de servicio. La subclase de servicio que se va a correlacionar debe existir en la superclase de servicio para la que se ha definido este conjunto de acciones de trabajo. Además, no puede especificar la subclase de servicio por omisión.
- Puede aplicarse más de una acción de trabajo a una única clase de trabajo, pero deben ser tipos de acciones de trabajo diferentes. Por ejemplo, puede aplicar una acción de trabajo de correlación y una acción de trabajo de recopilación a la misma clase de trabajo. Sin embargo, sólo puede aplicar una acción de trabajo del mismo tipo a la misma clase de trabajo. Por ejemplo, no puede aplicar más de una acción de trabajo de correlación a la misma clase de trabajo. La única excepción que existe consiste en si la acción de trabajo representa un umbral. Puede aplicar más de una acción de trabajo de umbral a una clase de trabajo única, pero cada umbral debe ser de diferente tipo.
- Si va a crear una acción de trabajo de recopilación de datos de actividad agregada, debe definirse el conjunto de acciones de trabajo para una superclase de servicio o para una carga de trabajo.

## Procedimiento

Para crear una acción de trabajo:

1. Utilice la palabra clave *definición\_acción\_trabajo* de la sentencia CREATE WORK ACTION SET, o la palabra clave ADD *definición\_acción\_trabajo* de la sentencia ALTER WORK ACTION SET. Especifique una o más de las opciones siguientes para la acción de trabajo:
  - Un nombre para la acción de trabajo. El nombre de la acción de trabajo debe ser exclusivo en el conjunto de acciones de trabajo.
  - El nombre de la clase de trabajo a la que se aplica esta acción de trabajo. La clase de trabajo debe ser una de las clases de trabajo del conjunto de clases de trabajo con la que está asociado el conjunto de acciones de trabajo. Por ejemplo, para aplicar esta acción de trabajo a la clase de trabajo LARGEDML, debe especificar:
 

```
ON WORK CLASS LARGEDML
```
  - La acción que se va a aplicar a las actividades que coinciden con la clase de trabajo para esta acción de trabajo:
    - Si el conjunto de acciones de trabajo está asociado con una superclase de servicio, puede especificar la palabra clave MAP ACTIVITY de manera que la acción de trabajo correlacione actividades con una subclase de servicio en la superclase de servicio. Por omisión, el hecho de correlacionar acciones de trabajo hace que las actividades que están anidadas se correlacionen con la misma subclase de servicio que su elemento padre. Un cursor que ha estado abierto en una rutina constituye un ejemplo de una actividad anidada.
 

Por ejemplo, si desea que la acción de trabajo se correlacione con la subclase de servicio SMALLREAD, y desea que todas las actividades anidadas se correlacionen con la misma subclase de servicio, deberá especificar:

```
MAP ACTIVITY TO SMALLREAD
```

También puede especificar:

```
MAP ACTIVITY WITH NESTED TO SMALLREAD
```

Si desea que la acción de trabajo se correlacione con la subclase de servicio y no con las actividades anidadas a esta subclase de servicio, debe especificar:

```
MAP ACTIVITY WITHOUT NESTED TO SMALLREAD
```

Si define la acción de trabajo como WITHOUT NESTED, las actividades anidadas se gestionan de acuerdo con su tipo de actividad en lugar de correlacionarse automáticamente con la misma subclase de servicio que la actividad padre. Por ejemplo, si una actividad CALL se correlaciona con la subclase de servicio subsc1, y la rutina contiene un cursor abierto, éste puede correlacionarse con una subclase de servicio diferente si entra en otra clase de trabajo que tenga otra acción de trabajo de correlación aplicada.

- Si el conjunto de acciones de trabajo está asociado con una base de datos o una carga de trabajo, puede especificar una palabra clave WHEN para indicar a un umbral que se aplique a la actividad, así como la acción que debe llevarse a cabo si la actividad hace que se infrinja el umbral. Puede especificar los umbrales siguientes para una acción de trabajo:
  - ACTIVITYTOTALTIME
  - ESTIMATEDSQLCOST
  - CONCURRENTDBCOORDACTIVITIES
  - CPUTIME
  - SQLROWSREAD
  - SQLTEMPSPACE
  - SQLROWSRETURNED

Si se infringe el umbral, puede especificar que se lleven a cabo las acciones siguientes:

- Si se van a recopilar datos de actividad acerca de la actividad que originó que se infringiera el umbral. Si se recopilan, cuando la actividad termina de ejecutarse, los datos de la actividad se graban en un supervisor activo de sucesos de actividades. Por omisión, no se recopila ningún dato sobre la actividad. Si desea recopilar datos sobre esta actividad, puede hacerlo desde el miembro coordinador, desde un miembro de base de datos específico, o desde todos los miembros de base de datos. Tiene la opción de recopilar estos datos con o sin detalles sobre la sentencia y su entorno de compilación. Si desea recopilar detalles sobre la sentencia y el entorno de compilación, también puede especificar los valores de los datos de entrada de la actividad.
- Si la actividad que causó la infracción del umbral podrá seguir ejecutándose o no. Por omisión, la actividad se detendrá.

Por ejemplo, si desea que la acción de trabajo compruebe qué sentencias DML tienen un coste superior a los 2.000 timerons, recopilar los datos básicos sobre esta actividad cuando se infrinja el umbral y continuar con la ejecución, debería especificar:

```
WHEN ESTIMATEDSQLCOST > 2000 COLLECT ACTIVITY DATA CONTINUE
```

- Para impedir la ejecución de las actividades que se corresponden con la clase de trabajo definida para esta acción de trabajo, puede utilizar la palabra clave PREVENT EXECUTION.
- Para contar el número de actividades de base de datos asociadas con la clase de trabajo sin aumentar la actividad general de otra acción (como recopilar datos o correlacionar una actividad), puede especificar la palabra clave COUNT ACTIVITY.
- Para recopilar los datos de actividad para las actividades que están bajo la clase de trabajo, especifique la palabra clave COLLECT ACTIVITY DATA. Si se recopilan, cuando la actividad termina de ejecutarse, los datos de la actividad se graban en un supervisor activo de sucesos de actividades. Por omisión, no se recopila ningún dato sobre la actividad. Si desea recopilar

datos sobre esta actividad, puede hacerlo desde el miembro coordinador o desde todos los miembros de base de datos. Si desea recopilar detalles de actividad, como información sobre el entorno de compilación y sobre sentencias, hágalo especificando la palabra WITH DETAILS. También puede utilizar la palabra clave AND VALUES para que los valores de datos de entrada (para las actividades que los contienen) se envíen al supervisor de sucesos de actividades.

Por ejemplo, supongamos que tiene un conjunto de acciones de trabajo que se ha aplicado a una superclase de servicio. Desea que los datos de actividad correspondientes a todas las actividades que están asignadas a esta acción de trabajo se graben en el supervisor de sucesos pertinente, incluida toda la información sobre actividades agregadas, información sobre el entorno de compilación y cualquier valor de datos de entrada. Debe especificar:

```
COLLECT ACTIVITY DATA ON ALL WITH DETAILS AND VALUES
```

- Si el conjunto de acciones de trabajo está asociado con una superclase de servicio o una carga de trabajo, puede especificar la palabra clave COLLECT AGGREGATE ACTIVITY DATA para recopilar los datos de actividades agregadas correspondientes a las actividades que están bajo la clase de trabajo. Si se recopilan, se capturan los datos de actividades agregadas y se envían al supervisor de sucesos pertinente. Esta información se recopila periódicamente a un intervalo especificado mediante el parámetro de configuración `wlm_collect_int` de la base de datos.

Por ejemplo, supongamos que tiene un conjunto de acciones de trabajo que se ha aplicado a una superclase de servicio. Desea que los datos de actividades agregadas correspondientes a todas las actividades que están asignadas a esta acción de trabajo se graben en el supervisor de sucesos pertinente, incluidos los datos básicos, el histograma del coste estimado de DML (lenguaje de manipulación de datos) de la actividad, y el histograma de tiempo entre llegadas de DML de la actividad. Debe especificar

```
COLLECT AGGREGATE ACTIVITY DATA EXTENDED
```

- Las plantillas de histograma que utiliza la acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para describir los histogramas creados para la clase de trabajo correspondiente. La especificación de las plantillas de histograma que utiliza una acción de trabajo añade las filas correspondientes en la vista SYSCAT.HISTOGRAMTEMPLATEUSE, que muestra las plantillas de histograma a las que la clase de servicio o la acción de trabajo hace referencia. Por ejemplo, si desea recopilar estadísticas de tiempo entre llegadas para la plantilla de histograma de tiempo entre llegadas por omisión, debe especificar:

```
INTERARRIVALTIME HISTOGRAM TEMPLATE SYSDEFAULTHISTOGRAM
```

Para obtener más información acerca de histogramas y plantillas de histograma, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.

- Si la acción de trabajo está habilitada o inhabilitada. Por omisión las acciones de trabajo se crean como habilitadas, pero puede decidir habilitarlas o inhabilitarlas mediante la palabra clave ENABLE o DISABLE. Si la acción de trabajo está inhabilitada, el servidor de datos no tendrá en consideración dicha acción cuando las actividades entren en la base de datos o en la superclase de servicio (según el objeto para el que haya creado el conjunto de acciones de trabajo).
2. Confirme los cambios. Cuando confirma los cambios, la acción de trabajo se añade a la vista SYSCAT.WORKACTIONS. Si la acción de trabajo es un umbral, éste se añade a la vista SYSCAT.THRESHOLDS.

Una acción de trabajo nueva solo surte efecto en la base de datos una vez que se ha confirmado, y no afecta a ninguna actividad de la base de datos que se esté ejecutando en ese momento.

## Alteración de una acción de trabajo

Si necesita alterar una acción de trabajo, utilice la sentencia ALTER WORK ACTION SET.

### Antes de empezar

Para alterar una acción de trabajo, necesita la autorización SQLADM, WLMADM o DBADM. Para especificar una cláusula que no sea COLLECT, el ID de autorización debe incluir la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para conocer los requisitos adicionales.

### Procedimiento

Para alterar una acción de trabajo:

1. Utilice la palabra clave ALTER de la sentencia ALTER WORK ACTION SET para cambiar una o más de las características siguientes de la acción de trabajo.
  - Puede alterar la clase de trabajo a la que se aplica la acción de trabajo. La clase de trabajo ya debe existir en el conjunto de clases de trabajo al que se aplica el conjunto de acciones de trabajo.
  - Si la acción de trabajo se correlaciona con una subclase de servicio, puede alterar la subclase de servicio con la que se va a correlacionar la actividad de la base de datos. Solo puede cambiar la correlación con una subclase de servicio en la misma superclase de servicio. No puede realizar la correlación con la subclase de servicio por omisión. También puede cambiar si las actividades anidadas de la actividad se van a correlacionar con la misma subclase de servicio o no. Por ejemplo, si la acción de trabajo está actualmente definida como WITH NESTED, puede cambiarlo a WITHOUT NESTED. Este cambio hará que las actividades anidadas se gestionen según su tipo de actividad en lugar de que se correlacionen automáticamente con la misma subclase de servicio que la actividad padre correspondiente. Por ejemplo, si una sentencia CALL se correlaciona con la subclase de servicio SUBSC1, y la rutina tiene un cursor abierto, éste puede correlacionarse con una subclase de servicio diferente si entra en otra clase de trabajo que tenga otra acción de trabajo de correlación aplicada.
  - Puede alterar el tipo de acción especificado para la acción de trabajo (es decir, correlación, umbral, impedimento de ejecución, actividad de recuento, acciones de recopilación), pero deberá hacerlo a un tipo de trabajo válido. Por ejemplo, si la acción de trabajo es correlacionar la actividad con una subclase de servicio, no podrá cambiar la acción de trabajo a un umbral, o viceversa. En este ejemplo, la razón estriba en que el conjunto de acciones de trabajo debe haberse aplicado a una superclase de servicio para disponer de una acción de correlación, y las acciones de umbral no son válidas para los conjuntos de acciones de trabajo aplicados a superclases de servicio. Si altera el tipo de acción de trabajo que es una acción de trabajo de umbral, o altera el tipo de acción de trabajo a un umbral, ocurrirá lo siguiente:
    - Si la acción de trabajo era un umbral y se ha cambiado a una acción que no es de umbral, el umbral se elimina de la vista SYSCAT.THRESHOLDS.
    - Si la acción de trabajo no era un umbral y se ha cambiado por un umbral, el nuevo umbral se creará en la vista SYSCAT.THRESHOLDS.

**Nota:** Si la acción es un umbral, no puede alterar el tipo de umbral por un umbral diferente. Así por ejemplo, si la acción de trabajo era un umbral SQLROWSRETURNED, no puede cambiarlo por un umbral SQLTEMPSPACE. Además, no puede cambiar el tipo de acción de trabajo de un umbral de acción de trabajo CONCURRENTDBCOORDACTIVITIES habilitado.

- Puede alterar las plantillas de histograma que utiliza la acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para describir los histogramas creados para la clase de trabajo correspondiente. La actualización de las plantillas de histograma que utiliza una acción de trabajo actualiza las filas correspondientes en la vista SYSCAT.HISTOGRAMTEMPLATEUSE, que muestra las plantillas de histograma a las que la clase de servicio o la acción de trabajo hace referencia. Para obtener más información acerca de histogramas y plantillas de histograma, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.
  - Puede alterar si desea habilitar o inhabilitar la acción de trabajo. Por omisión, las acciones de trabajo están habilitadas. Cuando están habilitadas, el servidor de datos toma en consideración la acción de trabajo de la aplicación frente a la actividad que está bajo la clase de trabajo correspondiente a la acción de trabajo. Si la acción de trabajo está inhabilitada, el servidor de datos la pasa por alto.
2. Confirme los cambios. Cuando confirma los cambios, la acción de trabajo se actualiza en la vista SYSCAT.WORKACTIONS.

## Inhabilitación de una acción de trabajo

Puede inhabilitar una acción de trabajo que no desee aplicar a una clase de trabajo. Durante la ejecución, a la acción de trabajo inhabilitada se le trata como si no existiera.

### Antes de empezar

Para inhabilitar una acción de trabajo, necesita la autorización WLMADM o DBADM.

### Procedimiento

Para inhabilitar una acción de trabajo:

1. Utilice una de las sentencias siguientes, en función de si va a crear o alterar un conjunto de acciones de trabajo:
  - Utilice las palabras clave DISABLE y ADD de la sentencia CREATE WORK ACTION SET. Por ejemplo:

```
ADD WORK ACTION
nombre_acción_trabajo ON WORK CLASS
nombre_clase_trabajo ... DISABLE
```
  - Utilice las palabras clave DISABLE y ALTER de la sentencia ALTER WORK ACTION SET. Por ejemplo:

```
ALTER WORK ACTION
nombre_acción_trabajo ... DISABLE
```
2. Confirme los cambios. Cuando confirma los cambios, la acción de trabajo se actualiza en la vista SYSCAT.WORKACTIONS.

### Descarte de una acción de trabajo

Si ya no necesita una acción de trabajo, puede descartarla del conjunto de acciones de trabajo.

## Antes de empezar

- Para descartar una acción de trabajo, necesita la autorización WLMADM o DBADM.
- Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para conocer los requisitos adicionales.

## Procedimiento

Para descartar una acción de trabajo:

1. Utilice la palabra clave DROP de la sentencia ALTER WORK ACTION SET. Si desea descartar una acción de trabajo de umbral CONCURRENTDBCOORDACTIVITIES, debe inhabilitar la acción de trabajo en una operación ALTER WORK ACTION SET, confirmar el cambio, verificar que no quedan actividades en cola, y luego descartar el umbral en una segunda operación ALTER WORK ACTION SET.
2. Confirme los cambios. Cuando confirma los cambios, la acción de trabajo se elimina de la vista SYSCAT.WORKACTIONS. Si la acción de trabajo es una acción de trabajo de umbral, éste también se eliminará de la vista SYSCAT.THRESHOLDS.

Un conjunto de acciones de trabajo y una acción de trabajo alterados solo surten efecto en la base de datos después de confirmarse, y no afectan a ninguna actividad de la base de datos que se esté ejecutando en ese momento.

## Creación de un conjunto de acciones de trabajo

Para crear una acción de trabajo y un conjunto de acciones de trabajo, utilice la sentencia CREATE WORK ACTION SET.

## Antes de empezar

Para crear un conjunto de acciones de trabajo, necesita la autorización WLMADM o DBADM.

Si necesita conocer los requisitos adicionales, consulte los temas siguientes:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18

Cuando se crea un conjunto de acciones de trabajo:

- Debe asociarlo con un conjunto de clases de trabajo. El conjunto de clases de trabajo ya debe estar creado.
- También debe asociarlo con una base de datos, una carga de trabajo o una superclase de servicio. Si va a asociar el conjunto de acciones de trabajo con una superclase de servicio, la clase de servicio ya debe estar creada. No puede definir el conjunto de acciones de trabajo para la clase de servicio del sistema por omisión (SYSDEFAULTSYSTEMCLASS), la clase de usuario por omisión (SYSDEFAULTUSERCLASS) o la clase de servicio de mantenimiento por omisión (SYSDEFAULTMAINTENANCECLASS). Si va a asociar el conjunto de acciones de trabajo con una carga de trabajo, ésta ya debe estar creada en el servidor actual. No puede asociar un conjunto de acciones de trabajo con la carga de trabajo de administración por omisión (SYSDEFAULTADMWORKLOAD).

## Procedimiento

Para crear un conjunto de acciones de trabajo:

1. Utilice la sentencia CREATE WORK ACTION SET con las opciones siguientes:

- Especifique un nombre para el conjunto de acciones de trabajo. El nombre del conjunto de acciones de trabajo debe ser exclusivo en la base de datos.
- Especifique el objeto con el que el conjunto de acciones de trabajo está asociado. Puede especificar una base de datos, una carga de trabajo o una superclase de servicio. Si especifica que el conjunto de acciones de trabajo está asociado con una base de datos, ninguna de las acciones de trabajo del conjunto de acciones de trabajo podrá correlacionar acciones de trabajo ni recopilar acciones agregadas. Si especifica que el conjunto de acciones de trabajo está asociado con una superclase de servicio, ninguna de las acciones de trabajo del conjunto de acciones de trabajo podrá ser un umbral. Si especifica que el conjunto de acciones de trabajo está asociado con una carga de trabajo, ninguna de las acciones de trabajo del conjunto de acciones de trabajo podrá correlacionar acciones de trabajo. Por ejemplo, para aplicar el conjunto de acciones de trabajo a la superclase de servicio REPORTS, deberá especificar:

```
FOR SERVICE CLASS REPORTS
```

Para aplicar el conjunto de acciones de trabajo a la base de datos, debe especificar:

```
FOR DATABASE
```

Para aplicar el conjunto de acciones de trabajo a una carga de trabajo llamada WL1, debe especificar:

```
FOR WORKLOAD WL1
```

- Especifique el conjunto de clases de trabajo con el que el conjunto de acciones de trabajo está asociado. Las clases de trabajo del conjunto de clases de trabajo clasifican las actividades de la base de datos a las que se aplicarán las acciones de trabajo del conjunto de acciones de trabajo. Por ejemplo, para asociar el conjunto de acciones de trabajo con el conjunto de clases de trabajo LARGEREADS, deberá especificar:

```
USING WORK CLASS SET LARGEREADS
```

- Opcional: cree una o más acciones de trabajo para el conjunto de acciones de trabajo. Para obtener más información, consulte “Creación de una acción de trabajo” en la página 107.
- Opcional: especifique si el conjunto de acciones de trabajo está habilitado o inhabilitado. Por omisión, el conjunto de acciones de trabajo está habilitado. Si el conjunto de acciones de trabajo está inhabilitado, el servidor de datos no tendrá en consideración dicho conjunto, ni ninguna acción de trabajo que contenga, cuando se ejecuten las actividades.

2. Confirme los cambios. Cuando confirma los cambios, el conjunto de acciones de trabajo se añade a la vista SYSCAT.WORKACTIONSETS.

Un conjunto de acciones de trabajo nuevo solo surte efecto en la base de datos una vez que se ha confirmado, y no afecta a ninguna actividad de la base de datos que se esté ejecutando en ese momento.

## Alteración de un conjunto de acciones de trabajo

Para añadir, alterar o descartar una acción de trabajo de un conjunto de acciones de trabajo, o para habilitar o inhabilitar el conjunto de acciones de trabajo, utilice la sentencia ALTER WORK ACTION SET.

### Antes de empezar

Para alterar un conjunto de acciones de trabajo, necesita la autorización SQLADM, WLMADM o DBADM. Para especificar una cláusula que no sea COLLECT, el id de autorización debe incluir la autorización WLMADM o DBADM.

Si necesita conocer los requisitos adicionales, consulte los temas siguientes:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Apéndice A, “Normas de denominación generales”, en la página 545

Cuando crea un conjunto de acciones de trabajo para que opere con un conjunto de clases de trabajo específico, no puede cambiarlo para que opere con un conjunto de clases de trabajo diferente porque las acciones de trabajo del conjunto de acciones de trabajo dependen de las clases de trabajo del conjunto de clases de trabajo. Si desea cambiar el conjunto de clases de trabajo al que se aplica este conjunto de acciones de trabajo, deberá descartar y luego volver a crear el conjunto de acciones de trabajo.

No puede cambiar el objeto al que se aplica el conjunto de acciones de trabajo porque el tipo de acciones de trabajo del conjunto de acciones de trabajo depende del objeto (base de datos, carga de trabajo o superclase de servicio) para el que está definido el conjunto de acciones de trabajo. Si desea cambiar el objeto con el que está asociado el conjunto de acciones de trabajo, debe descartar dicho conjunto y luego volver a crearlo.

## Procedimiento

Para alterar un conjunto de acciones de trabajo:

1. Si desea añadir una acción de trabajo nueva al conjunto de acciones de trabajo, utilice la palabra clave ADD. Para obtener información sobre los parámetros que puede especificar a la hora de añadir una acción de trabajo a un conjunto de acciones de trabajo, consulte “Creación de una acción de trabajo” en la página 107
2. Si desea alterar una acción de trabajo existente, utilice la palabra clave ALTER. Para obtener información sobre cómo alterar una acción de trabajo, consulte “Alteración de una acción de trabajo” en la página 111.
3. Si desea descartar una acción de trabajo, utilice la palabra clave DROP. Para obtener información sobre cómo descartar una acción de trabajo de un conjunto de acciones de trabajo, consulte “Descarte de una acción de trabajo” en la página 112.
4. Puede habilitar un conjunto de acciones de trabajo que esté actualmente inhabilitado, y viceversa. Si inhabilita un conjunto de acciones de trabajo habilitado, el servidor de datos lo pasará por alto después de confirmar los cambios. Para obtener más información, consulte “Inhabilitación de un conjunto de acciones de trabajo” en la página 116. Si habilita el conjunto de acciones de trabajo, después de confirmar los cambios, dicho conjunto se aplicará a la siguiente actividad aplicable que entre en la base de datos.

**Nota:** El hecho de inhabilitar un conjunto de acciones de trabajo no inhabilita las acciones de trabajo de dicho conjunto, sino que el conjunto ya no afectará a ningún trabajo. Si desea descartar un conjunto de acciones de trabajo que contiene un umbral de acción de trabajo de simultaneidad, primero debe inhabilitar la acción de trabajo de simultaneidad, porque para descartar los umbrales de simultaneidad primero deben estar inhabilitados.

5. Confirme los cambios. Cuando confirma los cambios, el conjunto de acciones de trabajo se actualiza en la vista SYSCAT.WORKACTIONSETS. La vista SYSCAT.WORKACTIONS se actualiza cada vez que se añade, se altera o se descarta alguna acción de trabajo.

## Inhabilitación de un conjunto de acciones de trabajo

Para inhabilitar un conjunto de acciones de trabajo, utilice la palabra clave `DISABLE` de la sentencia `CREATE WORK ACTION SET` o de la sentencia `ALTER WORK ACTION SET`.

### Antes de empezar

Para inhabilitar un conjunto de acciones de trabajo, necesita la autorización `WLMADM` o `DBADM`.

### Acerca de esta tarea

El hecho de inhabilitar un conjunto de acciones de trabajo no inhabilita las acciones de trabajo de dicho conjunto, sino que el conjunto ya no afectará a ningún trabajo. Durante la ejecución, a un conjunto de acciones de trabajo inhabilitado se le trata como si no existiera.

**Nota:** Si desea descartar un conjunto de acciones de trabajo que contiene un umbral de acción de trabajo de simultaneidad, primero debe inhabilitar la acción de trabajo de simultaneidad, porque para descartar los umbrales de simultaneidad primero deben estar inhabilitados.

Por ejemplo, supongamos que tiene un conjunto de acciones de trabajo llamado `READACTIVITIES` que está asociado con un conjunto de clases de trabajo llamado `READCLASSES`, y que el conjunto de acciones de trabajo está definido para una superclase de servicio llamada `READSERVICECLASS`. El conjunto de acciones de trabajo `SMALLREAD` tiene una acción de trabajo que correlaciona todas las sentencias `SELECT` con la subclase de servicio `SMALLREADSERVICECLASS`. Si el conjunto de acciones de trabajo `READACTIVITIES` está inhabilitado, todas las sentencias `SELECT` se considerarán como si el conjunto de acciones de trabajo `READACTIVITIES` no existiera, y se correlacionarán con la subclase de servicio por omisión.

### Procedimiento

Para inhabilitar un conjunto de acciones de trabajo:

1. Utilice una de las sentencias siguientes, en función de si va a crear o alterar un conjunto de acciones de trabajo:
  - Para crear un conjunto de acciones de trabajo que está inhabilitado:

```
CREATE WORK ACTION SET
nombre_conjunto_acciones_trabajo ...
DISABLE
```
  - Para inhabilitar un conjunto de acciones de trabajo que ya existe:

```
ALTER WORK ACTION SET
nombre_conjunto_acciones_trabajo ... DISABLE
```
2. Confirme los cambios. Cuando confirma los cambios, el conjunto de acciones de trabajo se actualiza en la vista `SYSCAT.WORKACTIONSETS`.

### Descarte de un conjunto de acciones de trabajo

Utilice la sentencia `DROP WORK ACTION SET` para descartar un conjunto de acciones de trabajo.

### Antes de empezar

Para descartar un conjunto de acciones de trabajo, necesita la autorización `WLMADM` o `DBADM`.

## Acerca de esta tarea

Al descartar un conjunto de acciones de trabajo también se descartan las acciones de trabajo que contiene.

Si el conjunto de acciones de trabajo contiene una acción de trabajo de umbral `CONCURRENTDBCOORDACTIVITIES`, deberá inhabilitar esa acción de trabajo antes de descartarla.

## Procedimiento

Para descartar un conjunto de acciones de trabajo:

1. Utilice la sentencia `DROP WORK ACTION SET`.
2. Confirme los cambios. Cuando confirma los cambios, el conjunto de acciones de trabajo se elimina de la vista `SYSCAT.WORKACTIONSETS`. Además, todas las acciones de trabajo que formaban parte del conjunto de acciones de trabajo también se eliminan de la vista `SYSCAT.WORKACTIONS`. Si el conjunto de acciones de trabajo contiene acciones de trabajo de umbral, los umbrales también se eliminan de la vista `SYSCAT.THRESHOLDS`.

## Acciones de trabajo y el dominio de conjunto de acciones de trabajo

Puede definir un conjunto de acciones de trabajo para una base de datos, una superclase de servicio o una carga de trabajo. El tipo de acciones de trabajo que puede definirse para un conjunto de acciones de trabajo dependerá del tipo de objeto para el que está definido el conjunto de acciones de trabajo.

Si se define el conjunto de acciones de trabajo para una base de datos, las acciones de trabajo del conjunto de acciones de trabajo deben ser alguna de las acciones siguientes:

- Un umbral

Los umbrales siguientes se aplican a cada actividad de la clase de trabajo coincidente:

- `ACTIVITYTOTALTIME`
- `CPUTIME`
- `ESTIMATEDSQLCOST`
- `SQLROWSREAD`
- `SQLROWSRETURNED`
- `SQLTEMPSPACE`

El umbral siguiente se aplica a todas las actividades de la clase de trabajo coincidente como grupo:

- `CONCURRENTDBCOORDACTIVITIES`

El umbral real se especifica mediante la cláusula `WHEN tipo-umbral`. Varias acciones de trabajo de umbral pueden aplicarse a una misma clase de trabajo si todos los umbrales son de tipos diferentes. Si se especifica esta acción, el umbral se aplica a todas las actividades de base de datos asociadas a la clase de trabajo.

- `PREVENT EXECUTION`

Si se especifica esta opción, no se permitirá la ejecución de ninguna actividad que coincida con la clase de trabajo asociada.

- `COLLECT ACTIVITY DATA`

Si se especifica esta acción, la información sobre las actividades de la base de datos correspondientes a la clase de trabajo para la que se ha definido esta acción de trabajo se grabará en el supervisor de sucesos ACTIVITIES activo cuando las actividades finalicen su ejecución. Para obtener más información, consulte el apartado “Recopilación de datos para actividades individuales”.

- COUNT ACTIVITY

Si se especifica esta acción, toda la actividad de la base de datos correlacionada con la clase de trabajo asociada hará que se incremente el contador para ese tipo de clase de trabajo. (El contador de la clase de trabajo se incrementa en uno cada vez que se asocia una actividad con esa clase de trabajo). La acción de trabajo COUNT ACTIVITY proporciona una manera eficiente de garantizar que se actualiza este contador. Si no se aplica ninguna acción de trabajo a una actividad correspondiente a una clase de trabajo, el contador de actividades de la clase de trabajo no se incrementará. A veces, la única acción que le interesará será obtener un recuento de actividades de un determinado tipo. Para obtener más información, consulte el apartado “Recopilación de datos para actividades individuales”.

Si las acciones de trabajo del conjunto de acciones de trabajo definidas para una base de datos no son ninguna de estas acciones, se devolverá un error SQL4720N.

Si define un conjunto de acciones de trabajo para una superclase de servicio, las acciones de trabajo del conjunto de acciones de trabajo deben ser cualquiera de las acciones siguientes:

- Una acción de correlación

Puede correlacionar una actividad con cualquier subclase de servicio en la superclase de servicio, salvo para la subclase de servicio por omisión. Especifique la subclase de servicio con la que se correlacionará la actividad utilizando la palabra clave MAP ACTIVITY TO SERVICE CLASS. Sólo puede aplicarse una acción de trabajo de correlación del conjunto de acciones de trabajo a la misma clase de trabajo.

- PREVENT EXECUTION

El comportamiento es el mismo que la acción de trabajo de base de datos.

- COLLECT ACTIVITY DATA

El comportamiento es el mismo que la acción de trabajo de base de datos.

- COLLECT AGGREGATE ACTIVITY DATA

Si se especifica esta opción, se recopilarán los datos de actividad de base de datos acumulados que correspondan a la clase de trabajo para la que está definida esta acción de trabajo.

- COUNT ACTIVITY

El comportamiento es el mismo que la acción de trabajo de base de datos.

Si las acciones de trabajo del conjunto de acciones de trabajo definidas para una superclase de servicio no son ninguna de estas acciones, se devolverá un error SQL4720N.

Si se define el conjunto de acciones de trabajo para una carga de trabajo, las acciones de trabajo del conjunto de acciones de trabajo deben ser alguna de las acciones siguientes:

- Un umbral

Los umbrales siguientes se aplican a cada actividad de la clase de trabajo coincidente:

- ACTIVITYTOTALTIME

- CPUTIME
- ESTIMATEDSQLCOST
- SQLROWSREAD
- SQLROWSRETURNED
- SQLTEMPSPACE

El umbral siguiente se aplica a todas las actividades de la clase de trabajo coincidente como grupo:

- CONCURRENTDBCOORDACTIVITIES

El umbral real se especifica mediante la cláusula *WHEN tipo-umbral*. Varias acciones de trabajo de umbral pueden aplicarse a una misma clase de trabajo si todos los umbrales son de tipos diferentes. Si se especifica esta acción, el umbral se aplica a todas las actividades de base de datos asociadas a la clase de trabajo.

- PREVENT EXECUTION

El comportamiento es el mismo que la acción de trabajo de base de datos.

- COLLECT ACTIVITY DATA

El comportamiento es el mismo que la acción de trabajo de base de datos.

- COLLECT AGGREGATE ACTIVITY DATA

El comportamiento es el mismo que la acción de trabajo de superclase de servicio.

- COUNT ACTIVITY

El comportamiento es el mismo que la acción de trabajo de base de datos.

Si las acciones de trabajo del conjunto de acciones de trabajo definidas para una carga de trabajo no son ninguna de estas acciones, se devolverá un error SQL4720N.

### **Ejemplo: Aplicación de actividades de clase de trabajo**

En la figura siguiente se muestra un ejemplo de cómo deben aplicarse las clases de trabajo en un conjunto de clases de trabajo denominado *LARGE ACTIVITIES* tanto a la base de datos como a una superclase de servicio. Para cumplir este objetivo, se crean dos conjuntos de acciones de trabajo, "Actividades grandes de base de datos" y "Actividades grandes de clase de servicio".

Aunque este ejemplo no lo muestra, también puede aplicar las clases del conjunto de clases de trabajo *LARGE ACTIVITIES* a una carga de trabajo; para ello, cree un conjunto de acciones de trabajo asociado a la carga de trabajo y, a continuación, asocie el conjunto de acciones de trabajo al conjunto de clases de trabajo *LARGE ACTIVITIES*.

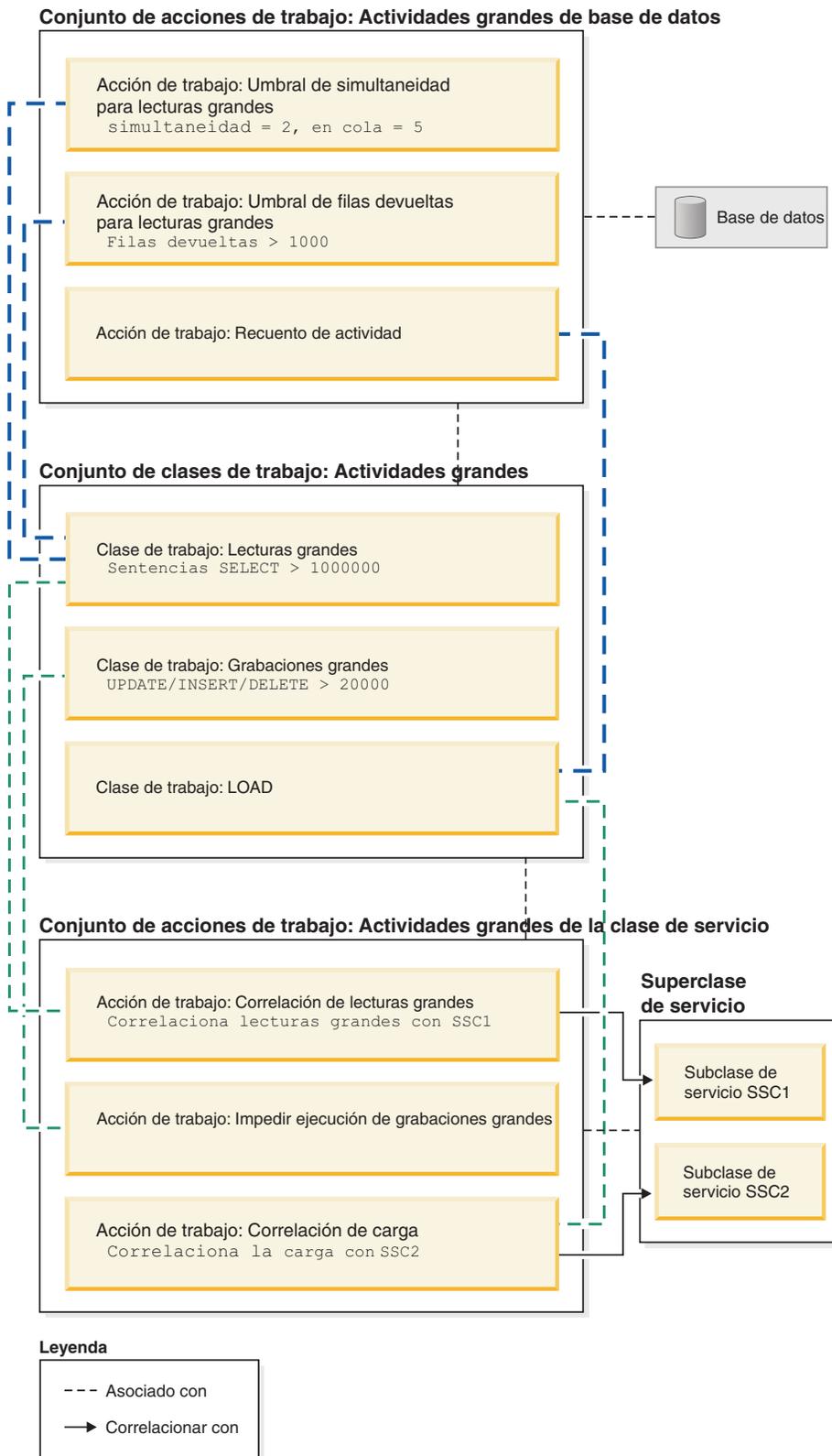


Figura 17. Ejemplo de acciones de trabajo, conjuntos de acciones de trabajo, clases de trabajo y conjunto de clases de trabajo

Los conjuntos de acciones de trabajo son los siguientes:

- Actividades grandes de base de datos contiene:

- Umbral de simultaneidad para lecturas grandes, que permite que se ejecuten simultáneamente dos lecturas grandes y colocar en cola cinco lecturas grandes.
- Umbral de filas devueltas para lecturas grandes, que impide que las lecturas grandes devuelvan más de 1000 filas.
- Recuento de actividad para la carga, que cuenta el número de veces que se ejecuta el programa de utilidad de carga en la base de datos.
- Actividades grandes de la clase de servicio contiene:
  - Correlación de lecturas grandes, que correlaciona lecturas grandes con la subclase de servicio 1
  - Correlación de grabaciones grandes, que impide la ejecución de grabaciones grandes.
  - Correlación de carga, que correlaciona las cargas con la subclase de servicio 2

Un conjunto de acciones de trabajo no tiene que contener necesariamente una acción por cada clase de trabajo del conjunto de clases de trabajo al que se aplica el conjunto de acciones de trabajo. Además, se puede aplicar a una clase de trabajo más de una acción de trabajo siempre que los tipos de acción sean diferentes. Una clase de trabajo puede tener aplicado más de un umbral siempre que los tipos de umbral sean diferentes.

## Umbrales que pueden utilizarse en acciones de trabajo

Los conjuntos de acciones de trabajo que puede definir para bases de datos pueden contener acciones de trabajo que especifican umbrales.

Se admiten los siguientes umbrales:

- Umbral agregado:
  - CONCURRENTDBCOORDACTIVITIES
  - CONCURRENTWORKLOADACTIVITIES
- Umbrales de actividad:
  - SQLTEMPSPACE
  - SQLROWSRETURNED
  - ACTIVITYTOTALTIME
  - ESTIMATEDSQLCOST
  - CPUTIME
  - SQLROWSREAD

## Aplicación de acciones de trabajo a actividades de la base de datos

Sólo puede aplicarse un conjunto de acciones de trabajo a una base de datos, una superclase de servicio o una carga de trabajo.

Cuando se envía un trabajo al servidor de datos, se asocia a una carga de trabajo, ya sea una carga de trabajo definida por el usuario, ya sea la carga de trabajo por omisión, y, a continuación se correlaciona con una clase de servicio.

En la figura siguiente se muestra el proceso de aplicación de una acción de trabajo a una actividad.

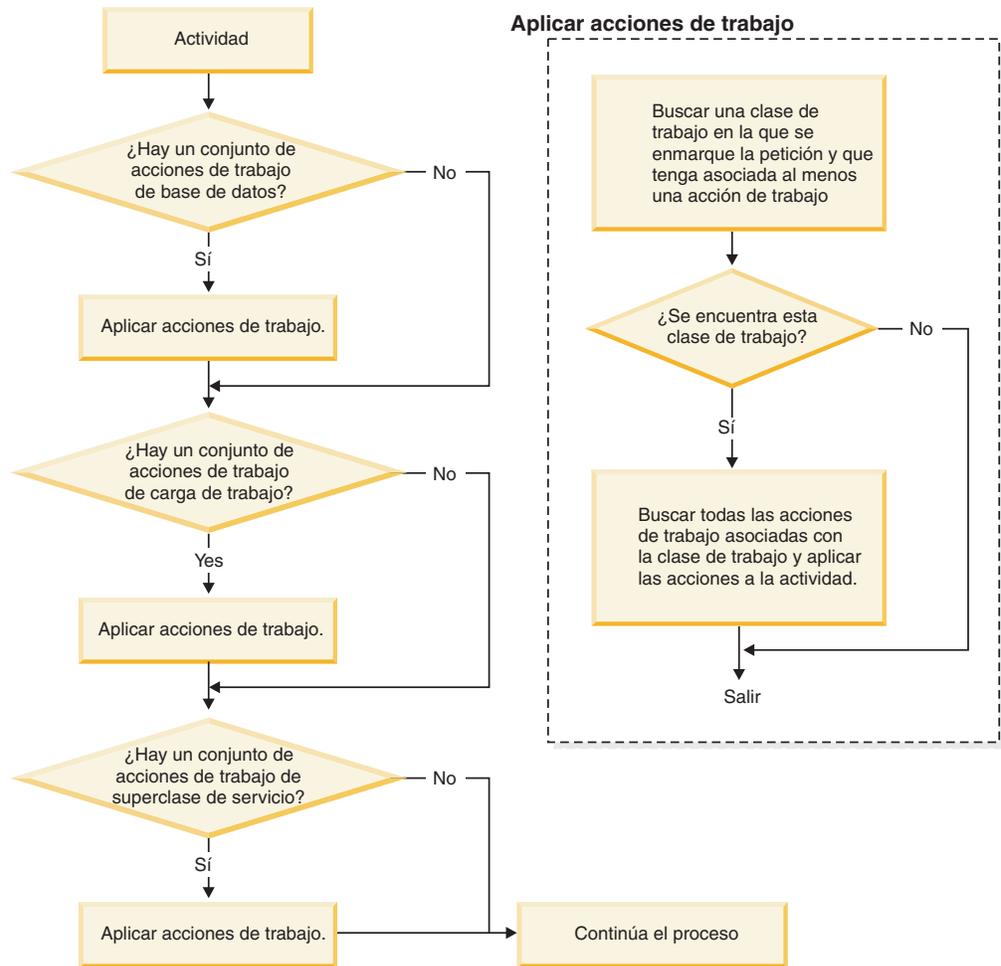


Figura 18. Aplicación de una acción de trabajo a una actividad

Una acción de trabajo se asigna a una actividad del modo siguiente:

1. Cuando una actividad se correlaciona con una superclase de servicio o una subclase de servicio, el servidor de datos comprueba si existe un conjunto de acciones de trabajo de nivel de base de datos habilitado.
2. Si existe un conjunto de acciones de trabajo de nivel de base de datos habilitado, el servidor de datos comprueba si la actividad se corresponde con alguna de las clases de trabajo del conjunto de clases de trabajo al que está asociado el conjunto de acciones de trabajo de nivel de base de datos.
3. Si la actividad se corresponde con una clase de trabajo y esta clase de trabajo tiene aplicadas acciones de trabajo, esas acciones de trabajo se aplicarán a la actividad.
4. El servidor de datos comprueba si existe un conjunto de acciones de trabajo de nivel de carga de trabajo habilitado. Si es así, el servidor de datos comprueba si la actividad se corresponde con alguna de las clases de trabajo del conjunto de clases de trabajo al que está asociado el conjunto de acciones de trabajo de nivel de carga de trabajo.
5. Si la actividad se corresponde con una clase de trabajo y esta clase de trabajo tiene aplicadas acciones de trabajo, esas acciones de trabajo se aplicarán a la actividad.

6. A continuación, si la carga de trabajo ha correlacionado la actividad con una superclase de servicio, el servidor de datos comprueba si el conjunto de acciones de trabajo se ha aplicado a la superclase de servicio.
7. Si se ha aplicado un conjunto de acciones de trabajo a la superclase de servicio, el servidor de datos comprueba si la actividad se corresponde con alguna de las clases de trabajo del conjunto de clases de trabajo al que está asociado el conjunto de acciones de trabajo de nivel de superclase de servicio.
8. Si la actividad se corresponde con una clase de trabajo y esta clase de trabajo tiene aplicadas acciones de trabajo, esas acciones de trabajo se aplicarán a la actividad.

Tenga en cuenta que, si se aplica una acción de trabajo de correlación a un procedimiento almacenado, en función de si se ha especificado la cláusula WITH NESTED o WITHOUT NESTED en la definición de la acción de trabajo, las actividades secundarias del procedimiento almacenado pueden ejecutarse en la misma subclase de servicio o en subclases de servicio diferentes de la de la actividad principal.

En las situaciones siguientes una actividad no se ve afectada por un conjunto de acciones de trabajo:

- Las actividades corresponden a las clases de servicio del sistema por omisión (SYSDEFAULTSYSTEMCLASS) y de mantenimiento por omisión (SYSDEFAULTMAINTENANCECLASS).
- Las actividades están asignadas a la carga de trabajo de administración por omisión, SYSDEFAULTADMWORKLOAD.
- Las actividades están dentro de una operación de carga. La misma operación de carga realiza la evaluación del conjunto de acciones de trabajo.
- Actividades secundarias de procedimientos almacenados de sistema. La única excepción es el procedimiento almacenado SYSPROC.ADMIN\_CMD. Las actividades secundarias de SYSPROC.ADMIN\_CMD realizan la evaluación del conjunto de acciones de trabajo.
- El conjunto de acciones de trabajo está inhabilitado.
- La carga de trabajo correlaciona la actividad directamente con una subclase de servicio.

## **Control de simultaneidad a nivel de carga de trabajo mediante conjuntos de acciones de trabajo**

A nivel de carga de trabajo, puede controlar la simultaneidad del trabajo entrante mediante umbrales de simultaneidad que se aplican con un conjunto de acciones de trabajo definidas en la carga de trabajo. Puede combinar el control de simultaneidad a nivel de carga de trabajo con la disminución de la prioridad a nivel de clase de servicio.

La Figura 1 ilustra un caso de ejemplo donde se utilizan conjuntos de acciones de trabajo para controlar la simultaneidad de trabajo entrante basándose en el origen de la conexión mientras que todo el trabajo de la base de datos se controla mediante la disminución de la prioridad.

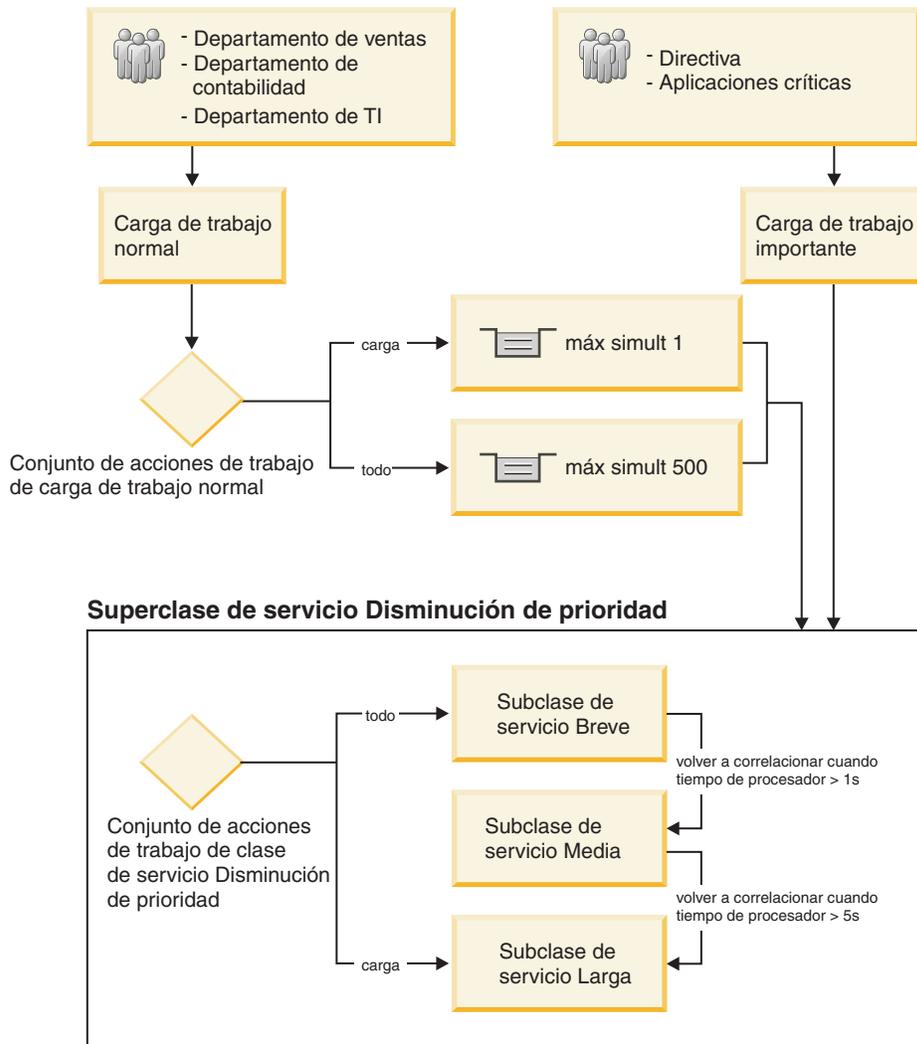


Figura 19. Control de simultaneidad a nivel de carga de trabajo mediante conjuntos de acciones de trabajo

En el caso de ejemplo, se crean dos cargas de trabajo para identificar y diferenciar el trabajo procedente de orígenes diferentes. Las conexiones a la base de datos desde los departamentos de ventas, contabilidad e informática se correlacionan con la carga de trabajo Normal. Las conexiones a la base de datos desde aplicaciones críticas y de gestión se correlacionan a la carga de trabajo Importante. El trabajo que procede de la carga de trabajo Importante tiene una prioridad más alta y debe poder completarse en el intervalo de tiempo más corto posible. Para garantizar que la base de datos tenga suficiente capacidad para el trabajo de la carga de trabajo Importante, los umbrales de simultaneidad se colocan en el trabajo de la carga de trabajo Normal. En la carga de trabajo Normal se crea un conjunto de acciones de trabajo a nivel de carga de trabajo, denominado Normal, y se aplica a un conjunto de clases de trabajo que tiene, a su vez, dos clases de trabajo. Las actividades de carga se correlacionan a una clase de trabajo, mientras que el resto de las actividades se correlacionan a la otra clase de trabajo. Se crea un umbral `CONCURRENTDBCOORDACTIVITIES` como acción de trabajo en el conjunto de acciones de trabajo a nivel de carga de trabajo Normal, para permitir tan solo una actividad de carga en el sistema a la vez, mientras que otras actividades de carga se ponen en cola. Además, se crea otro umbral `CONCURRENTDBCOORDACTIVITIES` como acción de trabajo en el conjunto de

acciones de trabajo a nivel de carga de trabajo Normal, para permitir un máximo de 500 actividades simultáneas, mientras que las actividades que sobrepasan el valor máximo se ponen en cola.

**Nota:** Si se especifica la acción de umbral CONTINUE para un umbral de cola, el tamaño de la cola no queda acotado, independientemente de los valores fijos que se hayan especificado para el tamaño de la cola.

Las conexiones a la base de datos desde las cargas de trabajo Normal e Importante se correlacionan con la superclase de servicio Disminución de prioridad. Esta superclase de servicio se crea para implementar la disminución de la prioridad que favorece las actividades breves. El conjunto de acciones de trabajo de la clase de servicio Disminución de prioridad se crea para que la superclase de servicio Disminución de prioridad separe las actividades de carga de ejecución prolongada de todas las actividades de ejecución breve. Todas las actividades que no sean de carga se correlacionan con la subclase de servicio Breve. La subclase de servicio Breve se configura de manera que tenga las prioridades más altas de agente, captación previa y agrupación de almacenamientos intermedios. Se crea un umbral CPUTIMEINSC en la subclase de servicio Breve para volver a correlacionar una actividad con la subclase de servicio Media después de que consuma más de 1 segundo de tiempo de procesador en la subclase de servicio Breve. La subclase de servicio Media tiene prioridad media de agente, captación previa y agrupación de almacenamientos intermedios. Se crea un umbral CPUTIMEINSC en la subclase de servicio Media para volver a correlacionar una actividad con la subclase de servicio Larga después de que consuma más de 5 minutos de tiempo de procesador en la subclase de servicio Media. La subclase de servicio Larga tiene la prioridad más baja de agente, captación previa y agrupación de almacenamientos intermedios. Las actividades de carga se correlacionan directamente con la subclase de servicio Larga mediante el conjunto de acciones de trabajo de la clase de servicio Disminución de prioridad porque las actividades de carga pueden tener un proceso largo, consumir muchos recursos y disponer de más tiempo para su finalización.

## **Comparación de cargas de trabajo y conjuntos de acciones de trabajo**

En función del tipo de control que desee instaurar en sus actividades de base de datos, puede utilizar las cargas de trabajo por sí solas o cargas de trabajo y clases de trabajo (si se utilizan con acciones de trabajo) para correlacionar las actividades con las clases de servicio.

En las cargas de trabajo, las peticiones se identifican y se asignan a una clase de servicio en función de los atributos de conexión. Las cargas de trabajo son el principal método para dirigir el trabajo a una clase de servicio de DB2 específica para su ejecución. Si desea definir con mayor precisión cómo se identifican las peticiones, puede utilizar clases de trabajo para clasificar las actividades en función de su tipo y otros atributos de actividades. Por ejemplo, puede clasificar actividades READ, actividades WRITE y actividades LOAD en diferentes clases de trabajo y hacer que cada tipo de actividad se trate de forma diferente.

Si utiliza clases de trabajo (que se agrupan en conjuntos de clases de trabajo), puede utilizar acciones de trabajo para ejercer control sobre los diferentes tipos de actividades. Por ejemplo, puede utilizar una acción de trabajo en un conjunto de acciones de trabajo definido en una superclase de servicio para correlacionar un tipo de actividad específico con una subclase de servicio. En un conjunto de acciones de trabajo definido en la base de datos o en la carga de trabajo, puede

definir una acción de trabajo para aplicar un control llamado umbral con el fin de garantizar que el mismo tipo de actividad no supere determinadas condiciones.

Las acciones de trabajo se agrupan en conjuntos de acciones de trabajo. Un conjunto de acciones de trabajo puede aplicarse a actividades en la base de datos, actividades en una superclase de servicio o actividades en una carga de trabajo. Sin embargo, el mismo conjunto de acciones de trabajo no puede aplicarse a más de un objeto. Los conjuntos de clases de trabajo y los conjuntos de acciones de trabajo funcionan conjuntamente. Es decir, es necesario que exista una clase de trabajo para clasificar una actividad como un tipo de trabajo específico antes de que se le pueda aplicar una acción de trabajo. Un conjunto de clases de trabajo puede asociarse con más de un conjunto de acciones de trabajo, pero un conjunto de acciones de trabajo sólo puede asociarse con un conjunto de clases de trabajo.

En la Figura 1 se muestra un ejemplo de implementación de gestor de carga de trabajo de DB2 que utiliza cargas de trabajo y conjuntos de acciones de trabajo. En esta figura, supongamos que se asigna una petición a la carga de trabajo WL\_A en función de los atributos de conexión de la conexión que ha enviado la petición. La carga de trabajo WL\_A especifica que la petición debe ejecutarse en la superclase de servicio SC\_A. Supongamos que una clase de trabajo en el conjunto de clases de trabajo WCS\_1 coincide con el tipo de trabajo que la petición asociada con la carga de trabajo WL\_A va a realizar.

Ahora supongamos que una actividad que no actualiza los catálogos (una actividad READ) accede al sistema. El conjunto de acciones de trabajo de nivel de base de datos WAS\_1 (asociado al conjunto de clases de trabajo WCS\_1) contiene una acción de trabajo que se aplica a la clase de trabajo READ. A continuación, la petición se correlaciona con la superclase de servicio SC\_A (mediante la carga de trabajo WL\_A). Aquí, la petición encuentra el conjunto de acciones de trabajo de nivel de superclase de servicio WAS\_2, que también está asociado con el conjunto de clases de trabajo WCS\_1, y se aplica a las actividades de la superclase de servicio SC\_A. Este conjunto de acciones de trabajo contiene una acción de trabajo de correlación, que también se aplica a la clase de trabajo READ, de modo que todas las actividades READ se correlacionarán con la subclase de servicio SSC\_1a en la superclase de servicio SC\_A.

Una situación algo parecida se produce con la petición asociada (nuevamente, en función de sus atributos de conexión) con la carga de trabajo WL\_B. La carga de trabajo WL\_B correlaciona las actividades con la superclase de servicio SC\_B. Supongamos que la petición es para una actividad LOAD y que el conjunto de clases de trabajo WCS\_2 contiene una clase de trabajo que se aplica a actividades LOAD. El conjunto de clases de trabajo WCS\_2 se asocia al conjunto de acciones de trabajo de nivel de superclase de servicio WAS\_3, que se aplica a actividades en la superclase de servicio SC\_B. Supongamos que el conjunto de acciones de trabajo WAS\_3 contiene una acción de trabajo de correlación que se aplica a la clase de trabajo LOAD, de modo que cuando una actividad LOAD se correlacione con la superclase de servicio SC\_B mediante la carga de trabajo WL\_B, la acción de trabajo la correlacionará a su vez con la subclase de servicio SSC\_1b para su ejecución.

La finalidad de la carga de trabajo WL\_C en este ejemplo consiste en correlacionar las peticiones entrantes directamente con la subclase de servicio SSC\_1b, independientemente del conjunto de acciones de trabajo de nivel de superclase de servicio WAS\_3 y su acción de trabajo de correlación. Si se asocia una petición entrante a la carga de trabajo WL\_C y es una actividad LOAD, esta petición también se correlacionará directamente con la subclase de servicio SSC\_1b para su

ejecución y no se verá afectada por la acción de trabajo de correlación que se aplica a la clase de trabajo LOAD.

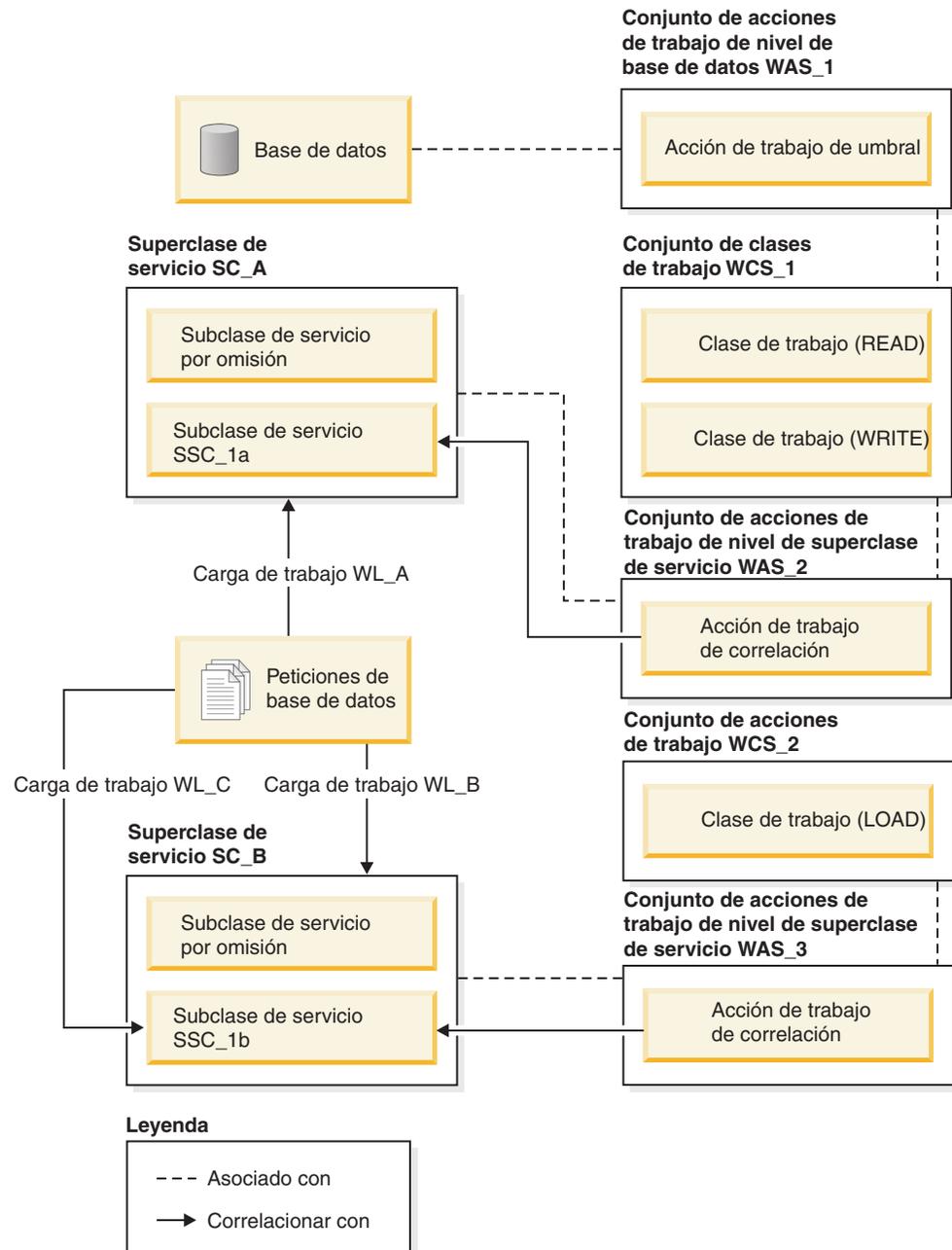


Figura 20. Cargas de trabajo y conjuntos de acciones de trabajo

## Ejemplo: Utilización de un conjunto de acciones de trabajo de base de datos y un umbral de base de datos

En este ejemplo se muestran diferentes enfoques en la utilización de conjuntos de acciones de trabajo y umbrales para controlar los recursos que consumen las actividades de DB2. Antes de crear objetos de gestión de carga de trabajo de DB2, debe comprender cómo se utilizan.

Supongamos que tiene un conjunto de clases de trabajo denominado ALLSQL, que contiene las siguientes clases de trabajo en este orden:

1. SMALLDML, que es para todas las sentencias de SQL de tipo DML con un coste estimado de menos de 1.000 timerons.
2. MEDDML, que es para todas las sentencias de SQL de tipo DML con un coste estimado de entre 1.000 y 20.000 timerons.
3. LARGEDML, que es para todas las sentencias de SQL de tipo DML con un coste estimado de más de 20.000 timerons.
4. ALLDDL, que es para todas las sentencias de SQL de tipo DDL.
5. ALLACTIVITY, que es para toda la actividad de la base de datos.

Las siguientes sentencias de SQL crean el conjunto de clases de trabajo y las clases de trabajo:

```
CREATE WORK CLASS SET ALLSQL
(WORK CLASS SMALLDML WORK TYPE DML FOR TIMERONCOST FROM 0 TO 1000,
WORK CLASS MEDDML WORK TYPE DML FOR TIMERONCOST FROM 1001 TO 20000,
WORK CLASS LARGEDML WORK TYPE DML FOR TIMERONCOST FROM 20001 TO UNBOUNDED,
WORK CLASS ALLDDL WORK TYPE DDL,
WORK CLASS ALLACTIVITY WORK TYPE ALL)
```

A estas clases de trabajo ya se les ha aplicado acciones de trabajo, como COUNT ACTIVITY, COLLECT, y umbrales (que no son umbrales ACTIVITYTOTALTIME).

Supongamos que desea permitir que las actividades de DML grandes se ejecuten durante no más de cinco horas. Todo el código SQL restante no puede tardar más de 30 minutos en ejecutarse. En los siguientes dos ejemplos se muestran posibles métodos para llevar a cabo este objetivo.

## Método 1

Un método consiste en configurar un conjunto de acciones de trabajo en el ámbito de una base de datos que contiene acciones de trabajo que especifican un umbral ACTIVITYTOTALTIME para cada clase de trabajo, como sigue:

Tabla 33. Umbral ACTIVITYTOTALTIME especificado por cada clase de trabajo

Acción de trabajo	Clase de trabajo a la que se aplica	Tipo y valor de umbral	Acciones
SMALLDMLTIMEALLOWED	SMALLDML	ACTIVITYTOTALTIME > 30 MINUTES	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
MEDDMLTIMEALLOWED	MEDDML	ACTIVITYTOTALTIME > 30 MINUTES	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
LARGEDMLTIMEALLOWED	LARGEDML	ACTIVITYTOTALTIME > 5 HOURS	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
ALLDDLTIMEALLOWED	ALLDDL	ACTIVITYTOTALTIME > 30 minutes	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>
ALLACTIVITYTIMEALLOWED	ALLACTIVITY	ACTIVITYTOTALTIME > 30 minutes	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>

Las sentencias de SQL para este método son:

```
CREATE WORK ACTION SET WASNICK FOR DATABASE USING WORK CLASS SET WCSNICK
(WORK ACTION SMALLDMLTIMEALLOWED ON WORK CLASS SMALLDML
  WHEN ACTIVITYTOTALTIME > 30 MINUTES COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION MEDDMLTIMEALLOWED ON WORK CLASS MEDDML
  WHEN ACTIVITYTOTALTIME > 30 MINUTES COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION LARGEDMLTIMEALLOWED ON WORK CLASS LARGEDML
  WHEN ACTIVITYTOTALTIME > 5 HOURS COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION ALLDDLTIMEALLOWED ON WORK CLASS ALLDDL
  WHEN ACTIVITYTOTALTIME > 30 MINUTES COLLECT ACTIVITY DATA STOP EXECUTION,
WORK ACTION ALLACTIVITYTIMEALLOWED ON WORK CLASS ALLACTIVITY
  WHEN ACTIVITYTOTALTIME > 30 MINUTES COLLECT ACTIVITY DATA STOP EXECUTION)
```

## Método 2

Otro método puede ser utilizar sólo una clase de trabajo, LARGEDML, y, a continuación, crear un conjunto de acciones de trabajo para la base de datos que sólo tenga una acción de trabajo, LARGEDMLTIMEALLOWED, aplicada a la clase de trabajo.

Tabla 34. Acción de trabajo LARGEDMLTIMEALLOWED aplicada a la clase de trabajo LARGEDML

Acción de trabajo	Clase de trabajo a la que se aplica	Tipo y valor de umbral	Acción
LARGEDMLTIMEALLOWED	LARGEDML	ACTIVITYTOTALTIME < 5 HOURS	<ul style="list-style-type: none"> <li>• Detener ejecución</li> <li>• Recopilar datos de actividad</li> </ul>

A continuación, aplicaría un umbral ACTIVITYTOTALTIME de menos de 31 MINUTES a la base de datos. Con este método, sólo se aplicará el umbral de cinco horas a aquellas actividades que correspondan a la clase de trabajo LARGEDML. A las demás actividades se les aplicará el umbral de tiempo de base de datos ACTIVITYTOTALTIME de menos de 31 minutos.

Las sentencias de SQL para este método son:

```
CREATE WORK ACTION SET WASNICK FOR DATABASE USING WORK CLASS SET WCSNICK
(WORK ACTION LARGEDMLTIMEALLOWED ON WORK CLASS LARGEDML
  WHEN ACTIVITYTOTALTIME > 5 HOURS COLLECT ACTIVITY DATA STOP EXECUTION)

CREATE THRESHOLD THTEST FOR DATABASE ACTIVITIES ENFORCEMENT DATABASE
  WHEN ACTIVITYTOTAL TIME > 30 MINUTES COLLECT ACTIVITY DATA STOP EXECUTION
```

## Ejemplo: Utilización de conjuntos de acciones de trabajo para determinar los tipos de trabajo que están en ejecución

Mediante la utilización de conjuntos de clases de trabajo, clases de trabajo, conjuntos de acciones de trabajo, acciones de trabajo y algunas de las funciones de supervisión del gestor de carga de trabajo de DB2, puede determinar los distintos tipos de trabajo que están ejecutándose en el sistema y la distribución del trabajo.

A continuación se describe una forma de llevar a cabo esta tarea. En primer lugar, debe crear un conjunto de clases de trabajo que contenga clases de trabajo para los distintos tipos de trabajo en los que está interesado. Por ejemplo, si desea conocer cuántas actividades READ, WRITE, DDL y LOAD están ejecutándose en el sistema, ha de crear un conjunto de clases de trabajo, ACTIVITYTYPES, tal como se muestra en el ejemplo siguiente:

```
CREATE WORK CLASS SET ACTIVITYTYPES
(WORK CLASS READWC WORK TYPE READ,
WORK CLASS WRITEWC WORK TYPE WRITE,
WORK CLASS DDLWC WORK TYPE DDL,
WORK CLASS LOADWC WORK TYPE LOAD)
```

A continuación, ha de crear un conjunto de acciones de trabajo de nivel de base de datos, COUNTACTIONS, para aplicarlo al conjunto de clases de trabajo ACTIVITYTYPES. El conjunto de acciones de trabajo ha de contener una acción de trabajo COUNT ACTIVITY para cada clase de trabajo del conjunto de clases de trabajo ACTIVITYTYPES, como en el ejemplo siguiente:

```
CREATE WORK ACTION SET COUNTACTIONS FOR DATABASE USING WORK CLASS SET ACTIVITYTYPES
(WORK ACTION COUNTREAD ON WORK CLASS READWC COUNT ACTIVITY,
WORK ACTION COUNTWRITE ON WORK CLASS WRITEWC COUNT ACTIVITY,
WORK ACTION COUNTDDL ON WORK CLASS DDLWC COUNT ACTIVITY,
WORK ACTION COUNTLOAD ON WORK CLASS LOADWC COUNT ACTIVITY)
```

Después de que haya transcurrido un período de tiempo suficiente, puede determinar el número de cada tipo de actividad que se ha ejecutado mediante la utilización de la función de tabla WLM\_GET\_WORK\_ACTION\_SET\_STATS:

```
SELECT SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
LAST_RESET,
SUBSTR(WORK_CLASS_NAME,1,15) AS WORK_CLASS_NAME,
SUBSTR(CHAR(ACT_TOTAL),1,14) AS TOTAL_ACTS
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS(CAST(NULL AS VARCHAR(128)), -2))
AS WASSTATS WHERE WORK_ACTION_SET_NAME = 'COUNTACTIONS'
ORDER BY WORK_CLASS_NAME, MEMB
```

---

## Control del trabajo con umbrales

Puede utilizar umbrales para mantener la estabilidad del sistema mediante la identificación de un trabajo cuyo comportamiento no es normal. Puede identificar un comportamiento no normal de forma predictiva, antes de que el trabajo empiece a ejecutarse, basándose en el impacto previsto. También puede identificar un comportamiento no normal de forma reactiva, a medida que el trabajo se ejecuta y consume recursos.

Un ejemplo de trabajo que puede controlarse con umbrales es una consulta que consume grandes cantidades de tiempo de procesador a expensas de todo el resto del trabajo que se ejecuta en el sistema. Esta consulta puede controlarse antes de que inicie su ejecución, basándose en el coste estimado, o después de que haya iniciado su ejecución y consuma más cantidad de recursos que lo permitido.

### Tipos de umbrales

#### Umbrales de conexión

Si desea limitar durante cuánto tiempo una conexión puede permanecer desocupada, utilice un umbral de conexión. Estos umbrales pueden utilizarse para detectar conexiones que están desocupadas durante un período de tiempo demasiado prolongado.

Tabla 35. Umbrales de conexión

Umbral	Descripción
CONNECTIONIDLETIME	Controla la cantidad de tiempo que una conexión está desocupada y no funciona en nombre de las peticiones de usuario. Utilice este umbral para detectar un uso no eficiente de los recursos del servidor de datos y las condiciones de espera de la aplicación.

### Umbrales de unidad de trabajo

Si desea limitar durante cuánto tiempo ha de ejecutarse una unidad de trabajo, utilice un umbral de unidad de trabajo. Estos umbrales limitan la cantidad máxima de tiempo que una unidad de trabajo puede emplear en el motor de DB2 y pueden utilizarse para detectar unidades que requieren demasiado tiempo para completarse.

Tabla 36. Umbrales de unidad de trabajo

Umbral	Descripción
UOWTOTALTIME	Controla la cantidad de tiempo que una unidad de trabajo tarda en ejecutarse.

### Umbrales de actividad

Si desea limitar el impacto que actividades específicas pueden tener en relación con la ejecución del servidor de datos, los umbrales de actividad le proporcionan un medio de hacerlo. Un tiempo de ejecución excesivo, la devolución de volúmenes de datos anormalmente altos o cantidades anormalmente altas de recursos consumidos son ejemplos avisos de la existencia de actividades potencialmente problemáticas que podrían consumir recursos excesivos, situación que puede controlar con umbrales de actividad.

Tabla 37. Umbrales de actividad

Umbral	Descripción
ACTIVITYTOTALTIME	Controla la cantidad de tiempo que cualquier actividad ha empleado desde su envío hasta su finalización, para el tiempo de ejecución y el tiempo en cola. Utilice este umbral para detectar trabajos que tardan una cantidad de tiempo anormalmente prolongada en completarse.
CPUTIME	Controla la cantidad máxima de tiempo combinado de procesador del sistema y de usuario que una actividad puede consumir en un miembro en particular durante la ejecución de la actividad. Utilice este umbral para detectar y controlar actividades que consumen excesivos recursos de procesador.
CPUTIMEINSC	Controla la cantidad máxima de tiempo combinado de procesador del sistema y de usuario que una actividad puede consumir en un miembro en particular al ejecutarse en una subclase de servicio específica. Utilice este umbral para detectar y controlar actividades que consumen excesivos recursos de procesador dentro de la clase de servicio actual.
DATATAGINSC	Controla los datos con los que una actividad podría estar relacionada o no al ejecutarse una subclase de servicio en particular.
ESTIMATEDSQLCOST	Controla las actividades de DML que el optimizador de consultas determina que tienen un coste estimado grande. Utilice este umbral para predecir SQL que potencialmente implicará gran consumo de recursos antes de que éste inicie su ejecución en el sistema e identificar SQL que se han escrito sin suficiente información.
SQLROWSREAD	Controla el número máximo de filas que una actividad puede leer en cualquier miembro de base de datos. Utilice este umbral para detectar y controlar actividades que leen un número excesivo de filas.
SQLROWSREADINSC	Controla el número máximo de filas que una actividad puede leer en un miembro en particular al ejecutarse en una subclase de servicio específica. Utilice este umbral para detectar y controlar actividades que leen un número excesivo de filas dentro de la clase de servicio actual.
SQLROWSRETURNED	Controla el número de filas que han de devolverse al ejecutar SQL. Utilice este umbral para identificar cuándo la cantidad de datos excede un volumen razonable.

Tabla 37. Umbrales de actividad (continuación)

Umbral	Descripción
SQLTEMPSPACE	Controla la cantidad de espacio de tablas temporal que una actividad en particular puede consumir en un miembro. Utilice este umbral para prevenir que determinadas sentencias de SQL utilicen una cantidad de espacio temporal desproporcionada que impida el progreso de otro trabajo.

El servidor de datos considera que las peticiones de programas de utilidad como REORGCHK, IMPORT y EXPORT son peticiones de lógica de usuario y, por lo tanto, estas peticiones están sujetas a cualquiera de los umbrales definidos.

#### Umbrales agregados

Si desea controlar el impacto colectivo que determinadas actividades, cargas de trabajo o conexiones pueden tener en nuestro servidor de datos, puede definir umbrales agregados. Los umbrales agregados con frecuencia, pero no siempre, aplican control de simultaneidad en aquellos casos en los que necesita limitar el número de determinadas actividades que se ejecutan al mismo tiempo. Algunos umbrales agregados incorporan una cola y se conocen como umbrales de colocación en cola.

Tabla 38. Umbrales agregados

Umbral	Descripción
AGGSQLTEMPSPACE	Controla la cantidad máxima de espacio de tablas temporal del sistema que pueden consumir en total todas las actividades de la subclase de servicio. Utilice este umbral para detectar y controlar actividades que pertenecen a una subclase de servicio cuyas actividades consumen demasiado espacio de tablas temporal del sistema en agregados en todas las actividades de la subclase de servicio.
CONCURRENTWORKLOADOCCURRENCES	Controla el número de apariciones activas de una carga de trabajo que pueden ejecutarse en un miembro coordinador al mismo tiempo. Se utiliza para controlar la dispersión de las conexiones desde un origen específico.
CONCURRENTWORKLOADACTIVITIES	Controla el número de actividades individuales que pueden ejecutarse dentro de una aparición de carga de trabajo. Se utiliza para limitar trabajo dentro de una aparición de carga de trabajo individual.
CONCURRENTDBCOORDACTIVITIES	Controla el número de actividades simultáneas del dominio al que se asocia el umbral (base de datos, acción de trabajo, superclase de servicio o subclase de servicio).
TOTALMEMBERCONNECTIONS	Controla el número de conexiones de base de datos con un miembro determinado que pueden establecerse al mismo tiempo. Se utiliza para impedir que un miembro concreto llegue a cargarse excesivamente.
TOTALSCMEMBERCONNECTIONS	Controla el número de conexiones de base de datos con un miembro determinado para el trabajo que se ejecuta dentro de una clase de servicio determinada al mismo tiempo. Es similar a las conexiones de miembro totales, pero es más granular porque la conexión se enlaza con una clase de servicio.

Para los umbrales agregados que dan soporte a ello, el control de la simultaneidad se proporciona mediante un sistema de "tíquets" de ejecución. Cada actividad de entrada debe reclamar un tíquet del umbral de simultaneidad aplicable antes de que pueda iniciar su ejecución.

Cuando se han consumido todos los tíquets, las actividades adicionales se colocan en cola hasta que un tíquet está disponible o bien se devuelve un error, en función de cómo ha definido el umbral. Si el umbral de simultaneidad tiene habilitada la colocación en cola, un tíquet pasa desde una actividad que ha completado su ejecución hasta otra actividad que está en la cola. Esta actividad puede, por lo tanto, abandonar la cola e iniciar su ejecución. La cantidad de tíquets que están disponibles por umbral de simultaneidad depende de cómo ha definido el umbral. Por ejemplo, si ha definido un umbral `CONCURRENTDBCOORDACTIVITIES` para limitar el número de actividades de base de datos en ejecución simultáneamente a 10, existen 10 tíquets de ejecución disponibles.

Para los procedimientos almacenados, se aplican umbrales de actividad al procedimiento almacenado y a sus actividades hijas. Los umbrales de simultaneidad sólo se aplican a las actividades hijas del procedimiento almacenado y no a la propia sentencia `CALL`.

## **Realización de acciones cuando se infringen umbrales**

La acción que ha de realizarse de forma dinámica cuando se infringe un umbral depende de cómo ha definido el umbral.

### **Detener la ejecución (STOP EXECUTION)**

Una acción común cuando se ha infringido un umbral es detener la ejecución de la actividad. En este caso, se devuelve un código de error a la aplicación que ha realizado el envío para indicar que se ha infringido el umbral. Tenga en cuenta que para los umbrales `TOTALMEMBERCONNECTIONS` y `TOTALSCMEMBERCONNECTIONS`, una acción `STOP EXECUTION` impide el establecimiento de una conexión. Para los umbrales `CONNECTIONIDLETIME`, la conexión se cierra. Para `CONCURRENTWORKLOADOCCURRENCES`, se impide la creación de una nueva aparición de carga de trabajo. Para todos los umbrales relacionados con actividades, la actividad se detiene para que no continúe su ejecución. Si existe un supervisor de sucesos `THRESHOLDVIOLATIONS` activo, se graba un registro en el supervisor de sucesos para indicar que se ha infringido el umbral.

### **Continuar la ejecución (CONTINUE)**

En algunas situaciones, detener la ejecución de una actividad es una respuesta excesiva. Una respuesta preferible sería permitir que la actividad continuara ejecutándose y recopilar los datos relacionados para que un administrador realice un análisis futuro y pueda determinar cómo evitar que vuelva a producirse esta condición. En esta situación, no se devuelve ningún código de error a la aplicación que realiza el envío. Si la acción es continuar, el usuario no recibe ninguna indicación acerca de la violación del umbral. Si existe un supervisor de sucesos `THRESHOLDVIOLATIONS` activo, se graba un registro en el supervisor de sucesos. Si se especifica una acción de umbral `CONTINUE` para un umbral de colocación en cola, éste representará con efectividad el tamaño de cola no acotado, con independencia de cualquier valor fijo que haya incluido.

### **Forzar la aplicación (FORCE APPLICATION)**

Cuando se infringe el umbral `UOWTOTALTIME`, puede especificar que ha de realizarse una desconexión del sistema forzada de los usuarios o aplicaciones locales o remotos. Este beneficiará a otras aplicaciones que compiten por los recursos del servidor.

### **Volver a correlacionar la actividad (REMAP ACTIVITY TO)**

Cuando una actividad infringe un límite determinado, puede que simplemente desee asignar distintos controles de recursos a ésta y permitir que la actividad siga ejecutándose. Esta respuesta le permite incrementar o reducir dinámicamente la cantidad de recursos que una actividad puede consumir durante toda su vida útil. En este caso, se permite que una actividad que ya está en ejecución continúe sin enviarse ninguna indicación al usuario o a la aplicación relacionada con la violación del umbral, aunque ahora la actividad se ejecute con distintos recursos disponibles. La actividad puede volver a correlacionarse con cualquiera de los umbrales de tipo en clase de servicio, como CPUTIMEINSC, SQLROWSREADINSC y DATATAGINSC.

### **Recopilar datos (COLLECT ACTIVITY DATA)**

Cuando se infringen algunos umbrales, se recopilan datos. Por omisión, el hecho de que se haya infringido un umbral de actividad se registra en un supervisor de sucesos de violaciones de umbral activado. Si desea obtener información más detallada acerca de la actividad que ha infringido el umbral, puede solicitar que la información de la actividad se grave en el supervisor de sucesos para actividades activo cuando la actividad haya completado su ejecución mediante la utilización de la cláusula COLLECT ACTIVITY DATA.

## **Dominio de umbral y ámbito de implantación**

Cada umbral opera en un dominio. El umbral solo afecta a las actividades que tienen lugar en el dominio de dicho umbral.

Existen los dominios de umbral siguientes:

- Base de datos
- Superclase de servicio
- Subclase de servicio
- Acción de trabajo
- Carga de trabajo
- Sentencia

En cada uno de estos dominios de umbral, un umbral tiene un ámbito en el que puede aplicarse, como una única aparición de carga de trabajo, un miembro, la ejecución de una sentencia específica o todos los miembros. Esto se denomina *ámbito de implantación* del umbral. Por ejemplo, los umbrales agregados de la clase de servicio pueden tener uno de los dos ámbitos de implantación posibles: base de datos y miembro. Un ejemplo de un umbral agregado que sólo se aplica en el nivel de miembro es el número máximo de conexiones simultáneas para una superclase de servicio de un miembro (TOTALSCMEMBERCONNECTIONS). Asimismo, la tabla siguiente demuestra que puede especificar el umbral de tiempo de procesador (CPUTIME) en la base de datos, superclase, subclase, acción de trabajo o dominio de carga de trabajo, y que se implanta en cada miembro. Es decir, el límite superior especifica la cantidad máxima de tiempo de procesador de usuario y de sistema por miembro que puede utilizar una actividad.

Tabla 39. Dominios de umbral y ámbitos de implantación

Dominio del umbral	Umbrales con ámbito de implantación de base de datos	Umbrales con ámbito de implantación de miembro	Umbrales con ámbito de implantación de aparición de carga de trabajo
Base de datos	<ul style="list-style-type: none"> <li>• “Umbral ACTIVITYTOTALTIME” en la página 144</li> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral CONNECTIONIDLETIME” en la página 142</li> <li>• “Umbral ESTIMATEDSQLCOST” en la página 148</li> <li>• “Umbral SQLROWSRETURNED” en la página 151</li> <li>• “Umbral UOWTOTALTIME” en la página 160</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154(El ámbito de implantación de base de datos solo se permite en entornos que no sean entornos DB2 pureScale. En un entorno DB2 pureScale, solo se permite el ámbito de implantación de miembro.)</li> <li>• “Umbral CPUTIME” en la página 145</li> <li>• “Umbral SQLROWSREAD” en la página 149</li> <li>• “Umbral SQLTEMPSPACE” en la página 152</li> <li>• “Umbral TOTALMEMBERCONNECTIONS” en la página 158</li> </ul>	No aplicable
Acción de trabajo	<ul style="list-style-type: none"> <li>• “Umbral ACTIVITYTOTALTIME” en la página 144</li> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral ESTIMATEDSQLCOST” en la página 148</li> <li>• “Umbral SQLROWSRETURNED” en la página 151</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral CPUTIME” en la página 145</li> <li>• “Umbral SQLROWSREAD” en la página 149</li> <li>• “Umbral SQLTEMPSPACE” en la página 152</li> </ul>	No aplicable
Superclase de servicio	<ul style="list-style-type: none"> <li>• “Umbral ACTIVITYTOTALTIME” en la página 144</li> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral CONNECTIONIDLETIME” en la página 142</li> <li>• “Umbral ESTIMATEDSQLCOST” en la página 148</li> <li>• “Umbral SQLROWSRETURNED” en la página 151</li> <li>• “Umbral UOWTOTALTIME” en la página 160</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral CPUTIME” en la página 145</li> <li>• “Umbral SQLROWSREAD” en la página 149</li> <li>• “Umbral SQLTEMPSPACE” en la página 152</li> <li>• “Umbral TOTALSCMEMBERCONNECTIONS” en la página 159</li> </ul>	No aplicable
Subclase de servicio	<ul style="list-style-type: none"> <li>• “Umbral ACTIVITYTOTALTIME” en la página 144</li> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral ESTIMATEDSQLCOST” en la página 148</li> <li>• “Umbral SQLROWSRETURNED” en la página 151</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral AGGSQLTEMPSPACE” en la página 153</li> <li>• “Umbral CONCURRENTDBCOORDACTIVITIES” en la página 154<sup>1</sup></li> <li>• “Umbral CPUTIME” en la página 145</li> <li>• “Umbral CPUTIMEINSC” en la página 146</li> <li>• “Umbral DATATAGINSC” en la página 147</li> <li>• “Umbral SQLROWSREAD” en la página 149</li> <li>• “Umbral SQLROWSREADINSC” en la página 150</li> <li>• “Umbral SQLTEMPSPACE” en la página 152</li> </ul>	No aplicable
Carga de trabajo	<ul style="list-style-type: none"> <li>• “Umbral ACTIVITYTOTALTIME” en la página 144</li> <li>• “Umbral ESTIMATEDSQLCOST” en la página 148</li> <li>• “Umbral SQLROWSRETURNED” en la página 151</li> <li>• “Umbral UOWTOTALTIME” en la página 160</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral CONCURRENTWORKLOADOCURRENCES” en la página 158</li> <li>• “Umbral CPUTIME” en la página 145</li> <li>• “Umbral SQLROWSREAD” en la página 149</li> <li>• “Umbral SQLTEMPSPACE” en la página 152</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral CONCURRENTWORKLOADACTIVITIES” en la página 156</li> </ul>
Sentencia	<ul style="list-style-type: none"> <li>• “Umbral ACTIVITYTOTALTIME” en la página 144</li> <li>• “Umbral ESTIMATEDSQLCOST” en la página 148</li> <li>• “Umbral SQLROWSRETURNED” en la página 151</li> </ul>	<ul style="list-style-type: none"> <li>• “Umbral CPUTIME” en la página 145</li> <li>• “Umbral SQLROWSREAD” en la página 149</li> <li>• “Umbral SQLTEMPSPACE” en la página 152</li> </ul>	No aplicable

## Orden de evaluación de umbrales

Los umbrales se evalúan en un orden específico cuando se definen en una base de datos.

Los umbrales siguientes se evalúan antes que los demás:

- TOTALMEMBERCONNECTIONS. Este umbral se evalúa cuando se establece una nueva conexión con una base de datos.
- CONCURRENTWORKLOADOCURRENCES. Este umbral se evalúa cuando se inicia una nueva aparición de carga de trabajo para una definición de carga de trabajo que tiene aplicado este umbral.
- TOTALSCMEMBERCONNECTIONS. Este umbral se evalúa cuando una conexión se une a una clase de servicio (ya sea una nueva conexión, ya sea una transferencia entre clases de servicio como resultado de una reasignación de la carga de trabajo).

Todos los demás umbrales se basan en actividades reconocidas resultantes de una sentencia de SQL o la ejecución de un programa de utilidad, como el programa de utilidad de carga, y se evalúan en el orden siguiente:

1. “Umbrales predictivos”
2. “Umbrales reactivos” en la página 137

## Umbrales predictivos

Los umbrales predictivos se comprueban antes que los umbrales reactivos, porque determinan si una actividad de la base de datos puede empezar a ejecutarse.

La secuencia en que se evalúan los umbrales predictivos que es la siguiente. Si no define un umbral en concreto, este paso se omite. Asimismo, los pasos descritos pueden combinarse en tiempo de ejecución por motivos de rendimiento.

1. Se comprueba si existe un umbral `CONCURRENTWORKLOADACTIVITIES` y, si es así, si se ha infringido. Si el umbral se ha infringido, se emprende la acción correspondiente. Si procede, se procede al paso siguiente.
2. Se comprueba si existe un umbral `ESTIMATEDSQLCOST` y, si es así, si se ha infringido. Si define este umbral en más de un dominio, el umbral se resuelve de acuerdo con las normas de resolución de ámbitos (consulte “Prioridad de dominio para umbrales de actividad” en la página 143 para obtener más información). El resultado de esta operación es un valor de `ESTIMATEDSQLCOST` aplicable a la actividad. Si el umbral se ha infringido, se emprende la acción correspondiente.
3. Se comprueba si existe un umbral `CONCURRENTDBCOORDACTIVITIES` para el dominio de umbrales del conjunto de acciones de trabajo de la carga de trabajo y, si es así, si se ha infringido. Si el umbral se ha infringido, se emprende la acción correspondiente.
4. Se comprueba si existe el umbral `CONCURRENTDBCOORDACTIVITIES` para el dominio de umbrales del conjunto de acciones de trabajo de la base de datos y, si es así, si se ha infringido. Si el umbral se ha infringido, se emprende la acción correspondiente.
5. Se comprueba si existe un umbral `CONCURRENTDBCOORDACTIVITIES` para el dominio de umbrales de la subclase de servicio y, si es así, si se ha infringido. Si el umbral se ha infringido, se emprende la acción correspondiente.
6. Se comprueba si existe un umbral `CONCURRENTDBCOORDACTIVITIES` para el dominio de umbrales de la superclase de servicio y, si es así, si se ha infringido. Si el umbral se ha infringido, se emprende la acción correspondiente.
7. Se comprueba si existe un umbral `CONCURRENTDBCOORDACTIVITIES` para el dominio de umbrales de la base de datos y, si es así, si se ha infringido. Si el umbral se ha infringido, se emprende la acción correspondiente.

**Consideraciones sobre umbrales de simultaneidad:** El orden de evaluación para umbrales de simultaneidad no sigue la jerarquía utilizada para resolver umbrales de actividad. Una actividad debe pasar por cada umbral de simultaneidad definido antes de que se permita su ejecución.

Para umbrales de simultaneidad, se comprueban primero los umbrales para los conjuntos de acciones de trabajo del nivel de la carga de trabajo, mientras que los conjuntos de acciones de trabajo del nivel de la base de datos se comprueban en segundo lugar. Los umbrales para conjuntos de acciones de trabajo se comprueban primero con el fin de evitar que los umbrales de conjuntos de acciones de trabajo en determinados tipos de trabajos bloqueen el trabajo de otros tipos, lo que podría

afectar a la simultaneidad. Por ejemplo, al comprobar primero los umbrales de simultaneidad del conjunto de acciones de trabajo del nivel de la base de datos se evita la siguiente situación.

Supongamos que se definen los siguientes umbrales:

- Se define un conjunto de acciones de trabajo de la base de datos que contiene un umbral de simultaneidad de acción de trabajo para actividades LOAD con el valor 1.
- El límite de simultaneidad de la superclase de servicio S1 se establece en 10.

Supongamos además que una actividad LOAD ya se está ejecutando en la base de datos (bajo cualquier superclase de servicio) y que nueve actividades ya se están ejecutando en la superclase de servicio S1. Una nueva actividad LOAD accede como décima actividad. Si se utilizara la jerarquía de resolución de ámbitos de umbrales de actividad durante la evaluación de los umbrales, la actividad LOAD entrante no infringiría el umbral de clase de servicio, con lo que se aumentaría la simultaneidad a 10. A continuación, la actividad LOAD se evalúa en relación con el límite de simultaneidad del umbral de acciones de trabajo del nivel de base de datos, que se infringe porque una actividad LOAD ya está en ejecución en la base de datos y el valor de simultaneidad del umbral de acciones de trabajo es tan sólo 1. Así, la actividad LOAD se colocará en cola.

Cualquier actividad nueva que llegue a la superclase de servicio S1 se colocará entonces en cola (porque ya se ha alcanzado el límite de simultaneidad de la clase de servicio). La cola del umbral de acciones de trabajo afectará a la clase de servicio, algo no deseable porque las actividades que intentan ejecutarse en la clase de servicio no necesariamente guardarán relación con la condición del umbral de acciones de trabajo (por ejemplo, una operación de inserción que intente ejecutarse en la superclase de servicio S1 no debería esperar a una actividad LOAD colocada en cola debido a una condición de umbral de acciones de trabajo del nivel de la base de datos). Por tanto, para evitar este tipo de situaciones, se comprueba el umbral de simultaneidad de acciones de trabajo del nivel de la base de datos antes que los umbrales de simultaneidad de subclase de servicio, superclase de servicio y base de datos. Debido a que el umbral de simultaneidad del conjunto de acciones de trabajo del nivel de la base de datos se comprueba en primer lugar, la décima actividad en la clase de servicio (que resulta ser una actividad LOAD) se bloquea en el nivel del umbral de acciones de trabajo de la base de datos antes de que pueda intentar consumir un espacio en la superclase de servicio S1.

## Umbrales reactivos

Los umbrales reactivos se evalúan de forma separada cuando una actividad se está ejecutando, y no se utiliza ningún orden específico para evaluarlos. Tiene disponibles los siguientes umbrales reactivos:

- ACTIVITYTOTALTIME
- AGGSQLTEMPSPACE
- CONNECTIONIDLETIME
- CPUTIME
- CPUTIMEINSC
- DATATAGINSC
- SQLTEMPSPACE
- SQLROWSREAD
- SQLROWSREADINSC

- SQLROWSRETURNED
- UOWTOTALTIME

## Creación de un umbral

Cree umbrales utilizando la sentencia DDL CREATE THRESHOLD (o la sentencia CREATE WORK ACTION SET). El umbral se crea para imponer un límite en el consumo de recursos.

### Antes de empezar

Para crear un umbral, necesita autorización WLMADM o DBADM.

Consulte los temas siguientes para obtener más información sobre los requisitos:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Normas de denominación

Para crear un umbral para un conjunto de acciones de trabajo, utilice la sentencia CREATE WORK ACTION SET o la sentencia ALTER WORK ACTION SET con las palabras clave ADD WORK ACTION. Para obtener más información, consulte la sentencia CREATE WORK ACTION SET o ALTER WORK ACTION SET.

### Procedimiento

Para crear un umbral:

1. Emita la sentencia CREATE THRESHOLD, especificando una o más de las propiedades siguientes para el umbral:
  - El nombre del umbral.
  - El dominio del umbral. El dominio del umbral es el objeto de base de datos al que el umbral está asociado y en el que opera. El dominio que se aplica depende del tipo de umbral; consulte “Dominio de umbral y ámbito de implantación” en la página 134 para obtener más información.
  - El ámbito de implantación del umbral. El ámbito del umbral es el rango de implantación del umbral en su dominio. El ámbito de implantación que se aplica depende del tipo de umbral; consulte “Dominio de umbral y ámbito de implantación” en la página 134 para obtener más información.
  - Opcional: inhabilite el umbral en el momento de crearlo. Por omisión, el umbral se crea habilitado. Si crea el umbral inhabilitado y espera a habilitarlo en otro momento, utilice la sentencia ALTER THRESHOLD.
  - El predicado del umbral para especificar el tipo de umbral y el valor máximo permitido. Cuando se infringe el valor máximo, se impone la acción especificada para el umbral. Para obtener más información sobre los umbrales que tiene disponibles, consulte “Umbrales de conexión” en la página 142, “Umbrales de actividad” en la página 143, “Umbrales agregados” en la página 153 y “Umbrales de unidad de trabajo” en la página 160.
  - Las acciones que se toman si se sobrepasa el valor máximo del umbral. Las acciones constan de una acción obligatoria que afecta a la ejecución de la actividad (STOP EXECUTION, CONTINUE, FORCE APPLICATION o REMAP ACTIVITY TO) y una acción de actividad de recopilación opcional (COLLECT ACTIVITY DATA). Las opciones que especifique para la acción de actividad de recopilación determinan qué información se recopila para la actividad que originó la infracción de la limitación del umbral.
2. Confirme los cambios. Cuando confirma los cambios, el umbral se añade a la vista SYSCAT.THRESHOLDS.

## Alteración de un umbral

Altere los umbrales utilizando la sentencia ALTER THRESHOLD. Puede alterar un umbral para modificar la limitación impuesta sobre un recurso específico.

### Antes de empezar

Para alterar un umbral, necesita la autorización SQLADM, WLMADM o DBADM. Para especificar una cláusula que no sea COLLECT, el ID de autorización debe incluir la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

Para alterar un umbral para un conjunto de acciones de trabajo, utilice la sentencia ALTER WORK ACTION SET con las palabras clave ADD WORK ACTION.

### Restricciones

No puede alterar un tipo de umbral que tenga la sentencia ALTER THRESHOLD. Por ejemplo: no puede cambiar un umbral TOTAL MEMBERCONNECTIONS por un umbral TOTALSCMEMBERCONNECTIONS. Si necesita un tipo de umbral diferente, descarte los umbrales existentes y a continuación cree un umbral nuevo.

### Procedimiento

Para alterar un umbral:

1. Especifique una o más de las propiedades siguientes para el umbral en la sentencia ALTER THRESHOLD. Puede cambiar las propiedades siguientes:
  - Los límites del predicado del umbral.
  - Las acciones a realizar, si se infringen los límites del umbral.
  - Si el umbral está habilitado o inhabilitado.
2. Confirme los cambios. Cuando confirma los cambios, el umbral se actualiza en la vista SYSCAT.THRESHOLDS.

## Descarte de un umbral

Descarte un umbral que ya no necesite utilizando la sentencia de DDL DROP THRESHOLD.

### Antes de empezar

Para descartar un umbral, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

Si desea descartar un umbral en un conjunto de acciones de trabajo, utilice la sentencia ALTER WORK ACTION SET. También puede descartar un umbral descartando completamente WORK ACTION SET mediante la sentencia DROP.

### Procedimiento

Para descartar un umbral:

1. Siga uno de estos pasos:

- Si el umbral es un umbral de puesta en cola, utilice la sentencia ALTER THRESHOLD para inhabilitarlo.
  - Si inhabilita un umbral de puesta en cola mediante la sentencia ALTER THRESHOLD, emita una sentencia COMMIT para confirmar el cambio.
2. Utilice la sentencia DROP THRESHOLD para descartar el umbral.
  3. Confirme los cambios. Cuando confirma los cambios, el umbral se elimina de la vista SYSCAT.THRESHOLDS.

## Ejemplo: Utilización de umbrales

Puede utilizar umbrales para una serie de finalidades. En este caso de ejemplo, se utilizan umbrales para controlar el número de trabajos grandes en ejecución, con el fin de permitir diferentes tiempos de ejecución a las diferentes aplicaciones y controlar el comportamiento de una aplicación que está en desarrollo.

Puede utilizar una solución de gestión de carga de trabajo de DB2 para dividir y gestionar los recursos de la base de datos para los diversos departamentos de una empresa. Por ejemplo, supongamos que el departamento de ventas ejecuta dos informes principales, que contienen las ventas mensuales y anuales. Supongamos también que el departamento de recursos humanos ejecuta una aplicación de nóminas en semanas alternas y que el equipo de desarrollo está trabajando en un nuevo tipo de informe a petición del equipo de dirección. Para definir entornos de ejecución diferentes para estos departamentos, cree clases de servicio:

```
CREATE SERVICE CLASS SALES
CREATE SERVICE CLASS HUMANRESOURCES
CREATE SERVICE CLASS DEVELOPMENT
```

En este caso, creará una definición de carga de trabajo por cada una de estas aplicaciones con el fin de correlacionar la aplicación con su superclase de servicio correspondiente:

```
CREATE WORKLOAD MONTHLYSALES APPLNAME('monthlyrpt.exe') SERVICE CLASS SALES
CREATE WORKLOAD YEARLYSALES APPLNAME('yearlyrpt.exe') SERVICE CLASS SALES
CREATE WORKLOAD PAYROLL APPLNAME('payroll.exe') SERVICE CLASS HUMANRESOURCES
CREATE WORKLOAD NEWREPORT APPLNAME('dev.exe') SERVICE CLASS DEVELOPMENT
```

Por tanto, el catálogo de la base de datos contiene las siguientes definiciones de carga de trabajo:

- MonthlySales, que se correlaciona con la superclase de servicio Sales
- YearlySales, que se correlaciona con la superclase de servicio Sales
- Payroll, que se correlaciona con la superclase de servicio Human Resources
- NewReport, que se correlaciona con la superclase de servicio Development

## Umbral sobre el número de trabajos grandes

Puesto que el informe YearlySales es muy grande, no desea que haya más de una aparición de esta aplicación en ejecución en la base de datos al mismo tiempo. Por tanto, crea un umbral para establecer el número máximo de apariciones simultáneas de esta carga de trabajo en 1:

```
CREATE THRESHOLD SINGLEYEARLYSALESRPT FOR WORKLOAD YEARLYSALES ACTIVITIES
ENFORCEMENT MEMBER
WHEN CONCURRENTWORKLOADOCCURRENCES > 1
STOP EXECUTION
```

Puede obtener una solución similar si asocia la aplicación YearlySales a una subclase de servicio YearlySalesReports (bajo la superclase de servicio Sales) y establece el umbral de simultaneidad máxima en un valor 1 para la subclase de servicio:

```
CREATE SERVICE CLASS YEARLYSALESREPORTS UNDER SALES

ALTER WORKLOAD YEARLYSALES SERVICE CLASS YEARLYSALESREPORTS UNDER SALES

CREATE THRESHOLD SINGLEYEARLYSALESREPORT FOR SERVICE CLASS YEARLYSALESREPORTS
  UNDER SALES ACTIVITIES ENFORCEMENT DATABASE
  WHEN CONCURRENTDBCOORDACTIVITIES > 1
  STOP EXECUTION
```

En ambas situaciones, puede establecer la acción del umbral en STOP EXECUTION para impedir que se ejecute más de una aparición de la carga de trabajo. También puede recopilar información de la actividad si desea información adicional sobre las condiciones en que se infringió el umbral.

### **Umbral sobre la vida útil de las actividades**

Puesto que se espera que todas las aplicaciones se completen en menos de una hora, puede crear un umbral con un dominio de base de datos, lo que impedirá que cualquier actividad se ejecute durante más de una hora. La única excepción a esta norma es el informe anual, que puede requerir hasta cinco horas para completarse. Por tanto, puede asociar un umbral de tiempo total de actividad de cinco horas a la carga de trabajo YearlySales. Así se altera temporalmente el umbral de tiempo total de actividad que se aplica al informe de ventas anual, con lo que se relajan las restricciones de tiempo. El nuevo valor de cinco horas se aplica ahora a la carga de trabajo YearlySales, si bien el valor global de una hora se aplicará en los demás casos en la base de datos:

```
CREATE THRESHOLD MAXDBACTIVITYTIME FOR DATABASE ACTIVITIES
  ENFORCEMENT DATABASE
  WHEN ACTIVITYTOTALTIME > 1 HOUR
  STOP EXECUTION

CREATE THRESHOLD MAXYRPTACTIVITYTIME FOR WORKLOAD YEARLYSALES
  ACTIVITIES ENFORCEMENT DATABASE
  WHEN ACTIVITYTOTALTIME > 5 HOURS
  STOP EXECUTION
```

### **Umbral sobre el número de actividades de coordinador y anidadas**

La aplicación NewReport utiliza de forma intensiva procedimientos almacenados y funciones definidas por el usuario, y todavía no se ha depurado totalmente, por lo que tiende a generar un gran número de actividades que inciden en el resto del sistema. Después de consultar con el desarrollador, se entera de que se supone que este nuevo informe no generará más de 20 actividades en total, de modo que define un umbral de tipo de actividades de carga de trabajo en la carga de trabajo NewReport y lo establece en 20. Inicialmente, establece la acción del umbral en STOP EXECUTION y COLLECT ALL para detener cualquier efecto secundario no deseado de la aplicación al iniciar un gran número de actividades y para ayudar al desarrollador a identificar los problemas:

```
CREATE THRESHOLD MAXDEVACTIVITIES FOR SERVICE CLASS DEVELOPMENT ACTIVITIES
  ENFORCEMENT DATABASE
  WHEN CONCURRENTDBCOORDACTIVITIES > 20
  COLLECT ACTIVITY DATA WITH DETAILS AND VALUES
  STOP EXECUTION
```

Cuando la aplicación se vuelve más estable, pasa a la fase de optimización. Durante esta fase, el desarrollador intenta reducir el número de actividades que genera la aplicación de entre 15 y 20 a 15. En este momento, modifica el umbral cambiando el valor de límite superior por 15 y la acción del umbral por CONTINUE. Esta definición de umbral ayuda a identificar y resolver situaciones en las que el número de actividades generadas supera las 15, pero en las que una mayor estabilidad de la aplicación no hace necesario detener su ejecución.

```
ALTER THRESHOLD MAXDEVACTIONS
  WHEN CONCURRENTDBCOORDACTIVITIES > 15
  COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS AND VALUES
  CONTINUE
```

## Umbral para limitar la duración de la ejecución de una unidad de trabajo

La aplicación LongUOW emite transacciones que ocasionalmente se ejecutan durante más tiempo que los diez minutos deseados. Esto da como resultado que se mantengan bloqueos durante demasiado tiempo, e impide que aplicaciones más importantes puedan continuar. En este caso, desea forzar la aplicación, en lugar de permitir que retenga otros trabajos. Puede restringir el tiempo de ejecución de las transacciones de esta aplicación a un período de tiempo definido por el administrador utilizando el umbral UOWTOTALTIME.

Primero cree una carga de trabajo para la aplicación LongUOW:

```
CREATE WORKLOAD LONG_UOW APPLNAME('LONGUOW') SERVICE CLASS SYSDEFAULTUSERCLASS
```

A continuación, cree un umbral para esta carga de trabajo que fuerce la aplicación LongUOW cuando alguna de las transacciones de la aplicación tarde más de diez minutos en finalizar:

```
CREATE THRESHOLD FORCELONGUOW FOR WORKLOAD LONG_UOW ACTIVITIES ENFORCEMENT DATABASE
  WHEN UOWTOTALTIME > 10 MINUTES FORCE APPLICATION
```

También puede aplicar este umbral en el nivel de subclase de servicio o en el nivel de base de datos.

## Umbrales de conexión

Un umbral de conexión aplica controles a las diferentes conexiones de la base de datos. Puede establecer umbrales de conexión para limitar el número total de conexiones simultáneas a la base de datos y el tiempo que una conexión puede permanecer desocupada.

### Umbral CONNECTIONIDLETIME

El umbral CONNECTIONIDLETIME especifica el tiempo máximo que una conexión puede estar inactiva (es decir, sin ejecutar una petición de usuario).

**Tipo** Conexión

**Dominio de definición**

Superclase de servicio o base de datos

**Ámbito de implantación**

Base de datos

**Trabajo con seguimiento**

Conexiones de usuario

**Colocación en cola**

No

**Unidad**

Intervalo de tiempo expresado en minutos, horas o días

**Predictivo o reactivo**

Reactivo

Si una conexión permanece inactiva durante más tiempo del especificado en el umbral, y la acción del umbral es STOP EXECUTION, la conexión se cierra.

## Umbrales de actividad

Un umbral de actividad se aplica a una actividad individual. Cuando la utilización de recursos de una determinada actividad infringe el límite superior del umbral que realiza el seguimiento, se desencadena la acción correspondiente y se aplica una vez a la actividad.

Después de que se aplique una vez, el umbral se desactiva para esa actividad y no se vuelve a aplicar.

Por ejemplo: supongamos que ha definido un umbral de tipo temporal que desencadena una acción CONTINUE transcurridos 5 minutos. Si una actividad infringe este umbral, la acción se aplica una vez, pero no se vuelve a aplicar cada 5 minutos.

### Prioridad de dominio para umbrales de actividad

Los umbrales de actividad se aplican a actividades individuales. Si varios umbrales se aplican a la misma actividad en ejecución, debe tomarse una decisión relacionada con qué umbral ha de aplicarse.

Los umbrales agregados no se ven afectados porque la misma actividad puede contribuir en varios agregados de actividades simultáneamente, como sucede con los umbrales de simultaneidad, por ejemplo.

La resolución sobre qué umbral de actividad debe aplicarse a una actividad en ejecución sigue la norma según la cual un valor definido en un dominio local prevalece sobre cualquier valor de un dominio más amplio o global. A continuación se muestra la jerarquía de dominios, del más local al más global:

1. Sentencia
2. Acción de trabajo (en el nivel de carga de trabajo)
3. Carga de trabajo
4. Subclase de servicio
5. Superclase de servicio
6. Acción de trabajo (en el nivel de base de datos)
7. Base de datos

En el ejemplo siguiente se muestra cómo se alteran temporalmente los umbrales:

### Ejemplos

En el ejemplo siguiente se muestra cómo se alteran temporalmente los umbrales:

- Un umbral que define un tiempo de ejecución máximo de 1 hora para todas las consultas de base de datos definidas en el dominio de base de datos se ve alterado temporalmente por un umbral que define un tiempo de ejecución máximo de 5 horas para una superclase de servicio configurada para gestionar consultas grandes.

- Esa misma superclase de servicio se ve alterada temporalmente por un umbral para consultas muy grandes que define el tiempo de ejecución máximo de 10 horas para una subclase de servicio.
- El tiempo de ejecución máximo de 1 hora definido en el dominio de base de datos puede alterarlo temporalmente un valor de 10 minutos en una segunda superclase de servicio cuyo objetivo sea garantizar que las consultas más cortas e importantes podrán completarse con rapidez.
- La ejecución de una sentencia con texto que coincide con el texto que se ha especificado para un umbral en el dominio de la sentencia da como resultado una violación de umbral que altera temporalmente todos los demás umbrales.

### **Umbral ACTIVITYTOTALTIME**

El umbral ACTIVITYTOTALTIME especifica el tiempo máximo que debe dedicar el servidor de datos en procesar una actividad. Si el umbral de puesta en cola pone en cola la actividad, el tiempo de actividad total incluye el tiempo empleado en la ejecución de la espera en cola.

**Tipo** Actividad

#### **Dominio de definición**

Base de datos, superclase de servicio, subclase de servicio, acción de trabajo, carga de trabajo y sentencia

#### **Ámbito de implantación**

Base de datos

#### **Trabajo con seguimiento**

Coordinador y actividades anidadas reconocidos (vea “Actividades” en la página 15)

#### **Colocación en cola**

No

#### **Unidad**

Intervalo de tiempo expresado en segundos, minutos, horas o días. Si la unidad de tiempo especificada es segundos, este valor debe ser un múltiple de 10.

#### **Predictivo o reactivo**

Reactivo

Cuando se abre el cursor, la actividad asociada con el cursor se prolonga hasta que el cursor se cierra.

Cuando se aplica un umbral de tiempo a un procedimiento almacenado, también se aplica al trabajo que tiene lugar en el procedimiento almacenado. En consecuencia, cuando caduca un umbral de tiempo del procedimiento almacenado, cualquier trabajo que tenga lugar en el procedimiento almacenado se detiene. Las jerarquías de las invocaciones de procedimientos almacenados pueden conducir a jerarquías de umbrales de tiempo que se estén aplicando a actividades que se ejecutan en los niveles más internos de anidación. Siempre se aplica el umbral de tiempo más restrictivo de la jerarquía (es decir, el umbral de tiempo que tiene la fecha de caducidad más próxima).

El servidor de datos considera IMPORT, EXPORT y otros mandatos CLP como lógica de usuario. Las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales.

## Umbral CPUTIME

El umbral CPUTIME especifica la cantidad máxima de tiempo de procesador de la combinación usuario/sistema que puede utilizar una actividad en un miembro particular mientras se ejecuta la actividad. Utilice este umbral para detectar y controlar las actividades que utilizan demasiados recursos de procesador.

**Tipo** Actividad

### Dominio de definición

Base de datos, superclase de servicio, subclase de servicio, acción de trabajo, carga de trabajo y sentencia

### Ámbito de implantación

Miembro

### Trabajo con seguimiento

Consulte la información que se encuentra más adelante en este tema

### Colocación en cola

No

### Unidad

Tiempo

### Predictivo o reactivo

Reactivo

El tiempo de procesador que tarda una actividad en ejecutarse se mide desde el momento en que la actividad comienza a ejecutarse en el miembro, después de que los umbrales coloquen elementos en cola, hasta el momento en que la actividad termina de ejecutarse.

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Todas las actividades de DML.
- Actividades CALL. El tiempo de procesador de una actividad CALL no incluye el tiempo de procesador de una actividad hija cualquiera. El tiempo de procesador que se emplea en procesos delimitados tampoco cuenta para el tiempo de procesador total de la actividad CALL.

Las actividades que inicia el gestor de base de datos a través de un programa de utilidad o de un procedimiento, con la excepción del procedimiento ADMIN\_CMD, no cuentan para esta condición. El servidor de datos considera IMPORT, EXPORT y otros mandatos CLP como lógica de usuario. Las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales. Este umbral no hace un seguimiento de las actividades hijas del mandato LOAD.

## Ejemplo

En el ejemplo siguiente se crea un umbral CPUTIME TH1 para el dominio de la base de datos con un ámbito de implantación del miembro. Este umbral detiene cualquier actividad que tarde más de 30 segundos en ejecutarse, lo cual se comprueba a intervalos de 5 segundos. Puede utilizar este umbral para asegurarse de que ninguna consulta del sistema utiliza una cantidad de tiempo de procesador poco razonable, lo cual podría impactar en otros trabajos que se estén ejecutando en el sistema.

```
CREATE THRESHOLD TH1 FOR DATABASE ACTIVITIES
  ENFORCEMENT MEMBER
  WHEN CPUTIME > 30 SECONDS CHECKING EVERY 5 SECONDS
  STOP EXECUTION;
```

## Umbral CPUTIMEINSC

El umbral de tipo en clase de servicio CPUTIMEINSC especifica la cantidad máxima de tiempo de procesador de la combinación usuario/sistema que puede utilizar una actividad en un miembro mientras se ejecuta en una subclase de servicio específica. Utilice este umbral para detectar y controlar las actividades que utilizan demasiados recursos de procesador.

**Clase** Actividad

### **Dominio de definición**

Subclase de servicio

### **Ámbito de implantación**

Miembro

### **Trabajo con seguimiento**

Consulte la información que se encuentra más adelante en este tema

### **Colocación en cola**

No

### **Unidad**

Tiempo

### **Predictivo o reactivo**

Reactivo

El tiempo de procesador que tarda una actividad en ejecutarse se mide desde el momento en que la actividad entra en la subclase de servicio actual hasta que la abandona o hasta termina de ejecutarse.

Este umbral difiere del umbral CPUTIME en que controla solamente la cantidad de tiempo de procesador que puede utilizarse en una subclase de servicio específica, no la cantidad total de tiempo de procesador que se utiliza durante la vida útil de la actividad.

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Todas las actividades de DML.
- Actividades CALL. Tenga en cuenta que el tiempo de procesador de una actividad CALL no incluye el tiempo de procesador de una actividad hija cualquiera. El tiempo de procesador que se emplea en procesos delimitados tampoco cuenta para el tiempo de procesador total de la actividad CALL.

Las actividades que inicia el gestor de base de datos a través de un programa de utilidad o de un procedimiento, con la excepción del procedimiento ADMIN\_CMD, no cuentan para esta condición. El servidor de datos considera IMPORT, EXPORT y otros mandatos CLP como lógica de usuario. Las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales. Este umbral no hace un seguimiento de las actividades hijas del mandato LOAD.

Puede utilizar la acción REMAP ACTIVITY para controlar actividades volviendo a correlacionarlas con una subclase de servicio con diferentes asignaciones de recursos.

## **Ejemplo**

En el ejemplo siguiente se crean dos subclases de servicio, A1 y A2, bajo una superclase A, con un único umbral de tipo en clase de servicio CPUTIMEINSC que vuelve a correlacionar las actividades entre subclases una vez transcurrido 1

minuto de tiempo de procesador durante la evaluación de consultas en la subclase de servicio A1. Queda anotado un registro del supervisor de sucesos.

```
CREATE SERVICE CLASS A;  
CREATE SERVICE CLASS A1 UNDER A;  
CREATE SERVICE CLASS A2 UNDER A;  
  
CREATE THRESHOLD T1 FOR SERVICE CLASS A1 UNDER A  
  ACTIVITIES ENFORCEMENT MEMBER  
  WHEN CPUTIMEINSC > 1 MINUTE CHECKING EVERY 30 SECONDS  
  REMAP ACTIVITY TO A2 LOG EVENT MONITOR RECORD;
```

## Umbral DATATAGINSC

El umbral de tipo en clase de servicio DATATAGINSC comprueba el valor del código de datos de un espacio de tablas o un grupo de almacenamiento al que una actividad accede. Utilice este umbral para correlacionar dinámicamente una actividad con una subclase de servicio de DB2 distinta en función de los datos a los que se accede.

**Tipo** Actividad

**Dominio de definición**  
Subclase de servicio

**Ámbito de implantación**  
Miembro

**Trabajo con seguimiento**  
Coordinador reconocido

**Colocación en cola**  
No

**Unidad**  
Código de datos

**Predictivo o reactivo**  
Reactivo

Los datos pueden codificarse con un valor numérico especificando el atributo DATA TAG al crear o alterar un espacio de tablas o un grupo de almacenamiento. Los umbrales de código de datos se evalúan al abrirse por primera vez una exploración en una tabla y cuando se realiza una inserción en una tabla. Cualquier nuevo umbral de código de datos que la actividad tome tras haberse abierto una exploración como resultado de una operación de nueva correlación no se aplicará a esa exploración.

El umbral sólo considerará los códigos de datos para los espacios de tablas de datos a los que se accede en una consulta. Los valores de los códigos de datos para los espacios de tablas de índice o largos no se tendrán en cuenta durante una evaluación de umbral. Por ejemplo, si la consulta SELECT COUNT(\*) FROM T1 realiza un acceso sólo al índice, y el índice está ubicado en un espacio de tablas que tiene el código de datos 3 y los datos de tabla están ubicados en un espacio de tablas que tiene el código de datos 1, cuando se ejecute la consulta, la evaluación de umbral utilizará el código de datos 1 (el código de datos para el espacio de tablas de datos). El comportamiento del umbral de código de datos es independiente del método de acceso (exploración de índice o directa) seleccionado en el plan de acceso de consulta.

A continuación se indican las actividades cuyo seguimiento realiza este umbral:

- Actividades de coordinador de tipo DML y trabajo de subagente correspondiente (ejecución de subsección).
- Actividades de DML anidadas derivadas de las aplicaciones de usuario. En consecuencia, este umbral no afecta a las actividades de DML que el gestor de bases de datos emite, como los programas de utilidad, los procedimientos SYSPROC (a excepción de SYSPROC.ADMIN\_CMD) o las sentencias de SQL internas.

En el siguiente escenario de uso, el usuario crea 3 espacios de tablas; TBHIGH contiene datos de alta prioridad y tiene el código de datos 1, TBMED contiene datos de prioridad media y tiene el código de datos 4 y TBLOW contiene datos de baja prioridad y tiene el código de datos 9. Las actividades se han correlacionado inicialmente con una subclase de servicio de alta prioridad. Si cualquiera de estas actividades está relacionada con datos de un espacio de tablas distinto de los del espacio de tablas TBHIGH, el usuario desea que la prioridad de esas actividades sea menor. En el ejemplo siguiente, una superclase de servicio MAINSC contiene tres subclases de servicio, SCHIGH, SCMED y SCLOW (para ejecutar trabajo de alta, media y baja prioridad), que ya se han creado.

Cree un umbral DATATAGINSC en la subclase de servicio de alta prioridad y, si cualquier actividad está relacionada con cualquiera de los datos que tienen un código de datos distinto de 1, correlacione ésta con la subclase de servicio de prioridad media.

```
CREATE THRESHOLD MAPTOMED FOR SERVICE CLASS SCHIGH UNDER MAINSC ACTIVITIES
  ENFORCEMENT MEMBER WHEN DATATAGINSC NOT IN (1) REMAP ACTIVITY TO SCMED
```

Cree un umbral SQLDATATAGINSC en la subclase de servicio de prioridad media y, si cualquier actividad está relacionada con cualquiera de los datos que tienen el código de datos 9, correlacione ésta con la subclase de servicio de baja prioridad.

```
CREATE THRESHOLD MAPTLOW FOR SERVICE CLASS SCMED UNDER MAINSC ACTIVITIES
  ENFORCEMENT MEMBER WHEN DATATAGINSC IN (9) REMAP ACTIVITY TO SCLOW
```

Si una actividad está relacionada con un espacio de tablas que tiene el código de datos 9, esa actividad volverá a correlacionarse con SCMED y, a continuación, volverá a correlacionarse inmediatamente con SCLOW (debido al umbral de la subclase de servicio SCMED). En este caso, el usuario verá dos violaciones de umbral.

## Umbral ESTIMATEDSQLCOST

El umbral ESTIMATEDSQLCOST especifica el coste máximo estimado permitido para las actividades de DML.

**Tipo** Actividad

### Dominio de definición

Base de datos, superclase de servicio, subclase de servicio, acción de trabajo, carga de trabajo y sentencia

### Ámbito de implantación

Base de datos

### Trabajo con seguimiento

Consulte la información que se encuentra más adelante en este tema

### Colocación en cola

No

### Unidad

Coste de SQL estimado expresado en timerons

## **Predictivo o reactivo**

Predictivo

Este umbral realiza el seguimiento de las actividades siguientes:

- Actividades de DML que se emiten en el miembro coordinador.
- Actividades de DML anidadas que se invocan desde una aplicación de usuario. En consecuencia, las actividades de DML que emite el servidor de datos internamente, como las actividades de DML emitidas desde los programas de utilidad de DB2, procedimientos almacenados SYSPROC y SQL interno, no se ven afectadas por este umbral a menos que su coste esté incluido en la estimación de la actividad padre. En esta situación, se hace un seguimiento indirecto de estas actividades. Un activador es un ejemplo de una actividad con seguimiento indirecto. IMPORT, EXPORT, y otros mandatos CLP se consideran como lógica de usuario. Las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales. Para obtener información sobre las actividades que se incluyen en una clase de trabajo con el tipo de trabajo DML, consulte “Identificación de trabajos por tipo, por coste o por medio de los datos a los que se accede con clases de trabajo” en la página 52.

## **Umbral SQLROWSREAD**

El umbral SQLROWSREAD especifica el número máximo de filas que puede leer una actividad de DML en cualquier miembro. Utilice este umbral para detectar y controlar actividades que lean un número excesivo de filas.

**Clase** Actividad

### **Dominio de definición**

Base de datos, superclase de servicio, subclase de servicio, acción de trabajo, carga de trabajo y sentencia

### **Ámbito de implantación**

Miembro

### **Trabajo con seguimiento**

Consulte la información que se encuentra más adelante en este tema

### **Colocación en cola**

No

### **Unidad**

Número de filas

## **Predictivo o reactivo**

Reactivo

Este umbral difiere del umbral SQLROWSRETURNED en que controla el número máximo de filas leídas durante la evaluación de una consulta, no el número de filas devueltas a una aplicación de cliente desde el servidor de datos.

Los accesos al índice no cuentan para el número total de filas leídas. Si un plan de acceso solo utiliza índices durante la evaluación de la consulta, no se infringe el umbral SQLROWSREAD.

Este umbral se evalúa a intervalos de tiempo configurables por el usuario; si el intervalo es mayor que la cantidad de tiempo que tarda en sobrepasar el número de filas leídas, es posible que el número de filas leídas de una actividad de un miembro sobrepase el límite del umbral antes de que se detecte la infracción.

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Las actividades coordinadoras del tipo DML y el trabajo de subagente correspondiente, como la ejecución de una subsección.
- Actividades de DML anidadas derivadas de las aplicaciones de usuario. En consecuencia, las actividades de DML que emite la lógica de DB2, como los programas de utilidad, los procedimientos SYSPROC o las sentencias de SQL internas, no se ven afectadas por este umbral. IMPORT, EXPORT y otros mandatos CLP se consideran como lógica del usuario; por lo tanto, las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales.

## Ejemplo

En el ejemplo siguiente se crea un umbral SQLROWSREAD TH1 para el dominio de la base de datos con un ámbito de implantación del miembro. Este umbral detiene la ejecución de cualquier actividad que lea más de 5.000.000 filas durante la evaluación de la consulta, que el umbral comprueba a intervalos de 10 segundos. Puede utilizar este umbral para asegurarse de que ninguna consulta del sistema lee un número de filas poco razonable, lo cual podría impactar en otros trabajos que se estén ejecutando en el sistema.

```
CREATE THRESHOLD TH1 FOR DATABASE ACTIVITIES
  ENFORCEMENT MEMBER
  WHEN SQLROWSREAD > 5000000 CHECKING EVERY 10 SECONDS
  STOP EXECUTION;
```

## Umbral SQLROWSREADINSC

El umbral de tipo en clase de servicio SQLROWSREADINSC especifica el número máximo de filas que puede leer una actividad de DML en un miembro de base de datos particular mientras se ejecuta en una subclase de servicio específica. Utilice este umbral para detectar y controlar actividades que lean un número excesivo de filas.

**Clase** Actividad

### Dominio de definición

Subclase de servicio

### Ámbito de implantación

Miembro

### Trabajo con seguimiento

Consulte la información que se encuentra más adelante en este tema

### Colocación en cola

No

### Unidad

Número de filas

### Predictivo o reactivo

Reactivo

Este umbral difiere del umbral SQLROWSREAD en que controla el número de filas leídas solo desde el momento en que una actividad entra en una subclase de servicio específica, no el número total de filas leídas durante la vida útil de la actividad. Este umbral también difiere del umbral SQLROWSRETURNED en que controla el número máximo de filas leídas durante la evaluación de una consulta en la subclase de servicio actual, no el número de filas devueltas a una aplicación de cliente desde el servidor de datos.

Los accesos al índice no cuentan para el número total de filas leídas. Si un plan de acceso solo utiliza índices durante la evaluación de la consulta, no se infringe el umbral SQLROWSREADINSC.

Este umbral se evalúa a intervalos de tiempo configurables por el usuario; si el intervalo es mayor que la cantidad de tiempo que tarda en sobrepasar el número de filas leídas, es posible que el número de filas leídas de una actividad de un miembro sobrepase el límite del umbral antes de que se detecte la infracción.

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Las actividades coordinadoras del tipo DML y el trabajo de subagente correspondiente, como la ejecución de una subsección.
- Actividades de DML anidadas derivadas de las aplicaciones de usuario. En consecuencia, las actividades de DML que emite la lógica de DB2, como los programas de utilidad, los procedimientos SYSPROC o las sentencias de SQL internas, no se ven afectadas por este umbral. IMPORT, EXPORT y otros mandatos CLP se consideran como lógica del usuario; por lo tanto, las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales.

Puede utilizar la acción REMAP ACTIVITY para controlar actividades volviendo a correlacionarlas con una subclase de servicio con diferentes asignaciones de recursos.

## Ejemplo

El ejemplo siguiente crea dos subclases de servicio, A1 y A2, bajo una superclase A, con un único umbral de tipo en clase de servicio SQLROWSREADINSC que vuelve a correlacionar las actividades entre las subclases después de haber leído 10.000 filas en la subclase de servicio A1 durante la evaluación de la consulta. Queda anotado un registro del supervisor de sucesos.

```
CREATE SERVICE CLASS A;  
CREATE SERVICE CLASS A1 UNDER A;  
CREATE SERVICE CLASS A2 UNDER A;  
  
CREATE THRESHOLD T1 FOR SERVICE CLASS A1 UNDER A  
  ACTIVITIES ENFORCEMENT MEMBER  
  WHEN SQLROWSREADINSC > 10000 REMAP ACTIVITY TO A2  
  LOG EVENT MONITOR RECORD;
```

## Umbral SQLROWSRETURNED

El umbral SQLROWSRETURNED especifica el número máximo de filas que puede devolver el servidor de datos al cliente.

**Tipo** Actividad

### Dominio de definición

Base de datos, superclase de servicio, subclase de servicio, acción de trabajo, carga de trabajo y sentencia

### Ámbito de implantación

Base de datos

### Trabajo con seguimiento

Consulte la información que se encuentra más adelante en este tema

### Colocación en cola

No

**Unidad**

Número de filas

**Predictivo o reactivo**

Reactivo

Cuando una sentencia CALL devuelve varios conjuntos de resultados, el umbral se aplica a cada conjunto de resultados por separado, no como un agregado al número total de filas que se devuelven a través de todos los conjuntos de resultados. Por ejemplo, si define el umbral para 20 filas y la sentencia CALL devuelve dos conjuntos de resultados que a su vez devuelven 15 filas y 19 filas respectivamente, el umbral no se activa.

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Actividades de DML que se emiten en el miembro coordinador.
- Actividades de DML anidadas que se invocan desde una aplicación de usuario. En consecuencia, las actividades de DML que emite el servidor de datos internamente, como las actividades de DML emitidas desde los programas de utilidad de DB2, procedimientos almacenados SYSPROC y SQL interno, no se ven afectadas por este umbral.

**Umbral SQLTEMPSPACE**

El umbral SQLTEMPSPACE especifica la cantidad máxima de espacio de tablas temporal del sistema que una actividad de DML puede consumir en cualquier miembro. Las actividades de DML suelen utilizar espacios de tablas temporales para realizar operaciones como clasificar y manipular los conjuntos de resultados intermedios.

**Tipo** Actividad

**Dominio de definición**

Base de datos, superclase de servicio, subclase de servicio, acción de trabajo, carga de trabajo y sentencia

**Ámbito de implantación**

Miembro

**Trabajo con seguimiento**

Consulte la información que se encuentra más adelante en este tema

**Colocación en cola**

No

**Unidad**

Cantidad de espacio de tablas temporal expresado en kilobytes (KB), megabytes (MB) o gigabytes (GB)

**Predictivo o reactivo**

Reactivo

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Actividades de DML que se emiten en el miembro coordinador.
- Actividades de DML anidadas derivadas de las aplicaciones de usuario. En consecuencia, las actividades de DML que emite la lógica de DB2 (como programas de utilidad, procedimientos SYSPROC o SQL interno) no se ven afectadas por este umbral.

El servidor de datos considera IMPORT, EXPORT y otros mandatos CLP como lógica de usuario. Las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales.

## Umbrales agregados

Un umbral agregado ejerce un control colectivo sobre los elementos de trabajo de una base de datos. El límite que defina mediante un umbral agregado funcionará como un total acumulado con el que contribuye cualquier trabajo del que haya hecho un seguimiento el umbral.

Cuando el trabajo del que se acaban de crear instancias hace que se infrinja el límite superior, se activa la acción correspondiente. El trabajo que hizo que se infringiera el límite superior será el único sobre el que actuará la acción activada.

### Colocación en cola de las actividades

Algunos umbrales incorporan una cola, lo que le permite establecer el número de actividades que pueden ejecutarse simultáneamente y colocar en cola todas las actividades adicionales una vez que se alcanza el límite de simultaneidad, así hasta superar el límite establecido para la cola.

Cuando el número de actividades sobrepasa el límite de simultaneidad correspondiente a un umbral de colocación en cola, las nuevas peticiones se colocan automáticamente en cola por orden de llegada hasta que la cola alcanza el tamaño especificado en el límite de la cola de la definición de umbral. Cuando la cola se llena, se alcanza el límite superior. La siguiente petición se considera que ha violado el umbral, y se le aplica la acción especificada para el umbral. Por ejemplo, una acción STOP EXECUTION provoca que se rechace el trabajo nuevo que llega.

También puede definir como ilimitado el límite superior de colocación en cola, en cuyo caso no existe límite superior para el tamaño de la cola. En esta situación, cuando a la cola se añade trabajo que acaba de llegar, el umbral no se considera violado, independientemente de la longitud que haya alcanzado la cola. Si define un límite fijo para el límite superior de la cola, y define CONTINUE como acción de violación del umbral, todo el trabajo nuevo que llega que sobrepase el límite fijo se considerará como que ha violado el umbral; se creará una violación de umbral, pero el nuevo trabajo se añadirá a la cola igualmente.

### Umbral AGGSQLTEMPSPACE

El umbral AGGSQLTEMPSPACE especifica la cantidad máxima del espacio de tablas temporal del sistema que puede utilizarse en total en todas las actividades de DML que se ejecutan simultáneamente en una subclase de servicio. Las actividades de DML suelen utilizar espacios de tablas temporales para realizar operaciones como clasificar y manipular los conjuntos de resultados intermedios.

**Clase** Agregado

**Dominio de definición**

Subclase de servicio

**Ámbito de implantación**

Miembro

**Trabajo con seguimiento**

Consulte la información que se encuentra más adelante en este tema

**Colocación en cola**

No

**Unidad**

Kilobytes, megabytes o gigabytes

**Predictivo o reactivo**

Reactivo

Las actividades de las que este umbral hace un seguimiento son las siguientes:

- Actividades de DML que se emiten en el miembro coordinador.
- Actividades de DML anidadas derivadas de las aplicaciones de usuario. En consecuencia, las actividades de DML que emite la lógica de DB2, como los programas de utilidad, los procedimientos SYSPROC o las sentencias de SQL internas, no se ven afectadas por este umbral. IMPORT, EXPORT y otros mandatos CLP se consideran como lógica del usuario; por lo tanto, las actividades que se invocan desde IMPORT, EXPORT y otros mandatos CLP están sujetos a los umbrales.

**Umbral CONCURRENTDBCOORDACTIVITIES**

El umbral CONCURRENTDBCOORDACTIVITIES especifica el número máximo de actividades coordinadoras reconocidas que pueden ejecutarse simultáneamente en el dominio de definición especificado y el ámbito de implantación.

El uso de este tipo de umbrales está especialmente indicado para aplicaciones que no ejecutan más de una actividad a la vez. Si una aplicación inicia más de una actividad a la vez, como al emitir una sentencia UPDATE de SQL mientras hay un cursor abierto, es posible que se produzcan algunos casos de cierta contención de colas en función del nivel de simultaneidad que permita el umbral y del comportamiento de las otras aplicaciones implicadas. Si este umbral se da en casos donde las aplicaciones pueden ejecutar más de una actividad a la vez, o el comportamiento de la aplicación es desconocido, se recomienda tener definido un umbral ACTIVITYTOTALTIME para esas actividades y contribuir así a resolver automáticamente cualquier caso de contención de colas potencial.

**Tipo** Agregado

**Dominio de definición**

Base de datos, acción de trabajo, superclase de servicio, subclase de servicio

**Ámbito de implantación**

Base de datos en entornos que no sean entornos DB2 pureScale

Miembro de un entorno DB2 pureScale

**Trabajo con seguimiento**

Coordinador y actividades anidadas reconocidos (vea "Identificación de trabajos por tipo, por coste o por medio de los datos a los que se accede con clases de trabajo" en la página 52)

**Colocación en cola**

Sí

**Unidad**

Número de actividades de base de datos simultáneas

**Predictivo o reactivo**

Predictivo

Este umbral es una generalización del umbral CONCURRENTWORKLOADACTIVITIES. El umbral CONCURRENTWORKLOADACTIVITIES solo se aplica a actividades que se

ejecutan en un dominio de carga de trabajo, pero puede aplicar el umbral `CONCURRENTDBCOORDACTIVITIES` a una gran variedad de dominios, desde una base de datos completa a una única acción de trabajo. El umbral `CONCURRENTDBCOORDACTIVITIES` hace un seguimiento de las actividades coordinadoras para sentencias que no son `CALL`, y de cualquier actividad anidada que generen las sentencias `CALL`. A diferencia del umbral `CONCURRENTWORKLOADACTIVITIES`, el umbral `CONCURRENTDBCOORDACTIVITIES` es un umbral de puesta en cola.

**Nota:** Si se especifica la acción de umbral `CONTINUE` para un umbral de cola, el tamaño de la cola no queda acotado, independientemente de los valores fijos que se hayan especificado para el tamaño de la cola.

Cuando se crean umbrales de cola del tipo `CONCURRENTDBCOORDACTIVITIES`, tenga cuidado con las configuraciones que pueden conducir a una contención irreconciliable basada en colas. Por ejemplo:

1. Se crea un umbral de simultaneidad del tipo `CONCURRENTDBCOORDACTIVITIES` con un valor máximo de simultaneidad de 1 y un tamaño de cola mayor que 1.
2. Una aplicación abre un cursor que el servidor de datos de DB2 reconoce como actividad A1, que consume el único ticket que está disponible para el umbral.
3. Mientras el cursor sigue abierto, la aplicación emite una sentencia `UPDATE`, que el servidor de datos reconoce como actividad A2, y que también está sujeta al umbral de simultaneidad. Puesto que la actividad A1 ya se está ejecutando, la nueva actividad A2 se pone en cola.

La aplicación está ahora en un estado de contención de colas que no puede resolverse. Está a la espera de que A2 se ejecute, pero A2 está a la espera de que A1 termine su ejecución. Esta situación no se resolverá sin la intervención externa a menos que también se haya definido un umbral `ACTIVITYTOTALTIME` en estas actividades. Si hay un umbral `ACTIVITYTOTALTIME` definido, la situación se resolverá cuando A1 infrinja dicho umbral al dedicar más tiempo del máximo que el servidor de datos debería tardar en procesar dicha actividad.

Este ejemplo puede generalizarse para varias aplicaciones y colas. Puede resolver esta situación incrementando los valores de simultaneidad, o cancelando determinadas actividades si los valores de simultaneidad están bien definidos.

Para reducir la ocasión de crear situaciones de contención de colas irreconciliables, el umbral `CONCURRENTDBCOORDACTIVITIES` afecta a diferentes tipos de actividades, como se indica a continuación:

- Las sentencias `CALL` no están controladas por el umbral, pero todas las actividades hijas anidadas están bajo el control del umbral. Tenga en cuenta tanto los bloques anónimos como las rutinas autónomas están clasificadas como sentencias `CALL`.
- Las funciones definidas por el usuario (UDF) están bajo el control del umbral, pero las actividades hijas anidadas en las UDF no están controladas. Si se llama a una rutina autónoma desde la función definida por un usuario, ni la rutina autónoma ni las actividades hijas de la rutina autónoma estarán bajo el control del umbral.
- Las acciones activadoras que invocan las sentencias `CALL` y las actividades hijas de estas sentencias `CALL` no están bajo el control del umbral. Tenga en cuenta que las propias sentencias `INSERT`, `UPDATE` y `DELETE` que tienen la facultad de desencadenar una activación, están bajo el control del umbral.

## **Umbral CONCURRENTWORKLOADACTIVITIES**

El umbral CONCURRENTWORKLOADACTIVITIES especifica el número máximo del coordinador y de las actividades anidadas que pueden ejecutarse simultáneamente en una aparición de una carga de trabajo.

**Tipo** Agregado

**Dominio de definición**  
Carga de trabajo

**Ámbito de implantación**  
Aparición de la carga de trabajo

**Trabajo con seguimiento**  
Coordinador y actividades anidadas reconocidos (vea “Actividades” en la página 15)

**Colocación en cola**  
No

**Unidad**  
Número de actividades de carga de trabajo simultáneas

**Predictivo o reactivo**  
Predictivo

Este umbral se aplica a una sola aparición de carga de trabajo. Si hay varias apariciones de una carga de trabajo ejecutándose a la vez, el umbral se aplica por separado a cada aparición de la carga de trabajo. Las actividades con seguimiento incluyen todas las actividades coordinadoras reconocidas y las actividades anidadas que se generan como resultado de la ejecución de la actividad coordinadora. Por ejemplo, a diferencia del umbral CONCURRENTDBCOORDACTIVITIES, si se llama a un procedimiento almacenado y dicho procedimiento ejecuta parte del SQL, tanto la sentencia CALL (que es la actividad coordinadora) como las sentencias de SQL ejecutadas mediante el procedimiento almacenado (que son las actividades anidadas) cuentan para el total del umbral.

COMMIT, ROLLBACK y ROLLBACK de las sentencias SAVEPOINT no se ven afectadas por este umbral.

### **Consideraciones sobre las actividades anidadas**

Las actividades anidadas de las que este umbral hace un seguimiento deben cumplir los criterios siguientes:

- Deben ser una actividad coordinadora reconocida. Las actividades coordinadoras anidadas que no sean tipos reconocidos tal como se describe en “Identificación de trabajos por tipo, por coste o por medio de los datos a los que se accede con clases de trabajo” en la página 52 no entran en consideración.
- Deben invocarse directamente desde la lógica del usuario, por ejemplo, un procedimiento almacenado escrito por el usuario que emita SQL, o desde el procedimiento almacenado SYSPROC.ADMIN\_CMD. Las actividades coordinadoras anidadas que se inician a partir de la invocación de una utilidad DB2 o de cualquier otro código en los esquemas SYSIBM, SYSFUN. o SYSPROC no cuentan para el límite superior especificado por este umbral.

## Ejemplo

En este ejemplo, el valor máximo del umbral `CONCURRENTWORKLOADACTIVITIES` se establece en 5. La lógica del usuario hace que se produzca la secuencia de operaciones siguiente en una aparición de carga de trabajo:

1. Se emite un mandato **load**: el número actual de actividades de carga de trabajo es 1.
  - El mandato **load** emite internamente algunas secuencias SQL. El número actual de actividades de carga de trabajo es 1. (El SQL generado por una utilidad no cuenta para el umbral `CONCURRENTWORKLOADACTIVITIES`.)
  - El mandato **load** finaliza. El número actual de las actividades de carga de trabajo es 0.
2. Llamar mediante `CALL` al procedimiento almacenado `SYSPROC.SP1`. El número actual de las actividades de carga de trabajo es 1.
  - El procedimiento almacenado `SYSPROC.SP1` genera parte del SQL. El número actual de actividades de carga de trabajo es 1. (El SQL generado por una utilidad no cuenta para el umbral `CONCURRENTWORKLOADACTIVITIES`.)
  - El procedimiento almacenado `SYSPROC.SP1` finaliza. El número actual de las actividades de carga de trabajo es 0.
3. Se abre un cursor `C1`. El número actual de las actividades de carga de trabajo es 1.
4. Se emite un mandato **runstats**. El número actual de las actividades de carga de trabajo es 1.
  - El mandato **runstats** genera parte del SQL. El número actual de las actividades de carga de trabajo es 1.
  - El mandato **runstats** finaliza. El número actual de las actividades de carga de trabajo es 1.
5. El cursor `C1` se cierra. El número actual de las actividades de carga de trabajo es 0.
6. Llamar mediante `CALL` al procedimiento almacenado `BOB.SP1`. El número actual de las actividades de carga de trabajo es 1.
  - El procedimiento almacenado `BOB.SP1` abre tres cursores. El número actual de las actividades de carga de trabajo es 4.
  - El procedimiento almacenado `BOB.SP1` llama al procedimiento almacenado `SYSPROC.SP2`. El número actual de las actividades de carga de trabajo es 5.
    - El procedimiento almacenado `SYSPROC.SP2` emite parte del SQL. El número actual de las actividades de carga de trabajo es 5.
    - El procedimiento almacenado `SYSPROC.SP2` finaliza. El número actual de las actividades de carga de trabajo es 4.
  - El procedimiento almacenado `BOB.SP1` llama al procedimiento almacenado `BOB.SP2`. El número actual de las actividades de carga de trabajo es 5.
    - El procedimiento almacenado `BOB.SP2` emite parte del SQL. En este punto, se activa el umbral.
    - El procedimiento almacenado `BOB.SP2` finaliza. El número actual de las actividades de carga de trabajo es 4.
  - El procedimiento almacenado `BOB.SP1` finaliza. El número actual de las actividades de carga de trabajo es 0.
7. Se abre un cursor `C2`. El número actual de las actividades de carga de trabajo es 1.

8. Llamar mediante CALL al procedimiento almacenado BOB.SP2. El número actual de las actividades de carga de trabajo es 2.

### **Umbral CONCURRENTWORKLOADOCCURRENCES**

El umbral CONCURRENTWORKLOADOCCURRENCES es un umbral agregado que especifica el número máximo de apariciones de carga de trabajo que pueden ejecutarse simultáneamente en el miembro coordinador.

**Tipo** Agregado

**Dominio de definición**  
Carga de trabajo

**Ámbito de implantación**  
Miembro

**Trabajo con seguimiento**  
Apariciones de carga de trabajo

**Colocación en cola**  
No

**Unidad**  
Número de apariciones simultáneas de la carga de trabajo

**Predictivo o reactivo**  
Predictivo

Cuando se inicia una aparición de una carga de trabajo, si el trabajo que genera se envía a miembros no coordinadores, el trabajo en dichos miembros no se tiene en cuenta para el total de umbrales simultáneos en el miembro coordinador. Por ejemplo, supongamos que se define el umbral CONCURRENTWORKLOADOCCURRENCES para permitir tan solo una aparición de la carga de trabajo A en un miembro. Luego supongamos que una aplicación se conecta al miembro 1, que da como resultado una aparición de la carga de trabajo A que se ha iniciado, y que esa carga de trabajo hace que el trabajo se envíe a los miembros 1, 2 y 3. En esta situación, el número total de instancias de la carga de trabajo A es de uno en el miembro 1, y de cero en los miembros 2 y 3. Por lo tanto, si otra aplicación se conecta al miembro 1 y otra instancia de la carga de trabajo A se inicia en el miembro 1, esa carga de trabajo se rechazará. No obstante, todavía podrán iniciarse nuevas apariciones de la carga de trabajo A en los miembros 2 y 3.

### **Umbral TOTALMEMBERCONNECTIONS**

El umbral TOTALMEMBERCONNECTIONS especifica el número máximo de conexiones de base de datos simultáneas en un miembro coordinador de una base de datos, es decir, este umbral controla el número máximo de clientes que pueden conectarse a la base de datos en cada uno de sus miembros de base de datos.

Este umbral no es de uso obligatorio para los usuarios con autorización DBADM y WLMADM.

**Tipo** Agregado

**Dominio de definición**  
Base de datos

**Ámbito de implantación**  
Miembro

**Trabajo con seguimiento**  
Conexiones

**Colocación en cola**

Sí (impuesto en 0)

**Unidad**

Número de conexiones simultáneas

**Predictivo o reactivo**

Predictivo

Por ejemplo, si establece el umbral TOTALMEMBERCONNECTIONS en 10 y la base de datos tiene cinco miembros, cada miembro puede tener un máximo de 10 clientes conectados a la vez, para un total de 50 conexiones de clientes en toda la base de datos.

El umbral TOTALMEMBERCONNECTIONS solamente controla conexiones de coordinador. Las conexiones realizadas por subagentes no cuentan para el umbral.

Este umbral es útil para aquellas situaciones en las que desea tener varias bases de datos en la misma instancia. Definir un umbral TOTALMEMBERCONNECTIONS en un miembro garantiza que las conexiones del cliente de una base de datos no puedan utilizar todas las conexiones disponibles en un miembro.

Asegúrese de definir el parámetro **max\_connections** de configuración del gestor de base de datos con un valor lo suficientemente alto como para dar soporte al número máximo de conexiones que espera en la base de datos. Si define un umbral TOTALMEMBERCONNECTIONS para una base de datos, deberá definir el parámetro **max\_connections** con un valor que sea como mínimo el valor del umbral. Si desea ejecutar varias bases de datos en la misma instancia, asegúrese de que define el parámetro **max\_connections** con un valor lo suficientemente alto como para dar soporte al número máximo de conexiones para todas las bases de datos. El servidor de datos no comprueba esta condición porque es imposible saber de antemano cuántas bases de datos estarán activas simultáneamente.

**Nota:** Aunque el tamaño de la cola suele imponerse en 0 para el umbral TOTALMEMBERCONNECTIONS, si se especifica una acción de umbral de CONTINUE para este umbral de puesta en cola, el tamaño de la cola será ilimitado, independientemente de los valores fijos especificados para el tamaño de cola.

**Umbral TOTALSCMEMBERCONNECTIONS**

El umbral TOTALSCMEMBERCONNECTIONS especifica el número máximo de conexiones de base de datos simultáneas de un miembro coordinador para una superclase de servicio.

**Tipo** Agregado

**Dominio de definición**

Superclase de servicio

**Ámbito de implantación**

Miembro

**Trabajo con seguimiento**

Conexiones

**Colocación en cola**

Sí

**Unidad**

Número de conexiones simultáneas en la clase de servicio

## **Predictivo o reactivo**

### Predictivo

Cuando se alcanza el umbral TOTALSCMEMBERCONNECTIONS en la clase de servicio, las conexiones de coordinador subsiguientes que unen la superclase de servicio se ponen en cola hasta que se alcanza el tamaño de cola especificado. Por omisión, el tamaño de la cola es cero, lo que significa que no puede ponerse en cola ninguna conexión. Si una conexión se une a la cola de un umbral TOTALSCMEMBERCONNECTIONS, se considera que la conexión está en estado *transitorio*.

**Nota:** Si se especifica la acción de umbral CONTINUE para un umbral de cola, el tamaño de la cola no queda acotado, independientemente de los valores fijos que se hayan especificado para el tamaño de la cola.

Las conexiones con seguimiento incluyen tanto conexiones de cliente nuevas como conexiones de cliente existentes que cambian a la clase de servicio desde otra clase de servicio. Las conexiones cambian de clases de servicio a través de la asociación con una definición de carga de trabajo diferente que se correlaciona con una clase de servicio diferente. La reevaluación de la carga de trabajo se produce únicamente en los límites de la transacción, por lo que las conexiones pueden cambiar de clase de servicio solo en dichos límites; sin embargo, puesto que los recursos que están asociados con los cursores WITH HOLD se mantienen a través de los límites de la transacción, las conexiones que tienen cursores WITH HOLD abiertos no pueden cambiar de superclases de servicio. Cuando el concentrador de conexiones está activado, las aplicaciones que se hayan conmutado dejan la clase de servicio. Cuando una aplicación se conmuta en una sentencia posterior, debe volver a unir la clase de servicio y, en consecuencia, pasar el umbral.

Cuando se alcanza el umbral del tamaño de la cola, se activa la acción del umbral. El umbral TOTALSCMEMBERCONNECTIONS solamente controla conexiones de coordinador. Las conexiones realizadas por subagentes no cuentan para el umbral.

Si establece un valor de umbral para TOTALMEMBERCONNECTIONS, elija un valor lo suficientemente alto como para acomodar el umbral que especifique para TOTALSCMEMBERCONNECTIONS. Por ejemplo, si define cinco superclases de servicio para una base de datos y cada una de ellas tiene un umbral TOTALSCMEMBERCONNECTIONS con valor 10, el valor del umbral TOTALMEMBERCONNECTIONS debe ser de al menos 50.

## **Umbrales de unidad de trabajo**

Un umbral de unidad de trabajo aplica controles a las unidades de trabajo individuales. Cuando el uso de recursos de una unidad de trabajo individual infringe el límite superior del umbral que le está haciendo el seguimiento, se activa la acción correspondiente y se aplica una vez tanto a la aplicación (en caso de que se trate de forzar una aplicación) como a la unidad de trabajo (en caso de que se trate de una retrotracción).

### **Umbral UOWTOTALTIME**

El umbral UOWTOTALTIME especifica el intervalo máximo de tiempo que una unidad de trabajo puede tardar en el motor de DB2.

**Tipo** Unidad de trabajo

#### **Dominio de definición**

Base de datos, carga de trabajo, superclase de servicio

**Ámbito de implantación**

Base de datos

**Trabajo con seguimiento**

Consulte la información más adelante en este tema.

**Colocación en cola**

No

**Unidad**

Duración etiquetada

**Predictivo o reactivo**

Reactivo

La acción STOP EXECUTION de un umbral UOWTOTALTIME retrotrae la unidad de trabajo. La acción FORCE APPLICATION fuerza la aplicación a la que pertenece la unidad de trabajo. Puede especificarse la opción COLLECT ACTIVITY DATA para este umbral, pero se pasa por alto.

Un umbral UOWTOTALTIME definido en el dominio de carga de trabajo anula temporalmente cualquier umbral UOWTOTALTIME definido en el dominio de la superclase de servicio. Un umbral UOWTOTALTIME definido en el dominio de la superclase de servicio anula temporalmente cualquier umbral UOWTOTALTIME definido en el dominio de la base de datos.

**Ejemplo**

El ejemplo siguiente crea un umbral FORCE10MINUTEUOW que emite un mandato FORCE APPLICATION cuando una unidad de trabajo ha estado ejecutándose durante más de 10 minutos.

```
CREATE THRESHOLD FORCE10MINUTEUOW FOR DATABASE ACTIVITIES
ENFORCEMENT DATABASE
WHEN UOWTOTALTIME > 10 MINUTES
FORCE APPLICATION;
```

---

**Disminución de la prioridad del trabajo en curso**

La disminución de la prioridad es una solución para la gestión de carga de trabajo mediante la cual la prioridad de las actividades en curso cambian automáticamente a lo largo del tiempo.

Cuanto más dure la ejecución de una actividad, menor será su prioridad y menos recursos recibirá. Puede utilizar la disminución de la prioridad para controlar actividades con largos periodos de ejecución, de modo que pueda mejorarse el rendimiento para actividades de ejecución más corta. La solución de disminución de la prioridad funciona cuando el control de recursos responde al movimiento de trabajo entre clases de servicio. Es decir, cuando el trabajo que ya se ha procesado cambia las clases de servicio, el movimiento queda reflejado en los recursos que recibe dicho trabajo (al igual que otros trabajos de la nueva clase de servicio). La mejor implementación de esta solución se consigue al utilizar los controles de CPU explícitos que proporciona el asignador de CPU o la integración con los productos de gestión de carga de trabajo del sistema operativo.

## Modificación de la prioridad de las actividades mediante una nueva correlación

Los recursos del sistema se asignan y controlan mediante clases de servicio. Con la disminución de la prioridad, la prioridad de una actividad puede modificarse si la actividad se mueve de una clase de servicio a otra. La prioridad aumenta si la nueva clase de servicio tiene más recursos, mientras que la prioridad descende si la nueva clase de servicio tiene menos recursos. Las actividades se mueven cuando se supera un umbral con una acción REMAP ACTIVITY en función del uso máximo por omisión de un recurso específico, como el tiempo de procesador o las filas leídas. Después de correlacionar una actividad con una clase de servicio, continúa ejecutándose con las nuevas restricciones de recursos en vigor.

Un enfoque sencillo que puede utilizar para ayudar a que las consultas cortas se ejecuten más rápidamente consiste en definir una serie de clases de servicio que sucesivamente reduzcan los niveles de prioridad de recursos y acciones de umbral que muevan las actividades entre las subclases del servicio. Con esta configuración puede reducir la prioridad de un trabajo de ejecución larga con el tiempo y tal vez mejorar los tiempos de respuesta para trabajos de ejecución más corta sin tener un conocimiento detallado de las actividades que se ejecutan en su servidor de datos.

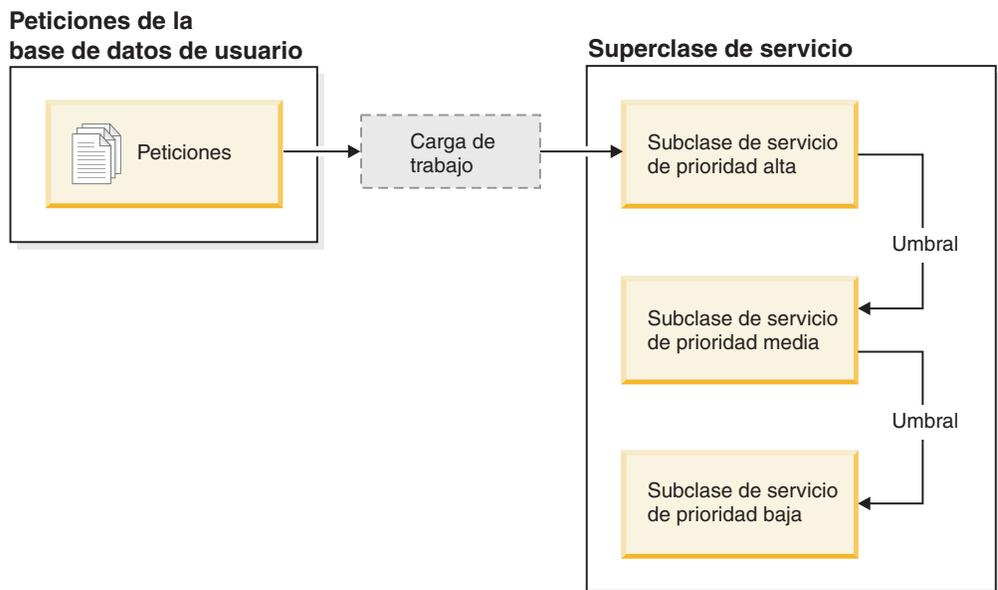


Figura 21. Configuración sencilla por niveles que muestra tres clases de servicio con una prioridad sucesivamente menor

Para crear esta configuración, puede asignar una prioridad alta a todos los recursos correspondientes a una clase de servicio, una prioridad media para una segunda clase de servicio y una prioridad baja para una tercera clase de servicio. A medida que entra trabajo en el sistema, se asigna automáticamente a la primera clase de servicio y empieza a ejecutarse utilizando los valores de prioridad alta de esa clase de servicio. Si además define umbrales para cada una de las clases de servicio que limitan el tiempo o los recursos utilizados durante la ejecución, el trabajo se reasigna dinámicamente a la clase de servicio inmediatamente inferior si se supera el umbral de la clase inmediatamente superior. Este control dinámico de los recursos se aplica repetidamente hasta que el trabajo se completa o se encuentra en la clase con la prioridad más baja, donde permanecerá hasta que se complete o se forzará su detención.

## Umbrales de tipo en clase de servicio

Es posible volver a correlacionar las actividades con cualquiera de los umbrales de tipo en clase de servicio, que controlan la cantidad de un recurso que puede utilizarse mientras una actividad se ejecuta en una determinada subclase de servicio. Ejemplos de recursos son la cantidad de tiempo de procesador utilizado (umbral CPUTIMEINSC) y el número de filas que lee una aplicación (umbral SQLROWSREADINSC) por actividad y miembro. Estos umbrales difieren de otros umbrales de actividad, que controlan los recursos utilizados a lo largo de la vida útil de una actividad.

Debido al control que proporcionan los umbrales de tipo en clase de servicio con respecto a las subclases de servicio, puede definir umbrales de tipo en clase de servicio sólo en un dominio de subclase de servicio. Los umbrales de tipo en clase de servicio proporcionan controles similares a las normas de DB2, que actúan sobre los elementos de supervisor de tiempo de procesador y filas leídas.

Cuando un umbral de tipo en clase de servicio se asocia a una acción REMAP ACTIVITY, los agentes que trabajan para la actividad comprueban periódicamente si se ha superado el umbral en cada miembro. Si un agente detecta una violación de umbral en un miembro, el agente activa la acción REMAP ACTIVITY para la actividad en el miembro y, a continuación, vuelve a correlacionarse con la subclase de servicio de destino. Los demás agentes que trabajan para la actividad en el mismo miembro se vuelven a correlacionar con la subclase de servicio de destino cuando detectan que la actividad se ha vuelto a correlacionar. Sólo un agente detecta la violación de umbral y vuelve a correlacionar la actividad, mientras que la actividad se considera nuevamente correlacionada después de que el agente haya detectado la violación de umbral y haya realizado la nueva correlación.

Dos elementos de supervisor proporcionan información sobre las nuevas correlaciones de actividades dentro de subclases de servicio. El elemento de supervisor **act\_remapped\_in** proporciona un contador que registra cuántas actividades se han vuelto a correlacionar en una subclase de servicio, y se incrementa para la subclase de servicio de destino cada vez que se vuelve a correlacionar una actividad. De modo parecido, el contador del elemento de supervisor **act\_remapped\_out** se incrementa para la subclase de servicio de origen cada vez que se anula la correlación de una actividad. Un elemento de supervisor adicional, **num\_remaps**, cuenta el número de veces en total que se ha vuelto a correlacionar una actividad entre subclases de servicio.

Una actividad puede volverse a correlacionar varias veces con diferentes subclases de servicio, mientras que una actividad puede regresar a su subclase de servicio original después de haberse correlacionado con otra subclase de servicio.

Los umbrales de clase de servicio se evalúan por separado para cada miembro en cada partición, sin coordinación. Puesto que no hay coordinación entre miembros, cuando una actividad se vuelve a correlacionar en una partición, es posible que la misma actividad esté en diferentes subclases de servicio en diferentes miembros al mismo tiempo.

Cuando se completa el trabajo del subagente para una actividad en un miembro remoto y más adelante se envía el trabajo posterior de la misma actividad al mismo miembro, la actividad se reinicia en la misma subclase de servicio que el agente que envió la petición al miembro. Si ha definido un umbral de tipo en clase de servicio para esta subclase de servicio, el temporizador o contador de la actividad en el miembro remoto se restablece en cero.

Cuando hay actividades anidadas, se realiza el seguimiento de las actividades principales y secundarias por separado. Por tanto, si una actividad secundaria utiliza una cantidad de recursos excesiva, sólo infringirá un umbral esta actividad, y no la actividad principal ni las homólogas.

## **Utilización de umbrales de tipo en clase de servicio**

En servidores de datos en los que las actividades de recursos primarias deben competir por el tiempo de procesador, utilice el umbral CPUTIMEINSC como primera medida de control. En servidores de datos en los que las consultas que leen muchas filas de tabla dan principalmente como resultado la contención de E/S, utilice SQLROWSREADINSC. En sistemas que experimentan una combinación de actividad intensa de procesador y E/S, utilice una combinación de los umbrales CPUTIMEINSC y SQLROWSREADINSC.

Debe establecer la prioridad de agente de las subclases de servicio en consonancia, de modo que su servidor de datos pueda tratar de forma diferente las actividades con diferentes prioridades empresariales. Tenga en cuenta que la prioridad de agente de la clase del sistema por omisión siempre debe ser mayor que cualquier clase de servicio definida por el usuario que cree para evitar un impacto negativo en el rendimiento. La prioridad de agente de la clase de mantenimiento por omisión puede establecerse en un nivel inferior que las clases de servicio definidas por el usuario.

La cantidad de un determinado recurso que permita consumir a las actividades de una subclase de servicio antes de volver a correlacionarlas con una subclase de servicio diferente dependerá en gran medida de su entorno en concreto. Para encontrar el mejor valor para cada condición de umbral, debe supervisar el modo en que las actividades se procesan en su servidor de datos. Si la cantidad máxima de tiempo de procesador que puede utilizarse o el número máximo de filas que puede leerse en una clase de servicio es demasiado alto, las actividades empezarán y acabarán inapropiadamente en la misma subclase de servicio, independientemente de la cantidad de recursos que requiera cada actividad. Si el valor de tiempo máximo de procesador o filas leídas es demasiado bajo, ninguna actividad acabará en la clase de servicio con la que estaba correlacionada originalmente, y todas las actividades se acabarán correlacionando nuevamente con otra clase de servicio, independientemente de su prioridad empresarial. En ambos casos, su configuración por niveles no se beneficiará del rendimiento global en su servidor de datos, y las actividades no se tratarán eficazmente de acuerdo con su prioridad empresarial.

Además de determinar la cantidad de un determinado recurso que puede consumir una actividad, algunos umbrales le permiten definir un intervalo de comprobación para establecer la frecuencia con la que el servidor de datos comprueba si existen violaciones de umbral. Esta prestación se proporciona para umbrales cuya comprobación resulta demasiado costosa cada vez que se consume una unidad del recurso bajo control y determina la latencia con la que se detectan infracciones de estos umbrales. Los umbrales CPUTIME y SQLROWSREAD y sus homólogos de clase de servicio CPUTIMEINSC y SQLROWSREADINSC admiten intervalos de comprobación. En instancias de base de datos en serie, el intervalo de comprobación equivale a la cantidad de tiempo real que desea que transcurra entre comprobaciones de una determinada violación de umbral. En entornos de bases de datos de varios miembros o en instancias SMP, el intervalo de comprobación debe establecerse en un valor que sea inferior a la cantidad de tiempo real transcurrido, con el fin de tener en cuenta el hecho de que puede haber más de un agente que acumule tiempo de procesador simultáneamente para esa actividad. Para calcular

el intervalo de comprobación aproximado en entornos de bases de datos de varios miembros o instancias SMP, divide la cantidad de tiempo real que desea que transcurra entre comprobaciones por el grado de paralelismo de la actividad y utilice el valor resultante para la cláusula CHECKING EVERY.

Por ejemplo: en una base de datos de un solo miembro, si desea que un umbral CPUTIMEINSC active una acción REMAP ACTIVITY después de 30 segundos de tiempo de procesador consumido, puede establecer el intervalo de comprobación en 30 segundos y tener la certeza de que la acción de umbral se activará no más tarde de que se hayan consumido 30 segundos de tiempo de procesador (el tiempo de procesador utilizado no puede superar el tiempo real transcurrido). En un entorno de base de datos de varios miembros, si define un umbral CPUTIMEINSC establecido en 5 segundos con un intervalo de comprobación de 5 segundos y una actividad tiene un agente de miembro coordinador y cuatro subagentes que trabajan en su nombre, es posible que la actividad consuma 5 segundos de tiempo de CPU en tan sólo 1 segundo de tiempo real, puesto que los cinco agentes acumulan simultáneamente 1 segundo de procesador cada uno. Para evitar que la actividad consuma un múltiplo de 5 segundos de tiempo de procesador, el intervalo de comprobación debe establecerse en este caso en 1 segundo.

Para obtener información adicional sobre cómo utilizar los umbrales, consulte los scripts de definición de niveles de ejemplo y los casos de ejemplo de disminución de la prioridad.

## **Efecto de una nueva correlación en los umbrales**

Los umbrales que sigan aplicándose después de una nueva correlación mediante una acción REMAP ACTIVITY dependerán de si los umbrales se aplican únicamente a una subclase de servicio específica o a toda la vida útil de una actividad.

Cuando vuelve a correlacionar una actividad con una nueva subclase de servicio, sólo cambian los umbrales de tipo en clase de servicio, como CPUTIMEINSC y SQLROWSREADINSC. Estos umbrales de tipo en clase de servicio ya no afectan a la actividad después de que esta abandone la subclase de servicio de origen, y se sustituyen por los umbrales correspondientes de la superclase de destino, si los ha definido. Todos los demás umbrales de actividad de la subclase de servicio con la que estaba correlacionada originalmente la actividad permanecen intactos, y los temporizadores y contadores de umbrales correspondientes no se restablecen. La actividad no se vuelve a evaluar en relación con otros umbrales que ha definido para la subclase de servicio de destino.

Por ejemplo, supongamos que dos subclases de servicio con umbrales se definen como sigue:

- Subclase de servicio A con los umbrales siguientes:
  - Un umbral de vida útil ACTIVITYTOTALTIME llamado TH1 y una acción STOP EXECUTION después de 30 minutos
  - Un umbral de tipo en clase de servicio SQLROWSREADINSC llamado TH2 con una acción REMAP ACTIVITY para la subclase B de servicio después de haber leído más de 2000 filas
- Subclase B de servicio con los umbrales siguientes:
  - Un umbral de vida útil ACTIVITYTOTALTIME llamado TH3 y una acción STOP EXECUTION después de 5 minutos

- Un umbral SQLROWSREADINSC llamado TH4 con una acción STOP EXECUTION después de haber leído más de 1000 filas

Cuando una actividad entra en el sistema en la subclase A de servicio, se aplican los dos umbrales TH1 y TH2 a la actividad. Si la actividad lee más de 2000 filas durante la evaluación de la consulta, volverá a correlacionarse dinámicamente a la subclase B de servicio. Debido a la correlación de la actividad a la subclase B, los umbrales de tipo en clase de servicio aplicables cambian, y ahora se aplica el umbral TH4 en lugar del umbral TH2 a la actividad. Los contadores para ambos umbrales se restablecen en cero, y aunque la actividad se haya leído durante más de 2000 filas en la subclase de servicio original, el contador de TH4 se restablecen en cero; la actividad debe leerse durante más de 1000 filas mientras se ejecuta en la subclase de servicio B antes de que se infrinja el umbral TH4. El umbral TH1, que se aplica durante toda la vida útil de la actividad, seguirá aplicándose aunque la actividad se ejecute ahora en una subclase diferente. El umbral TH3 no ejerce ningún control sobre la actividad que se ha vuelto a correlacionar, porque no se aplicó a la primera subclase de servicio en la que entró la actividad cuando empezó a ejecutarse.

## Scripts de ejemplo de disminución de la prioridad

Utilice los scripts de ejemplo que se proporcionan para crear rápidamente una configuración de clase de servicio por niveles en el servidor de datos. Con una configuración por niveles, puede dirigir los objetivos de rendimiento específicos disminuyendo la prioridad de las peticiones que tienen una duración más larga; este proceso se conoce como disminución de la prioridad. También puede modificar los scripts según sus propias prioridades empresariales a la hora de adaptarlos a su entorno.

Los dos scripts de muestra `wlmtiersdefault.db2` y `wlmtierstimerons.db2` pretenden demostrar cómo debe utilizar la disminución de la prioridad en el servidor de datos con el fin de mejorar el rendimiento general. El gestor de carga de trabajo de DB2 proporciona los controles que contribuirán al rendimiento general del servidor de datos, pero para obtener todas las ventajas de los scripts, y de la disminución de la prioridad en general, será necesario que ejecute su servidor de datos durante un período de tiempo prolongado y supervisar cómo se lleva a cabo este trabajo; luego deberá ajustar los valores del umbral y la clase de servicio en consecuencia.

Los scripts se proporcionan en el directorio `samples/admin_scripts` bajo el directorio de instalación.

Los scripts `wlmtiersdefault.db2` y `wlmtierstimerons.db2` crean tres subclases de servicio bajo una superclase común con unos valores de recursos que se van reduciendo progresivamente junto con los umbrales `CPUTIMEINSC` que trasladan o correlacionan de nuevo las actividades como respuesta al consumo del tiempo de procesador. Los scripts difieren en la forma como se correlacionan las actividades con una clase de servicio la primera vez que entran en el servidor de datos. Un tercer script, `wlmtiersdrop.db2`, descarta los objetos WLM creados por los otros dos scripts de ejemplo por niveles.

### **wlmtiersdefault.db2**

Todas las actividades que entran en el servidor de datos se correlacionan con una subclase de servicio de prioridad alta, `WLM_SHORT`, sin diferencias entre tipos de trabajo. Las actividades se completarán en la subclase de servicio de prioridad alta en tanto que se les asigna la prioridad más alta, a menos que sobrepasen la cantidad máxima de tiempo

de procesador que tienen permitido consumir. Las actividades que consumen demasiado tiempo de procesador vuelven a correlacionarse en primer lugar mediante una acción de umbral REMAP ACTIVITY con una subclase de servicio de prioridad media, WLM\_MEDIUM, y a continuación, si todavía no han finalizado sin sobrepasar el tiempo de procesador asignado en esa subclase, vuelven a correlacionarse con una subclase de servicio de prioridad baja, WLM\_LONG, donde siguen procesándose hasta que finalizan. Las actividades que no pueden volver a correlacionarse mediante un umbral CPUTIMEINSC, se correlacionan directamente a la subclase de servicio WLM\_MEDIUM, donde permanecerán.

#### **wlmtierstimerons.db2**

Las actividades de DML que entran en el servidor de datos se evalúan según su coste estimado, y se correlacionan con una de las tres subclases de servicio. Las actividades de DML que se prevén breves se correlacionan con una subclase de servicio de prioridad alta, WLM\_SHORT, las actividades de DML que se prevén de duración media se correlacionan con una subclase de servicio WLM\_MEDIUM que recibe la prioridad media, y las actividades de DML prolongadas se correlacionan con una subclase de servicio WLM\_LONG que recibe la prioridad más baja. Las actividades que no son de DML entran en la subclase de servicio de prioridad más alta. A medida que se procesan, las actividades que consumen más tiempo de procesador que el asignado a una subclase de servicio vuelven a correlacionarse sucesivamente mediante una acción de umbral REMAP ACTIVITY con la siguiente subclase de servicio de prioridad más baja, hasta que vuelvan a correlacionarse con la subclase de servicio de prioridad más baja, donde seguirán procesándose hasta que finalicen. Las actividades que no pueden volver a correlacionarse mediante un umbral CPUTIMEINSC, se correlacionan directamente a la subclase de servicio WLM\_MEDIUM, donde permanecerán.

#### **wlmtiersdrop.db2**

Este script descarta todas las clases de servicio del gestor de carga de trabajo de DB2, los umbrales, las cargas de trabajo, los conjuntos de clases de trabajo y los conjuntos de acciones de trabajo que se crean mediante los scripts `wlmtiersdefault.db2` y `wlmtierstimerons.db2`.

Por omisión, los scripts `wlmtiersdefault.db2` y `wlmtierstimerons.db2` utilizan las definiciones de umbral y de clase de servicio siguientes:

*Tabla 40. Clases de servicio con cuotas de CPU y valores de prioridad de captación previa creadas por los scripts*

Clase de servicio	Cuotas de CPU	Prioridad de captación previa
WLM_SHORT (prioridad alta)	6000	Alta
WLM_MEDIUM (prioridad media)	3000	Media
WLM_LONG (prioridad baja)	1000	Baja
Clase del sistema por omisión	Valor por omisión	Alta
Clase de mantenimiento por omisión	Valor por omisión	Baja

Tabla 41. Umbrales creados por los scripts

Umbral	Cantidad máxima de tiempo de procesador que puede utilizarse en la clase de servicio antes de volver a correlacionar
WLM_TIERS_REMAP_SHORT_TO_MEDIUM	30 segundos
WLM_TIERS_REMAP_MEDIUM_TO_LONG	30 segundos

El script de muestra `wlmtiersdefault.db2` crea el conjunto de acciones de trabajo y el conjunto de clases de trabajo siguientes, que se utilizan para correlacionar actividades que no pueden volver a correlacionarse mediante el umbral `CPUTIMEINSC` directamente con la subclase de servicio `WLM_MEDIUM`. Estas actividades permanecerán en la subclase de servicio `WLM_MEDIUM` mientras dure su ejecución.

Tabla 42. Conjunto de clases de trabajo creadas mediante el script de muestra `wlmtiersdefault.db2`

Clase de trabajo	Acción de trabajo
WLM_DML_WC	Para las actividades de DML, se correlacionan inicialmente con la clase de servicio <code>WLM_SHORT</code> . Estas actividades pueden volver a correlacionarse mediante un umbral <code>CPUTIMEINSC</code> .
WLM_CALL_WC	Para las actividades <code>CALL</code> , se correlacionan inicialmente con la clase de servicio <code>WLM_SHORT</code> . Estas actividades pueden volver a correlacionarse mediante un umbral <code>CPUTIMEINSC</code> .
WLM_OTHER_WC	Para actividades que no puedan volver a correlacionarse mediante un umbral <code>CPUTIMEINSC</code> , se correlacionan con la clase de servicio <code>WLM_MEDIUM</code> . Estas actividades permanecerán en la subclase de servicio <code>WLM_MEDIUM</code> .

El script de muestra `wlmtierstimerons.db2` también crea el conjunto de acciones de trabajo y el conjunto de clases de trabajo siguientes, que se utiliza para correlacionar actividades según su coste estimado:

Tabla 43. Conjunto de clases de trabajo creadas mediante el script de muestra `wlmtierstimerons.db2`

Clase de trabajo	Rango de coste estimado en timerons (activaciones de temporizador) y acciones de trabajo
WLM_SHORT_DML_WC	Para actividades de DML con un coste estimado de 0 a 999 timerons, se correlacionan inicialmente con la clase de servicio <code>WLM_SHORT</code> . Estas actividades pueden volver a correlacionarse mediante un umbral <code>CPUTIMEINSC</code> .

Tabla 43. Conjunto de clases de trabajo creadas mediante el script de muestra `wlmtierstimerons.db2` (continuación)

Clase de trabajo	Rango de coste estimado en timerons (activaciones de temporizador) y acciones de trabajo
WLM_MEDIUM_DML_WC	Para actividades de DML con un coste estimado de 1.000 a 99.999 timerons, se correlacionan inicialmente con la clase de servicio WLM_MEDIUM. Estas actividades pueden volver a correlacionarse mediante un umbral CPUTIMEINSC.
WLM_LONG_DML_WC	Para actividades de DML con un coste estimado de 100.000 a infinitos timerons, se correlacionan inicialmente con la clase de servicio WLM_LONG.
WLM_CALL_WC	Para las actividades CALL, se correlacionan inicialmente con la clase de servicio WLM_SHORT. Estas actividades pueden volver a correlacionarse mediante un umbral CPUTIMEINSC.
WLM_OTHER_WC	Para actividades que no puedan volver a correlacionarse, se correlacionan con la clase de servicio WLM_MEDIUM.

## Modificación de los scripts para su entorno

Cuando modifica los scripts de muestra para adaptarlos a su entorno, el valor más importante que debe considerarse es la cantidad máxima de tiempo de procesador que puede utilizarse en cada clase de servicio. El tiempo de procesador que permita consumir a las actividades en cada subclase de servicio dependerá principalmente de su entorno particular. Para averiguar cuáles son los mejores valores, deberá supervisar cómo se procesan las actividades en su servidor de datos. Por omisión, los scripts `wlmtiersdefault.db2` y `wlmtierstimerons.db2` anotarán cronológicamente los registros de supervisión de sucesos en el supervisor de sucesos de violaciones de umbral, si está activo, con la opción de activar y habilitar el supervisor de sucesos de actividades y recopilar los datos de actividad (a cambio de una sobrecarga adicional de recursos). Para `wlmtiersdefault.db2`, si la cantidad máxima de tiempo de procesador que puede utilizarse en cada clase de servicio se establece con un valor demasiado alto, la mayoría de las actividades siempre se iniciarán y terminarán en la clase de prioridad alta, independientemente de la cantidad de tiempo de procesador real requiera cada una. Si la cantidad máxima de tiempo de procesador se establece con un valor demasiado bajo, no terminará ninguna actividad en la clase de servicio de prioridad alta, y todas las actividades terminarán correlacionándose de nuevo con una clase de servicio de prioridad media o baja independientemente de cuál sea la prioridad empresarial. En cualquiera de los casos, el script no mejorará el rendimiento general del servidor de datos y las actividades no se considerarán según su prioridad empresarial efectiva. Ocurre lo mismo, a menor escala, con `wlmtierstimerons.db2`, donde las actividades se diferencian inicialmente por correlacionarse con subclases de servicio según el coste estimado. Si la cantidad máxima de tiempo de procesador que puede utilizarse en cada clase de servicio se establece de manera incorrecta, las actividades no podrán volver a correlacionarse con una subclase de servicio más apropiada si consumen demasiado tiempo de procesador, o se correlacionarán demasiado deprisa a pesar de tener la prioridad empresarial más alta.

Si desea más información sobre los objetos del gestor de carga de trabajo de DB2 específicos creados por los scripts y sobre cómo ejecutarlos, consulte los scripts.

## Casos de ejemplo

Se han incluido dos ejemplos en la documentación, donde se muestra cómo puede adaptar los scripts de ejemplo por niveles en el servidor de datos para poder utilizar la disminución de la prioridad.

### **Caso de ejemplo: Control de informes de inteligencia empresarial de uso intensivo de recursos con disminución de la prioridad**

El siguiente caso de ejemplo muestra cómo puede configurar el servidor de datos para reducir dinámicamente la prioridad de los informes de inteligencia empresarial que no pueden identificarse antes de que se inicie la ejecución con el fin de mantener el rendimiento del sistema para las otras consultas.

El problema: existe un informe de inteligencia empresarial que cualquier usuario final puede ejecutar y que utiliza muchos recursos. Cada vez que se ejecuta el informe, éste compromete el rendimiento del sistema. La herramienta de componente frontal utilizada para generar el informe no establece ninguna información de cliente que pueda utilizarse para identificar el informe por adelantado que le permita correlacionar éste con una clase de servicio de prioridad baja utilizando una carga de trabajo.

La solución: puede utilizar los scripts de definición de niveles de ejemplo `wlmtiersdefault.db2` para configurar el servidor de datos con una configuración por niveles que reduzca o disminuya dinámicamente la prioridad de las actividades que hacen un uso intensivo del procesador durante su vida útil con el fin de impedir que se comprometa el rendimiento del servidor de datos para todos los demás usuarios. Después de que una carga de trabajo ha correlacionado inicialmente todo el trabajo con una subclase de servicio de prioridad alta, el umbral de tipo en clase de servicio `CPUTIMEINSC` detecta los informes que utilizan muchos recursos basándose en la cantidad de tiempo de procesador consumido. Si una actividad infringe el umbral `CPUTIMEINSC` por utilizar la cantidad máxima del tiempo de procesador permitido, una `REMAP ACTIVITY` mueve la actividad a una subclase de servicio de prioridad más baja. La actividad puede volver a correlacionarse como respuesta al consumo de tiempo de procesador nuevamente hasta que se ejecute en la subclase de servicio de prioridad más baja, donde continuará hasta que se complete o hasta que intervenga manualmente. Las otras actividades que no exceden los umbrales continúan ejecutándose en la subclase de servicio de prioridad alta, donde reciben prioridad de agente más alta.

Cada vez que vuelve a correlacionarse una actividad se graba un registro de supervisor de sucesos, si ha creado un supervisor de sucesos de violaciones de umbral. Si desea recopilar información adicional acerca de las actividades que se han correlacionado para continuar con la investigación, puede añadir la cláusula `COLLECT ACTIVITY DATA` a la sentencia `ALTER THRESHOLD` en el script `wlmtiersdefault.db2`. Sólo ha de volver a ejecutar el script para que el cambio entre en vigor.

Tras haber ejecutado la carga de trabajo durante un período de tiempo, puede utilizar la función de tabla `WLM_GET_SERVICE_SUBCLASS_STATS` para ver cuántas actividades han vuelto a correlacionarse entre las subclases de servicio:

```

SELECT substr(service_superclass_name,1,21) AS superclass,
       substr(service_subclass_name,1,21) AS subclass,
       substr(char(coord_act_completed total),1,10) AS completed,
       substr(char(act_remapped_in),1,10) AS remapped_in,
       substr(char(act_remapped_out),1,10) AS remapped_out,
       substr(char(last_reset),1,19) AS last_reset
FROM table( WLM_GET_SERVICE_SUBCLASS_STATS(
           CAST(NULL AS VARCHAR(128)),
           CAST(NULL AS VARCHAR(128)),
           -2 )
          ) AS TF_subcls_stats@

SELECT SUBSTR(WORKLOAD_NAME,1,19) AS WL_NAME,
       COORD_ACT_LIFETIME_AVG,
       COORD_ACT_LIFETIME_STDDEV
FROM TABLE(WLM_GET_WORKLOAD_STATS
           (CAST(NULL AS VARCHAR(128)), -2))
       AS WLSTATS
ORDER BY WL_NAME@

```

SUPERCLASS	SUBCLASS	COMPLETED	REMAPPED_IN	REMAPPED_OUT	LAST_RESET
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0	0	0	2008-10-06-20.53.47
SYSDEFAULTMAINTENANCE	SYSDEFAULTSUBCLASS	3	0	0	2008-10-06-20.53.47
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0	0	2008-10-06-20.53.47
WLM_TIERS	SYSDEFAULTSUBCLASS	0	0	0	2008-10-06-20.53.47
WLM_TIERS	WLM_SHORT	999	0	35	2008-10-06-20.53.47
WLM_TIERS	WLM_MEDIUM	19	35	16	2008-10-06-20.53.47
WLM_TIERS	WLM_LONG	16	16	0	2008-10-06-20.53.47

7 registro(s) seleccionado(s).

Si observa que ninguna o muy pocas actividades han vuelto a correlacionarse con las subclases de servicio de prioridad más baja, reduzca el valor del umbral CPUTIMEINSC y compruebe el intervalo que las sentencias ALTER THRESHOLD utilizan en el script para mejorar la correlación de las actividades en los niveles de clase de servicio de acuerdo con la prioridad empresarial. Si la mayoría o casi todas las actividades se han correlacionado con las subclases de servicio de prioridad más baja, incremente el valor del umbral CPUTIMEINSC y compruebe el intervalo para las sentencias ALTER THRESHOLD para permitir que más actividades puedan completarse con una prioridad más alta. Tras haber completado los cambios, vuelva a ejecutar el script wlmtiersdefault.db2 para que éstos entren en vigor.

### Caso de ejemplo: Nueva correlación de consultas correlacionadas incorrectamente mediante disminución de la prioridad

El siguiente caso de ejemplo muestra cómo puede configurar el servidor de datos para volver a correlacionar o disminuir dinámicamente la prioridad de las actividades que consumen más tiempo de procesador que el estimado originalmente con el fin de mantener el rendimiento del sistema para las otras consultas.

El problema: puede que haya correlacionado actividades que hacen un uso intensivo de recursos basándose en el coste estimado de SQL con una subclase de servicio de prioridad más baja para que estas actividades no influyan negativamente en el rendimiento de las actividades más cortas, que hacen un menor uso de los recursos. Esta correlación puede obtenerse definiendo un conjunto de acciones de trabajo en el nivel de la superclase de servicio. Sin embargo, si el coste estimado de SQL no es correcto porque las estadísticas no están actualizadas, por ejemplo, una actividad de uso intensivo de recursos podría correlacionarse incorrectamente con una subclase de servicio de prioridad alta en la que empieza a consumir una cantidad excesiva de recursos, a expensas de todas las demás actividades de prioridad alta.

La solución: puede utilizar el script de definición de niveles de ejemplo wlm-tierstimerons.db2 para configurar el servidor de datos con una configuración por niveles que evalúe las actividades de entrada de acuerdo con su coste

estimado y las correlacione con una de las tres subclases de servicio, donde cada una de éstas tendrán prioridades de agente distintas. Si una actividad consume demasiado tiempo de procesador, el servidor de datos reducirá dinámicamente la prioridad de la actividad durante su vida útil volviendo a correlacionar ésta entre niveles de rendimiento. Este proceso dinámico de nueva correlación de las actividades para reducir su prioridad también se denomina disminución de la prioridad.

Después de que una actividad se ha correlacionado con su clase de servicio inicial e inicia su ejecución, el script utiliza el umbral de tipo en clase de servicio CPUTIMEINSC para controlar la cantidad de tiempo de procesador que una actividad puede consumir. Si la actividad infringe el umbral por utilizar la cantidad máxima del tiempo de procesador permitido, se activa una acción REMAP ACTIVITY que mueve la actividad a una subclase de servicio con una prioridad de agente más baja. La actividad puede volver a correlacionarse como respuesta al consumo de tiempo de procesador hasta que se ejecute en la subclase de servicio de prioridad más baja, donde continuará hasta que se complete o hasta que intervenga manualmente.

Cada vez que vuelve a correlacionarse una actividad se graba un registro de supervisor de sucesos. Si desea recopilar información adicional acerca de las actividades que se han correlacionado para continuar con la investigación, puede añadir la cláusula COLLECT ACTIVITY DATA a la sentencia ALTER THRESHOLD en el script wlmtiersdefault.db2. Sólo ha de volver a ejecutar el script para que el cambio entre en vigor.

Tras haber ejecutado la carga de trabajo durante un período de tiempo, puede utilizar la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS para ver cuántas actividades han vuelto a correlacionarse entre las subclases de servicio:

```
SELECT substr(service_superclass_name,1,21) AS superclass,
       substr(service_subclass_name,1,21) AS subclass,
       substr(char(coord_act_completed_total),1,10) AS completed,
       substr(char(act_remapped_in),1,10) AS remapped_in,
       substr(char(act_remapped_out),1,10) AS remapped_out,
       substr(char(last_reset),1,19) AS last_reset
FROM table( WLM_GET_SERVICE_SUBCLASS_STATS(
           CAST(NULL AS VARCHAR(128)),
           CAST(NULL AS VARCHAR(128)),
           -2 )
          ) AS TF_subcls_stats@
```

```
SELECT SUBSTR(WORKLOAD_NAME,1,19) AS WL_NAME,
       COORD_ACT_LIFETIME_AVG,
       COORD_ACT_LIFETIME_STDDEV
FROM TABLE(WLM_GET_WORKLOAD_STATS
           (CAST(NULL AS VARCHAR(128)), -2))
       AS WLSTATS
ORDER BY WL_NAME@
```

SUPERCLASS	SUBCLASS	COMPLETED	REMAPPED_IN	REMAPPED_OUT	LAST_RESET
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0	0	0	2008-10-06-20.59.27
SYSDEFAULTMAINTENANCE	SYSDEFAULTSUBCLASS	3	0	0	2008-10-06-20.59.27
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0	0	2008-10-06-20.59.27
WLM_TIERS	SYSDEFAULTSUBCLASS	0	0	0	2008-10-06-20.59.27
WLM_TIERS	WLM_SHORT	651	0	5	2008-10-06-20.59.27
WLM_TIERS	WLM_MEDIUM	36	5	7	2008-10-06-20.59.27
WLM_TIERS	WLM_LONG	16	7	0	2008-10-06-20.59.27

7 registro(s) seleccionado(s).

Para este caso de ejemplo, debería ver que son relativamente pocas las actividades que vuelve a correlacionarse entre subclases de servicio, pues las actividades inicialmente han de correlacionarse casi siempre con la subclase de servicio

adecuada, en función del coste estimado. Si observa que las actividades suelen completarse sólo en la clase de servicio WLM\_SHORT o WLM\_LONG, puede ajustar los valores de coste estimado que la sentencia ALTER WORK CLASS SET ha utilizado en el script para mejorar la correlación de las actividades en los niveles de clase de servicio con el fin de que las actividades más cortas se correlacionen con la clase de trabajo WLM\_SHORT\_DML\_WC y las actividades más largas se correlacionen con la clase de trabajo WLM\_MEDIUM\_DML\_WC o WLM\_LONG\_DML\_WC. Si observa que la mayoría de las actividades vuelven a correlacionarse, puede incrementar los valores de umbral que se han utilizando en las sentencias ALTER THRESHOLD para mejorar la correlación inicial de las actividades con las subclases de servicio. Tras haber completado los cambios, vuelva a ejecutar el script wlmstimerons.db2 para que éstos entren en vigor.

## **Nueva correlación de actividades entre subclases de servicio**

Puede habilitar una nueva correlación especificando una acción REMAP ACTIVITY con las sentencias CREATE y ALTER THRESHOLD. La acción de nueva correlación, cuando la desencadena una violación de umbral, mueve una actividad de una subclase de servicio a otra subclase de servicio bajo la misma superclase.

### **Antes de empezar**

A fin de poder volver a correlacionarse con otra subclase de servicio, la subclase de servicio destino debe existir bajo la misma superclase de servicio que la subclase de servicio original de la actividad. La subclase de servicio de destino u original puede ser la subclase por omisión de la superclase. La acción REMAP ACTIVITY no puede aplicarse a subclases de servicio bajo la clase de sistema por omisión, la clase de mantenimiento por omisión o la clase de usuario por omisión.

### **Acerca de esta tarea**

La acción REMAP ACTIVITY moverá una actividad a una subclase de servicio diferente dentro de la misma superclase de servicio. La nueva correlación está disponible con cualquiera de los umbrales de tipo en clase de servicio, como CPUTIMEINSC y SQLROWSREADINSC. El proceso dinámico de volver a correlacionar las actividades se utiliza para reducir la prioridad a lo largo del tiempo, algo que también se denomina disminución de la prioridad. Disminuir la prioridad de algunas actividades a lo largo del tiempo puede liberar recursos del sistema, que entonces pueden aplicarse a otras actividades de mayor importancia para la empresa.

Puede utilizar la acción REMAP ACTIVITY con el umbral de tipo en clase de servicio DATATAGINSC que correlaciona una actividad con una subclase de servicio de DB2 distinta en función de los datos a los que se estima que se accederá.

Los agentes que trabajan para la actividad comprobarán periódicamente si se ha infringido un umbral en cada miembro, sin coordinación entre miembros. Cuando un agente detecte una violación de umbral de tipo en clase de servicio en un miembro, este agente desencadenará la acción REMAP ACTIVITY para la actividad del miembro y, a continuación, se volverá a correlacionar con la subclase de servicio de destino, tras lo cual se considerará que la actividad se ha vuelto a correlacionar. Todos los demás agentes que trabajan para la actividad en el mismo miembro se correlacionarán con la subclase de servicio de destino cuando detecten que se ha correlacionado la actividad.

## Restricciones

La subclase de servicio de destino no puede ser la misma que la subclase de servicio original; debe volver a correlacionarla con otra subclase de servicio diferente antes de volver a correlacionarla con la original.

Si una actividad se vuelve a correlacionar con una subclase de servicio inhabilitada, la actividad se tratará como si la subclase inhabilitada la hubiera rechazado, y se devolverá al cliente el mensaje de error -4714.

## Procedimiento

1. Identifique aquellas actividades que desea controlar mediante la disminución de la prioridad. Una configuración por niveles se compone de subclases de servicio bajo la misma superclase de servicio que tienen umbrales de tipo en clase de servicio con acciones de umbral REMAP ACTIVITY definidas. Como punto de partida, puede consultar el ejemplo básico de disminución de la prioridad y los casos de ejemplo:
  - a. "Disminución de la prioridad del trabajo en curso" en la página 161
  - b. "Caso de ejemplo: Control de informes de inteligencia empresarial de uso intensivo de recursos con disminución de la prioridad" en la página 170
  - c. "Caso de ejemplo: Nueva correlación de consultas correlacionadas incorrectamente mediante disminución de la prioridad" en la página 171
2. Seleccione las diferentes subclases de servicio con las que se correlacionarán las actividades. Esto incluye tanto la subclase de servicio con la que la actividad está correlacionada originalmente al iniciarse la ejecución de la actividad, como cualquier otra subclase de servicio o clases con las que se correlacione la actividad durante su vida útil. Para obtener más información sobre clases de servicio, incluido cómo crearlas, consulte "Asignación de recursos con clases de servicio" en la página 73.
3. Cree o modifique los umbrales para controlar las actividades. Para obtener más información sobre umbrales, consulte "Disminución de la prioridad del trabajo en curso" en la página 161.
  - a. Defina sus umbrales de tipo en clase de servicio de modo que incluyan una acción REMAP ACTIVITY, que se desencadenará cuando se infrinja el umbral. Tenga en cuenta que un umbral de tipo en clase de servicio se aplica a una actividad y se ve afectada por ésta sólo mientras la actividad está correlacionada con la subclase de servicio asociada; los contadores y temporizadores afectados se restablecen después de la nueva correlación. Plántese si debe anotarse un registro de violación de umbral cada vez que se vuelve a correlacionar una actividad. Estos registros proporcionan información sobre las clases de servicio en las que la actividad ha pasado tiempo durante su ejecución y puede utilizarlos para el análisis del rendimiento. Tenga en cuenta que la anotación cronológica de registros de violación de umbral puede empezar a consumir cantidades significativas de espacio de disco si se producen con frecuencia nuevas correlaciones de actividades entre subclases de servicio.
  - b. También puede definir los umbrales que desee aplicar en toda la vida útil de la actividad, pero tenga en cuenta que sólo los umbrales de la primera subclase de servicio con la que la actividad estaba correlacionada originalmente se seguirán aplicando durante toda la vida útil de la actividad. Si también define umbrales en alguna de las subclases de servicio con las que una actividad se correlacionará posteriormente, no se aplicarán.
4. Confirme los cambios. Cuando confirme los cambios, los umbrales se añadirán a la vista SYSCAT.THRESHOLDS.

5. Permita que el servidor de datos ejecute las actividades que tiene por objeto con sus umbrales de tipo en clase de servicio y supervise su progreso durante su vida útil. Las actividades permanecerán en su subclase de servicio original durante la ejecución siempre que no infrinjan un umbral de tipo en clase de servicio. A medida que se infrinjan umbrales de tipo en clase de servicio durante la ejecución de las actividades, las actividades desencadenarán una acción REMAP ACTIVITY, que volverá a correlacionarlas dinámicamente con subclases de servicio diferentes. Una vez correlacionadas de nuevo, las actividades continuarán su ejecución y estarán controladas ahora por las restricciones de recursos que haya implantado en la subclase de servicio de destino.
6. Si es necesario, defina con más detalle su enfoque de la disminución de la prioridad para alcanzar sus objetivos de rendimiento propuestos.

## Ejemplo

En el ejemplo siguiente se crea una configuración sencilla de tres niveles que disminuye la prioridad de las actividades en curso a lo largo del tiempo. Tres subclases de servicio bajo la misma superclase A proporcionan el entorno de ejecución en el que deben ejecutarse todas las consultas. Supongamos que una carga de trabajo de usuario por omisión correlaciona las consultas entrantes con la subclase de servicio A1, que es una subclase de prioridad alta que tiene por objeto permitir que las consultas de ejecución más corta se ejecuten rápidamente. Una subclase de servicio de prioridad media A2 tiene como finalidad permitir que las consultas de ejecución más larga se ejecuten, si bien con controles de recursos más estrictos. La subclase de servicio A3 proporciona contención para consultas muy grandes que requieren una cantidad de tiempo de procesador excesiva para completarse.

Tres umbrales proporcionan control sobre el consumo de recursos de las consultas en curso. Se permite que una consulta se ejecute en la subclase de servicio A1 siempre que requiera menos de un minuto de tiempo de procesador para completarse. Después de consumir un minuto de tiempo de procesador, el umbral T1 vuelve a correlacionar automáticamente la actividad con la subclase A2, donde puede continuar ejecutándose siempre que consuma menos de diez minutos de tiempo de procesador. Si la consulta todavía no se ha completado al cabo de diez minutos de consumir tiempo de procesador, el umbral T2 vuelve a correlacionar la actividad con la subclase de servicio de prioridad más baja, A3. Se permite que las consultas de la subclase A3 continúen indefinidamente, aunque se anota un registro del supervisor de sucesos y se recopilan los datos de la actividad con detalles cuando el tiempo de procesador supera la hora.

```
CREATE SERVICE CLASS A
CREATE SERVICE CLASS A1 UNDER A
CREATE SERVICE CLASS A2 UNDER A
CREATE SERVICE CLASS A3 UNDER A

CREATE THRESHOLD T1 FOR SERVICE CLASS A1 UNDER A
ACTIVITIES ENFORCEMENT MEMBER
WHEN CPUTIMEINSC > 1 MINUTE REMAP ACTIVITY TO A2

CREATE THRESHOLD T2 FOR SERVICE CLASS A2 UNDER A
ACTIVITIES ENFORCEMENT MEMBER
WHEN CPUTIMEINSC > 10 MINUTES REMAP ACTIVITY TO A3

CREATE THRESHOLD T3 FOR SERVICE CLASS A3 UNDER A
```

```
ACTIVITIES ENFORCEMENT MEMBER
WHEN CPUTIMEINSC > 1 HOUR LOG EVENT MONITOR RECORD
COLLECT ACTIVITY DATA WITH DETAILS
CONTINUE
```

---

## Visión general del asignador de gestión de carga de trabajo

El asignador de gestión de carga de trabajo de DB2 es una tecnología de DB2 incorporada mediante la cual se pueden asignar específicamente recursos de CPU a trabajo que se está ejecutando en un servidor de bases de datos. Las titularidades de recursos de CPU se pueden controlar mediante atributos de cuotas de CPU y atributos de límite de CPU en clases de servicio de DB2.

### Visión general

El asignador, de particular utilidad en condiciones de cargas de trabajo que hacen un uso intensivo de los recursos de CPU del sistema, puede gestionar con efectividad las titularidades de recursos de CPU de clase de servicio con atributos como las cuotas de CPU flexibles sin límite (prácticamente sin restricciones), y con los dos atributos con límite (con restricciones): las cuotas de CPU fijas y los límites de CPU. Puede utilizar las cuotas de CPU flexibles sin límite para otorgar cualquier recurso de CPU no utilizado a las clases de servicio relacionadas con los trabajos de alta prioridad. Asimismo, puede aplicar controles para las titularidades de recursos de CPU mediante la utilización de las cuotas de CPU fijas con límite o los límites de CPU que se han asignado a las clases de servicio relacionadas con trabajos de baja prioridad, limitando de esta forma el impacto de éstos en el trabajo de alta prioridad. En aquellas condiciones en las que el uso de los recursos de CPU del sistema sea menos intensivo, puede utilizar uno o los dos atributos con límite, sin que sea necesario asignar atributos con límite, para poder controlar de forma eficiente las titularidades de recursos de CPU entre clases de servicio que se ejecutan en un entorno de baja utilización de CPU típico.

La infraestructura del asignador de gestión de carga de trabajo opera en el ámbito de la instancia, en el gestor de bases de datos DB2. El funcionamiento del asignador consiste en limitar el número de agentes en ejecución que han de asignarse al sistema operativo (SO) y durante cuánto tiempo está permitido que se ejecute cada agente, en cualquier momento. El número de agentes en ejecución que pueden asignarse al mismo tiempo se denomina nivel de simultaneidad de asignación.

El asignador de gestión de carga de trabajo de DB2 ofrece las ventajas siguientes:

- Es fácil de implementar, pues requiere menos tiempo y esfuerzo que los que se necesitan para implementar un WLM de SO como AIX WLM o Linux WLM.
- Admite una asignación de CPU flexible para los altibajos diarios habituales del uso del sistema. Esta flexibilidad se consigue mediante el suministro tanto de asignaciones permanentes que se aplican siempre (cuotas de CPU fijas y límites de CPU), como de asignaciones dinámicas que se aplican únicamente cuando la demanda supera la capacidad (cuotas de CPU flexibles).
- Autocontenido dentro del gestor de bases de datos DB2, el establecimiento de titularidades de recursos de CPU le proporciona un control de la carga de trabajo efectivo en todas las plataformas, debido a su independencia de un WLM de SO como AIX WLM o Linux WLM.
- Puede continuar utilizando los productos de WLM de SO como mecanismo de control de la carga de trabajo, aunque no es necesario si la complejidad de la implementación adicional (por ejemplo, configurar AIX WLM en cada miembro) o las reticencias organizativas (por ejemplo, el administrador del sistema se

muestra reacio a implementar o permitir el uso de WLM de SO) constituyen una barrera. Como alternativa, puede utilizar los productos de WLM de SO para supervisar objetivos a la vez que recurre al asignador de gestión de carga de trabajo de DB2 para el control de la carga de trabajo.

## Información adicional

En este apartado se proporciona información acerca de los conceptos administrativos, tareas y escenarios de uso para las áreas siguientes:

- Asignador de gestión de carga de trabajo de DB2
- Cuotas de CPU fijas
- Cuotas de CPU flexibles
- Límite de CPU
- Nivel de utilización de CPU mínimo para que la clase de servicio se considere activa
- Nivel de simultaneidad de asignación
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo

## Asignador de gestión de carga de trabajo

El asignador de gestión de carga de trabajo gestiona las titularidades de recursos de CPU para las cargas de trabajo que se ejecutan en el gestor de bases de datos DB2. Aquí se proporciona información detallada acerca de sus características y funciones.

### Introducción

Puede utilizar el asignador de gestión de carga de trabajo de DB2 incorporado para gestionar las cargas de trabajo de DB2 mediante la configuración de titularidades de recursos de CPU sin necesidad de software de gestión de carga de trabajo de terceros. El asignador puede controlar el consumo de recursos de CPU específicos de DB2 a través de la configuración basada en cuotas de CPU y de los valores de límite de la CPU. Las cuotas de CPU flexibles representan una asignación de recursos sin límite y pueden utilizarse para permitir que el trabajo de alta prioridad consuma más recursos de los que dispone su cuota asignada si hay recursos de CPU sin utilizar. Las cuotas de CPU fijas y los límites de CPU representan una asignación de CPU limitada que puede utilizarse para impedir que el trabajo de baja prioridad interfiera en la ejecución del trabajo de alta prioridad. Los límites de la CPU pueden utilizarse para controlar de manera estricta la CPU que consume una carga de trabajo a fin de proporcionar aislamiento para otro trabajo que se ejecute en el sistema, mientras que las cuotas de CPU fijas ofrecen una flexibilidad adicional al permitir que el trabajo de baja prioridad consuma recursos de CPU sin utilizar siempre que no haya carga de trabajo de alta prioridad en ejecución. Este caso de ejemplo puede producirse durante el horario de trabajo de menor actividad.

En los siguientes apartados se describen más detalladamente las características y funciones del asignador de gestión de carga de trabajo. Se ha incluido un apartado con diversos escenarios que le ayudará a entender las características y funciones del asignador de gestión de carga de trabajo con ejemplos de uso.

## Características y funciones

La infraestructura del asignador de gestión de carga de trabajo opera en el ámbito de la instancia, en el gestor de bases de datos DB2. Sin embargo, en el ámbito de la base de datos, han de especificarse las titularidades de recursos de CPU mediante la utilización de cuotas de CPU y atributos de límite de CPU, que pueden especificarse en una clase de servicio por medio de las sentencias CREATE SERVICE CLASS y ALTER SERVICE CLASS.

Como se ha indicado, los atributos de las cuotas de CPU fijas y flexibles pueden adquirir distintas condiciones bajo las que los recursos de CPU no utilizados, a los que han renunciado una o varias clases de servicio, son susceptibles de ser reclamados por el resto de las clases de servicio que compiten por los recursos. Sin embargo, los atributos de las cuotas de CPU fijas y flexibles tienen el mismo comportamiento cuando han de devolverse esos recursos de CPU reclamados. Por ejemplo, cuando una clase de servicio renuncia a parte o a toda su cuota total de recursos de CPU, las clases de servicio que han reclamado los recursos de CPU no utilizados deben volver a recibir los recursos de CPU para la clase de servicio que intenta recuperar su cuota total cuando ésta empieza a incrementar su demanda de recursos de CPU. Esto conduce a la total restauración de la titularidad de recursos de CPU especificada por el usuario para la clase de servicio que se recupera si existe demanda. Cada segundo, una hebra de planificador del gestor de bases de datos DB2 evalúa la utilización de recursos de CPU del trabajo que está en progreso y realiza ajustes para garantizar, con el transcurso del tiempo, que el asignador proporciona las cuotas de CPU y asignaciones de límite de CPU configuradas.

El funcionamiento del asignador de gestión de carga de trabajo consiste en limitar tanto el número de agentes en ejecución que se asignan simultáneamente al sistema operativo (SO) como el tiempo durante el cual está permitido que se ejecute cada agente. El número de agentes en ejecución que pueden asignarse al mismo tiempo se denomina nivel de simultaneidad de asignación, que puede establecer mediante la utilización del parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**.

Por omisión, el asignador de gestión de carga de trabajo no está habilitado. Para controlar las titularidades de recursos de CPU específicas del gestor de bases de datos de DB2, primero debe habilitar el asignador de gestión de carga de trabajo. Para habilitar el asignador de gestión de carga de trabajo, debe establecer el parámetro de configuración del gestor de bases de datos **wlm\_dispatcher** en YES (por omisión, este parámetro de configuración está establecido en NO). Para obtener completa información acerca de cómo habilitar el asignador de gestión de carga de trabajo, consulte: "Habilitación de un asignador de gestión de carga de trabajo" en la página 198.

Cuando el asignador de gestión de carga de trabajo está habilitado, todo el trabajo que se ejecuta en las clases de servicio de usuario y de mantenimiento dentro del gestor de bases de datos DB2 está bajo el control del asignador. El trabajo que se ejecuta en la clase de servicio del sistema no puede configurarse para el control de recursos de CPU porque a los subsistemas DB2 críticos que se ejecutan en esta clase de servicio se les otorga máxima prioridad y no están sujetos al control del asignador de gestión de carga de trabajo. Por omisión, el asignador, después de haberse activado, sólo puede gestionar los recursos de CPU por medio de valores de límite de CPU. Para que el asignador pueda gestionar los recursos de CPU utilizando cuotas de CPU y límites de CPU, debe establecer el parámetro de configuración del gestor de bases de datos **wlm\_disp\_cpu\_shares** en YES.

Antes o después de habilitar el asignador de gestión de carga de trabajo por primera vez, puede supervisar las cargas de trabajo para descubrir los recursos relativos de CPU que están consumiendo. En función de cuál sea la prioridad relativa de las cargas de trabajo, esta información podrá ayudarle a tomar decisiones relacionadas con la creación de clases de servicio a las que pueda asignarse el trabajo, el tipo de cuotas de CPU (fijas o flexibles) que han de asignarse a clases de servicio en particular, la cantidad relativa de cuotas de CPU que han de asignarse a cada clase de servicio y si han de utilizarse o no límites de CPU.

Otro punto a considerar que le proporcionará máxima flexibilidad para controlar el comportamiento del gestor de bases de datos DB2 es la opción de establecer el porcentaje mínimo de utilización de recursos de CPU para que se considere que las clases de servicio están activas. Después de haber establecido el parámetro de configuración del gestor de bases de datos `wlm_disp_min_util`, se considera que las clases de servicio que utilizan recursos de CPU equivalentes o superiores a ese porcentaje mínimo están activas en el sistema principal o en la partición lógica (LPAR) y las cuotas de CPU de las clases de servicio activas forman parte de los cálculos de titularidad de recursos de CPU.

Mediante el asignador de gestión de carga de trabajo, puede controlar la asignación de recursos de CPU entre clases de servicio del sistema por medio del atributo de cuotas de CPU de clase de servicio. El atributo de cuotas representa la titularidad relativa de tiempo de CPU que cada clase de servicio recibe en comparación con otras clases de servicio del sistema. Mediante la asignación de más cuotas de CPU a una clase de servicio y menos a otra, puede controlar la cantidad de recursos de CPU que se asignan a cada clase de servicio y proporcionar mejor calidad de servicio a una clase de servicio respecto a otra.

Después de haber determinado que el asignador de gestión de carga de trabajo habilitado anteriormente puede gestionar mejor los recursos de CPU de los que se hace un uso intensivo mediante la utilización de cuotas de CPU y de límites de CPU, debe habilitar las cuotas de CPU estableciendo el parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares` en YES. El valor por omisión de este parámetro es NO. Puede establecer y ajustar las cuotas de CPU y los límites de CPU mediante las sentencias CREATE SERVICE CLASS y ALTER SERVICE CLASS.

En función del número de cuotas de CPU que se asignan a una clase de servicio, el asignador de gestión de carga de trabajo calcula la parte de recursos de CPU para la que cada clase de servicio dispone de titularidad de uso. Para determinar la parte de recursos de CPU para la que cada superclase de servicio dispone de titularidad, puede utilizar la siguiente fórmula para convertir el número de cuotas de CPU de una superclase de servicio en particular en un porcentaje de recursos de CPU que el asignador de gestión de carga de trabajo asigna:

$$\% \text{ CPU}(\text{superclase}) = (\text{Número de cuotas de superclase} / \text{Número total de cuotas de todas las superclases activas}) \times 100$$

Para determinar la parte de recursos de CPU para la que cada subclase de servicio dispone de titularidad, puede utilizar la siguiente fórmula para convertir el número de cuotas de CPU de una subclase de servicio en particular en un porcentaje de recursos de CPU que el asignador de gestión de carga de trabajo asigna:

$$\% \text{ CPU}(\text{subclase}) = \% \text{ CPU}(\text{superclase}) \times (\text{Número de cuotas de subclase} / \text{Número total de cuotas de todas las subclases activas en la superclase})$$

**Nota:** Se calcula el número total de cuotas de CPU (fijas y flexibles) de todas las superclases activas de todas las bases de datos y todos los miembros del sistema principal o LPAR.

Para obtener ejemplos de cálculos de proporción de CPU, consulte el apartado "Escenarios de uso".

## Resumen de las características de planificación de recursos de CPU

En la siguiente tabla se proporciona un resumen de las características de los distintos atributos de control de recursos de CPU disponibles para las clases de servicio de DB2 que el asignador de gestión de carga de trabajo gestiona.

Tabla 44. Resumen de las características de planificación de recursos de CPU del asignador de gestión de carga de trabajo

Atributo de control de recursos de CPU	Uso de recursos de CPU desocupados en el mismo contexto	Límites de titularidad de recursos de CPU que son relativos respecto a otras clases de servicio activas en el mismo contexto	Límites de titularidad de recursos de CPU que son fijos en función de la capacidad de CPU del sistema global	Control de consumo de CPU en condiciones de alta utilización de recursos de CPU	Control de consumo de CPU en condiciones de baja utilización de recursos de CPU	Notas de uso
Cuotas flexibles	Y	Y	N	Y	N	Ideal para que el trabajo de alta prioridad pueda explotar cualquier recurso de CPU de reserva que esté disponible en el sistema principal o LPAR para maximizar el progreso de la carga de trabajo
Cuotas fijas	N	Y	N	Y	Y	Ideal para garantizar que el trabajo no interfiere con el trabajo de alta prioridad que podría estar ejecutándose en el sistema principal o LPAR en diversas cantidades e intensidad

Tabla 44. Resumen de las características de planificación de recursos de CPU del asignador de gestión de carga de trabajo (continuación)

Atributo de control de recursos de CPU	Uso de recursos de CPU desocupados en el mismo contexto	Límites de titularidad de recursos de CPU que son relativos respecto a otras clases de servicio activas en el mismo contexto	Límites de titularidad de recursos de CPU que son fijos en función de la capacidad de CPU del sistema global	Control de consumo de CPU en condiciones de alta utilización de recursos de CPU	Control de consumo de CPU en condiciones de baja utilización de recursos de CPU	Notas de uso
Límites de CPU	N	N	Y	Y	Y	Ideal para el recinto de seguridad y para aplicar asignaciones de recursos de CPU fijos coherentes entre clases de servicio o para limitar el consumo de recursos de CPU de clase de servicio con independencia de otro trabajo que pudiera estar ejecutándose en el sistema principal o LPAR

### Efectos de trabajar fuera del control del asignador de gestión de carga de trabajo

Existe un asignador de gestión de carga de trabajo para cada instancia de DB2. Sólo puede detectar y gestionar directamente el trabajo de DB2 que está dentro de su propia instancia. Cualquier otro trabajo que se ejecute en el mismo sistema principal o LPAR no está bajo el control directo del asignador. La naturaleza aleatoria de la demanda de recursos de CPU del trabajo que se ejecuta fuera del gestor de bases de datos DB2 implica que, aunque pueden aplicarse los límites de CPU y las cuotas relativas de recursos de CPU existentes entre el trabajo que se ejecuta dentro de DB2, en un momento dado, el trabajo que el asignador gestiona todavía debe competir con el trabajo que se ejecuta fuera de su propia instancia de DB2 y fuera del control directo del asignador. El trabajo que el asignador de gestión de carga de trabajo no detecta ni gestiona incluye lo siguiente:

- Trabajo que realizan las aplicaciones o productos de middleware que ejecutan parte de su trabajo fuera del gestor de bases de datos DB2
- Trabajo que realizan las entidades que se ejecutan en la clase de servicio del sistema de DB2
- Trabajo que realizan otras instancias de DB2

- Trabajo no DB2 que se realiza en procesos en modalidad delimitada, como los procedimientos almacenados delimitados
- Trabajo no DB2 que se realiza en rutinas fiables. Los procedimientos fiables y las funciones definidas por el usuario (UDF) se tienen en cuenta en la planificación de recursos de CPU del asignador, pero éste no puede limitar la utilización de CPU cuando una rutina fiable se ejecuta en código de usuario durante largos periodos de tiempo.

### Sistema

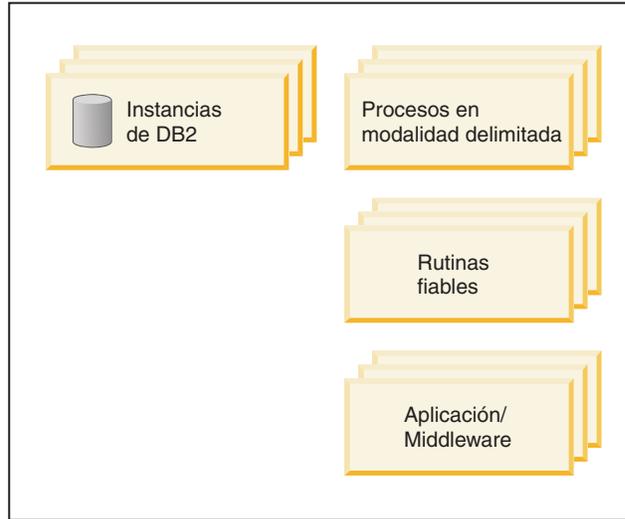


Figura 22. Ejemplo de diagrama del sistema

Imaginemos que tiene un sistema principal o LPAR que sólo ejecuta el gestor de bases de datos DB2. Ha creado una base de datos con las clases de servicio A y B y ha asignado una cuota del 50% de los recursos de CPU a cada clase de servicio. Si se da por supuesto que el sistema principal o LPAR global utiliza todos los recursos de CPU en todo momento, la medición que obtendrá de la utilización de CPU total (%) con el transcurso del tiempo tendrán un aspecto similar al que se muestra en el panel A de la Figura 23 en la página 183.

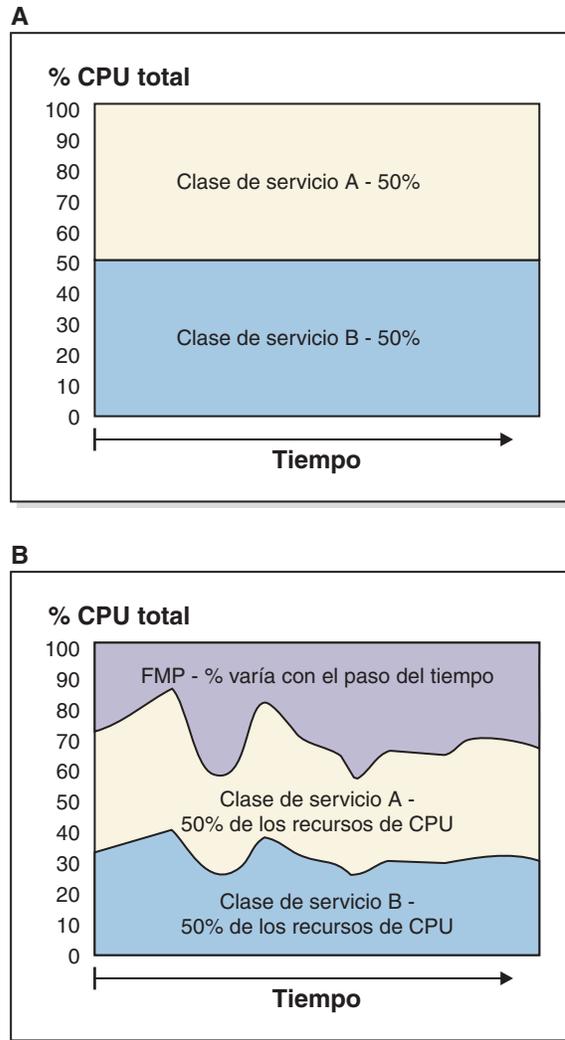


Figura 23. Porcentaje de utilización de CPU total

Ahora, vamos a considerar qué sucede cuando se ejecutan procesos en modalidad delimitada (FMP) en el sistema principal o LPAR. Estos procesos no están bajo el control del asignador de gestión de carga de trabajo, por lo tanto, la cantidad de recursos de CPU que éstos utilizan varía de forma aleatoria. Sin embargo, del resto de recursos de CPU que el asignador puede gestionar, las clases de servicio A y B siguen obteniendo, cada una de ellas, una división igual del 50% de los recursos de CPU restantes, tal como se muestra en el panel B.

### Interacciones entre los valores del asignador y otros valores de clase de servicio

En la siguiente tabla se proporciona un resumen de cómo interactúan los valores del asignador de gestión de carga de trabajo en una clase de servicio (cuotas de CPU fijas y flexibles y límite de CPU) con otros valores de la misma clase de servicio.

Tabla 45. Interacciones entre los valores del asignador y otros valores de clase de servicio

Valor de clase de servicio del asignador	Otro valor de clase de servicio	Interacción
Cuotas de CPU fijas y flexibles y límite de CPU	Prioridad de agente <b>Nota:</b> Se ha dejado de asociar cada clase de servicio de DB2 con una prioridad de agente, que controla la prioridad relativa del sistema operativo de los agentes en la clase de servicio. Es una función que podría eliminarse definitivamente en una versión futura. Utilice el asignador de gestión de carga de trabajo en lugar de la prioridad de agente para controlar el consumo de la CPU.	Cuando el parámetro de configuración del gestor de bases de datos <b>wlm_dispatcher</b> se establece en YES, la prioridad de agente se pasa por alto y los límites de CPU se activan. Cuando el parámetro <b>wlm_dispatcher</b> se establece en NO, los límites de CPU se pasan por alto y ya no se altera temporalmente la prioridad de agente.
Cuotas de CPU fijas y flexibles y límite de CPU	Correlacionador de salida (integración de WLM de SO)	Las cuotas de CPU fijas y flexibles, el límite de CPU y el WLM de SO pueden estar activos al mismo tiempo.
Cuotas de CPU fijas y flexibles y límite de CPU	Prioridad de agrupación de almacenamientos intermedios	Las cuotas de CPU fijas y flexibles, el límite de CPU y la prioridad de agrupación de almacenamientos intermedios pueden estar activos al mismo tiempo.
Cuotas de CPU fijas y flexibles y límite de CPU	Prioridad de captación previa	Las cuotas de CPU fijas y flexibles, el límite de CPU y la prioridad de captación previa pueden estar activos al mismo tiempo.

### Interacción entre los valores y umbrales del asignador con una acción REMAP

Algunos umbrales de gestión de carga de trabajo dan soporte a una acción REMAP ACTIVITY para mover el trabajo en ejecución desde una subclase de servicio hasta otra dentro de la misma superclase de servicio. Cuando el trabajo se mueve hasta una subclase de servicio que tiene una cuota de CPU o límite de CPU distintos de los de la subclase de servicio desde la que se mueve, se aplican los valores de cuota de CPU y de límite de CPU de la subclase de servicio hasta la que se mueve el trabajo.

### Comparación entre el asignador de gestión de carga de trabajo y los umbrales de simultaneidad de gestión de carga de trabajo

Aunque puede controlar la planificación de la carga de trabajo y el consumo de recursos mediante el asignador de gestión de carga de trabajo y los umbrales de simultaneidad, conocer las diferencias en el grado de control de la carga de trabajo y sus ventajas puede ayudarle a decidir cuándo aplicar uno o ambos en su entorno. En la siguiente tabla se comparan las posibilidades de control de recursos de CPU de éstos.

Tabla 46. Comparación de las posibilidades de control de recursos de CPU entre el asignador de gestión de carga de trabajo y los umbrales de simultaneidad de gestión de carga de trabajo

Asignador de gestión de carga de trabajo	Umbrales de simultaneidad de gestión de carga de trabajo de DB2
Puede controlar directamente las asignaciones de CPU del gestor de bases de datos DB2 después de que una actividad empieza a consumir recursos de CPU.	Puede controlar indirectamente tanto el consumo de la CPU como de otros recursos, como la memoria, limitando el número de conexiones o actividades que puedan ejecutarse simultáneamente.
Puede configurar asignaciones relativas de cuotas flexibles y fijas de recursos de CPU para las clases de servicio y límites absolutos de consumo de CPU para una única clase de servicio.	Puede imponer un límite absoluto para la simultaneidad de las actividades en ejecución o abrir conexiones para una clase de servicio, pero no puede especificar ningún tipo de asignación de CPU relativa.

### Comparación entre el asignador de gestión de carga de trabajo y las gestiones de carga de trabajo de sistema operativo (AIX WLM y Linux WLM)

El asignador de gestión de carga de trabajo puede utilizarse conjuntamente con un WLM de sistema operativo (AIX WLM o Linux WLM) en determinadas circunstancias para proporcionar un control incluso mayor. A continuación se facilitan algunos ejemplos de circunstancias en las que pueden utilizarse conjuntamente el asignador de gestión de carga de trabajo y un WLM de sistema operativo:

- Los WLM de sistema operativo suelen disponer de una supervisión de sistema operativo que complementa las posibilidades de supervisión del asignador de gestión de carga de trabajo.
- Los WLM de SO pueden gestionar todas las aplicaciones del sistema, incluido el DBMS. El asignador de gestión de carga de trabajo puede proporcionar asignación específica de DB2, mientras que el WLM de SO gestiona todas las demás aplicaciones del sistema.
- No intente controlar las asignaciones de recursos de CPU para clases de servicio individuales con el WLM de sistema operativo y el asignador de gestión de carga de trabajo al mismo tiempo. Puede integrar un WLM de SO en el ámbito de la clase de servicio para realizar tareas de supervisión cuando el asignador de gestión de carga de trabajo está habilitado. También puede integrar un WLM de SO para controlar las aplicaciones externas o los recursos de CPU globales asignados a una instancia de DB2, o ambas cosas.

Tabla 47. Comparación de las posibilidades del asignador de gestión de carga de trabajo y del WLM de SO

Asignador de gestión de carga de trabajo	WLM de SO
Una única solución que funciona en toda las plataformas.	Un WLM de SO es específico de cada SO. Cada uno de ellos tiene su propia interfaz de usuario y sus propias limitaciones.
Puede controlar las asignaciones de CPU dentro del gestor de bases de datos DB2 sin necesidad de ninguna integración específica del SO ni configuración; un DBA puede gestionar estos valores sin necesitar acceso de usuario root para el sistema.	Un WLM de SO suele requerir privilegios de usuario root para la configuración.

Tabla 47. Comparación de las posibilidades del asignador de gestión de carga de trabajo y del WLM de SO (continuación)

Asignador de gestión de carga de trabajo	WLM de SO
Puede beneficiarse del conocimiento específico de la base de datos en la toma de decisiones relacionadas con la planificación de recursos de CPU que, de otro modo, no serían posibles con los recursos disponibles en el SO.	Un WLM de SO trata al gestor de bases de datos DB2 como a cualquier otra aplicación y, por lo tanto, no puede utilizar el conocimiento específico de la base de datos en la toma de decisiones relacionadas con la planificación de los recursos de CPU.
En los entornos de varios miembros, todos los miembros pueden configurarse con un único mandato. Sin embargo, el resultado es que no puede tener una configuración distinta en cada miembro.	En los entornos de varios miembros, debe configurarse un WLM de SO en cada miembro. Puede tener una configuración distinta en cada miembro.
Controla el uso de recursos de CPU que realiza sólo el trabajo de DB2 que está en ejecución en las clases de servicio de usuario y de mantenimiento dentro del ámbito de una única instancia de DB2. El asignador no puede controlar el uso de recursos de CPU que realizan las aplicaciones que se ejecutan de forma externa respecto a la instancia de DB2.	Un WLM de SO puede controlar por completo los recursos de CPU de todo el sistema.
El control de recursos de CPU se realiza por medio de cuotas de CPU flexibles y fijas.	Las soluciones de WLM de SO AIX y Linux no proporcionan actualmente un equivalente de las cuotas de CPU fijas de DB2.

## Comparación entre el asignador de gestión de carga de trabajo y la prioridad de agente

Puede utilizarse el control de la carga de trabajo mediante la prioridad de agente. En la siguiente tabla se comparan las posibilidades de control de recursos de CPU de la prioridad de agente con las del asignador de gestión de carga de trabajo.

Tabla 48. Comparación de las posibilidades de control de recursos de CPU entre el asignador de gestión de carga de trabajo y la prioridad de agente

Asignador de gestión de carga de trabajo	Prioridad de agente
Es un método que permite imponer límites o realizar una asignación coherente y precisa del uso de recursos de CPU entre clases de servicio.	La prioridad de agente no puede imponer límites relacionados con el uso de los recursos de CPU que una clase de servicio realiza. La prioridad de agente sólo puede dar prioridad a una clase de servicio respecto a otra, sin que exista ningún control granular relacionado con las titularidades de recursos de CPU.

**Nota:** Se ha dejado de asociar cada clase de servicio de DB2 con una prioridad de agente, que controla la prioridad relativa del sistema operativo de los agentes en la clase de servicio. Es una función que podría eliminarse definitivamente en una versión futura. Utilice el asignador de gestión de carga de trabajo en lugar de la prioridad de agente para controlar el consumo de la CPU.

## Escenarios de uso

### Escenario 1: Cálculos de titularidad de CPU

En la Figura 24 se muestra el servidor de datos configurado como se ha indicado. Para que la ilustración pueda servir de ayuda para describir los conceptos básicos, imaginemos que existe trabajo en ejecución inapreciable en las clases de servicio de usuario, mantenimiento y sistema por omisión. Al principio del escenario, imaginaremos que existe suficiente trabajo en ejecución en las superclases de servicio de usuario A, B y C para utilizar todas las titularidades de recursos de CPU para esas clases de servicio. Para el siguiente escenario, sólo existe una única instancia de DB2 con una base de datos y un único miembro lógico en este sistema principal o LPAR.

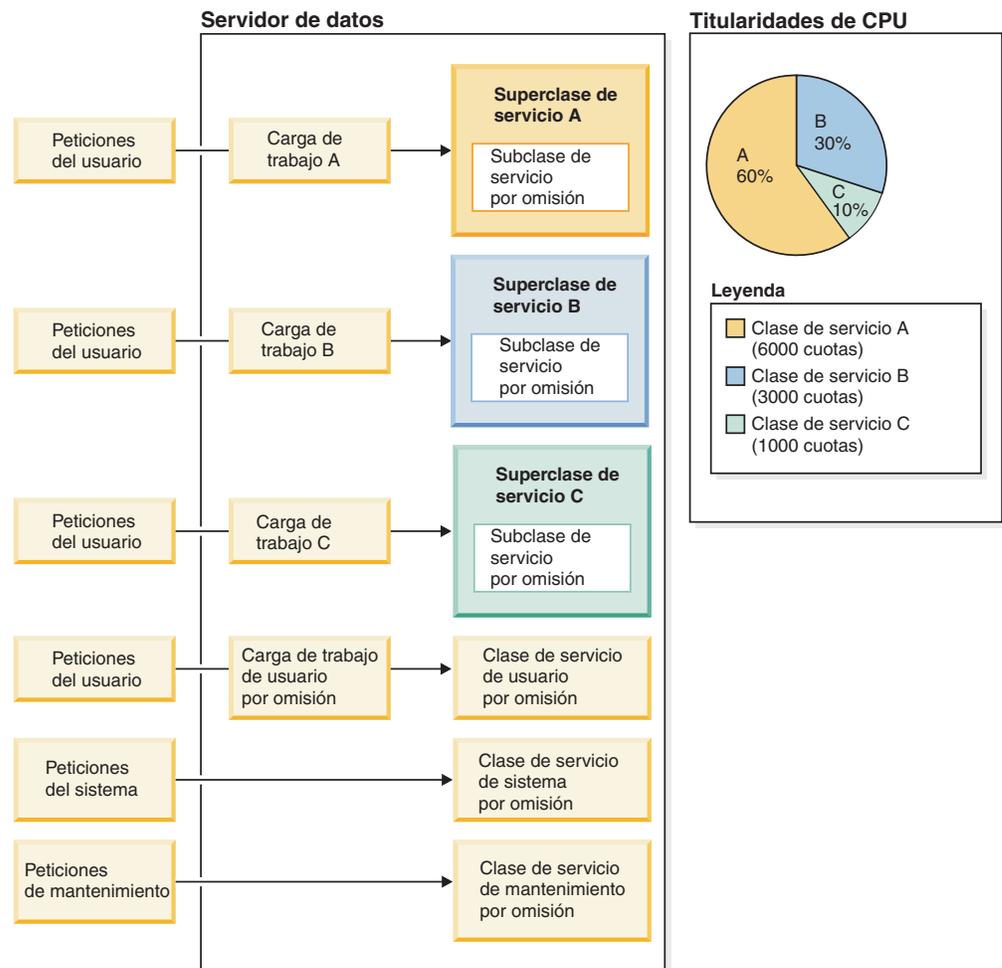


Figura 24. Configuración del servidor de datos

Como se muestra en la Figura 24, el porcentaje de titularidad de CPU para cada una de las superclases de servicio se calcula de la forma siguiente:

- La superclase de servicio A tiene el 60%  
 $(6000 / (6000 + 3000 + 1000)) \times 100$
- La superclase de servicio B tiene el 30%  
 $(3000 / (6000 + 3000 + 1000)) \times 100$

- La superclase de servicio C tiene el 10%  
 $(1000 / (6000 + 3000 + 1000)) \times 100$

Pueden asignarse cuotas de CPU para la superclase de servicio y la subclase de servicio. Si no especifica cuotas de CPU al crear una superclase de servicio o una subclase de servicio, el gestor de bases de datos DB2 asigna un valor por omisión de 1000 cuotas de CPU fijas a estas clases de servicio. Para las superclases, el valor de las cuotas de CPU representa cómo se dividen los recursos de CPU del sistema entre las superclases de servicio de DB2. Para las subclases, el valor de las cuotas de CPU representa la división de los recursos de CPU disponibles para una superclase en particular entre sus subclases activas.

**Escenario 2: El asignador de gestión de carga de trabajo funciona en el ámbito de la instancia; las clases de servicio funcionan en el ámbito de la base de datos**

El asignador de gestión de carga de trabajo funciona en el ámbito de la instancia, pero las configuraciones (cuotas de CPU y límites de CPU) que el asignador utiliza para asignar recursos de CPU se especifican en superclases de servicio y subclases de servicio individuales dentro de bases de datos individuales. Cuando existen varias bases de datos dentro de la misma instancia, la suma de los valores de cuota de CPU de las superclases de servicio de cada base de datos determina la asignación relativa de recursos de CPU que cada base de datos recibe. Por ejemplo, existen dos bases de datos en la misma instancia, la base de datos A y la base de datos B, cada una de las cuales tiene dos superclases de servicio (SC1 y SC2), tal como se muestran en el panel A de la Figura 25 en la página 189.

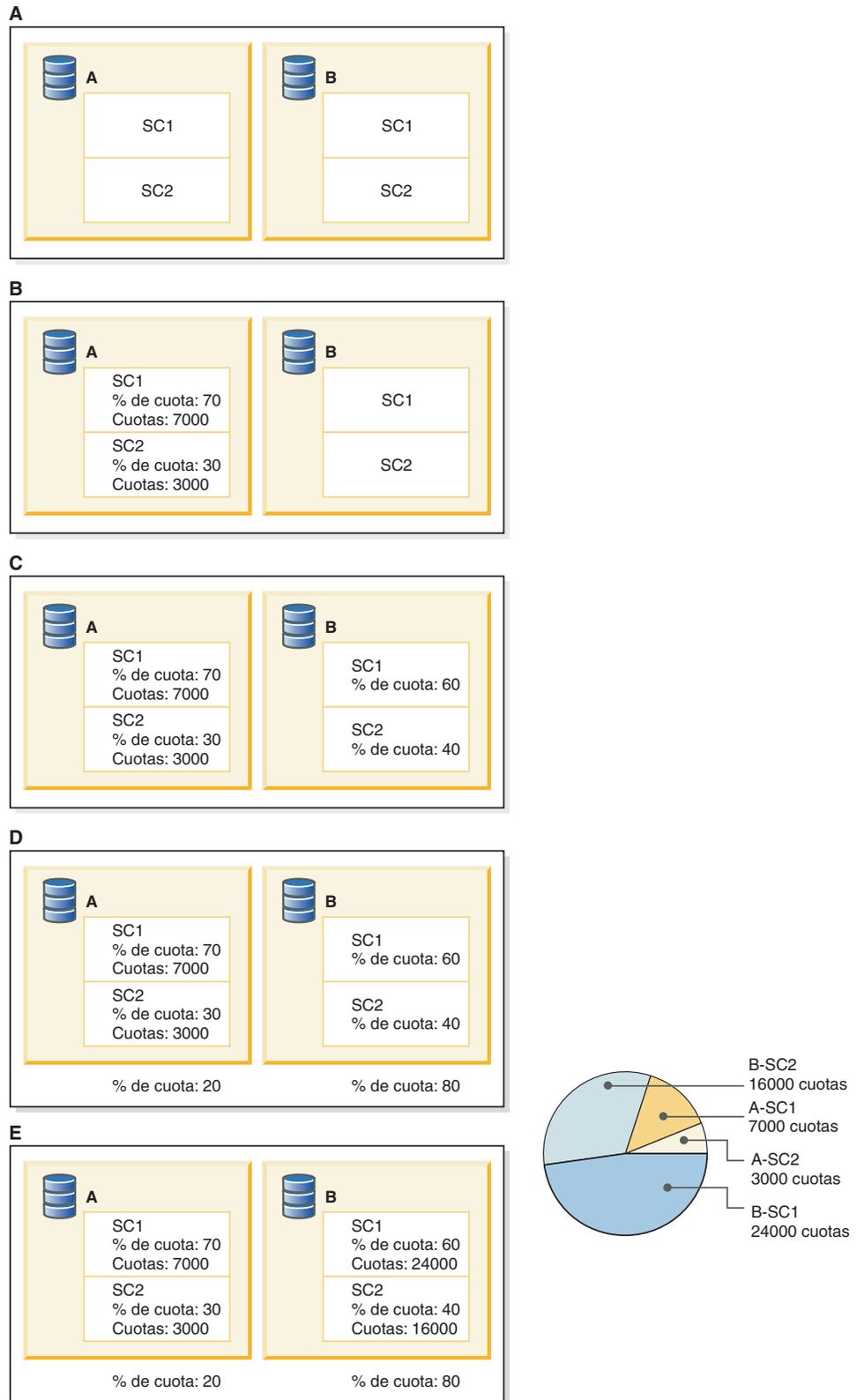


Figura 25. Asignaciones de cuotas de CPU a las clases de servicio para dos bases de datos en una única instancia

En este escenario de ejemplo, deseamos una división porcentual de 70/30 de los recursos de CPU en la base de datos A que se ha configurado utilizando 7000 y

3000 para las cuotas de CPU flexibles para SC1 y SC2 respectivamente, tal como se muestra en el panel B. En la base de datos B, deseamos una división porcentual de 60/40 de los recursos de CPU para SC1 y SC2 respectivamente, tal como se muestra en el panel C.

Antes de decidir cómo ha de traducirse esta división porcentual 60/40 en cuotas de CPU en las dos superclases de servicio para la base de datos B, debemos decidir cuáles son las titularidades de recursos de CPU relativas necesarias entre las bases de datos A y B. El trabajo que se ejecuta en la base de datos B es mucho más importante que el trabajo que se ejecuta en la base de datos A. Por lo tanto, deseamos que la base de datos B tenga una titularidad de recursos de CPU del 80% y que la base de datos A tenga un 20%.

La suma de las cuotas de CPU de la base de datos A es 10.000. Estas 10.000 cuotas de CPU representan en 20% de las cuotas de CPU totales de las dos bases de datos, y necesitamos 50.000 cuotas de CPU en total para representar el 100% de los recursos de CPU totales. Por lo tanto, a la base de datos B se le debe asignar el 80% de esas 50.000 cuotas de CPU, que serían 40.000 cuotas de CPU. Para obtener la división porcentual 60/40 entre las dos superclases de la base de datos B, la superclase de servicio SC1 recibe el 60% de las 40.000 cuotas de CPU totales asignadas en la base de datos B, que serían 24.000 cuotas de CPU. La superclase de servicio SC2 recibe el 40% de las 40.000 cuotas de CPU totales asignadas en la base de datos B, que son 16.000 cuotas de CPU.

En el gráfico circular se muestran las asignaciones de cuotas de CPU que constituyen las asignaciones de recursos de CPU que deseábamos.

### **Escenario 3: Dar prioridad a una carga de trabajo respecto a otra en condiciones de alta utilización de CPU y proteger la carga de trabajo prioritaria en condiciones de baja utilización de CPU**

En este escenario de uso de ejemplo, la finalidad es gestionar la calidad de servicio de varias cargas de trabajo a la vez que, simultáneamente, se maximiza la utilización de CPU del sistema. Sin embargo, las cargas de trabajo de este escenario, además de experimentar contención por los recursos de CPU, también experimentan contención por otros recursos. La asignación de recursos de CPU no utilizados a algunas cargas de trabajo cuando todavía existen cargas de trabajo que están compitiendo, y que no consumen toda su titularidad de CPU, puede dar lugar a una degradación del rendimiento debido a la contención de antememoria de CPU, contención de E/S u otros efectos secundarios.

En este escenario, es necesario proteger la carga de trabajo a la que se ha dado prioridad respecto a las cargas de trabajo a las que no se ha dado prioridad, a pesar de que exista la posibilidad de que los recursos de CPU no se utilicen al completo. Puede darse el caso de que existan recursos de CPU que se utilizan por debajo de sus posibilidades cuando las cargas de trabajo no prioritarias están limitadas y la carga de trabajo prioritaria está activa, pero no utiliza toda su titularidad de CPU. Podemos llevar a cabo la protección de la carga de trabajo prioritaria asignando cuotas de CPU fijas en las cargas de trabajo no prioritarias y asignando cuotas de CPU flexibles en la carga de trabajo prioritaria.

Utilicemos nuevamente dos cargas de trabajo: "prioritaria" y "otra". La carga de trabajo "prioritaria" ha de disponer del 70% de los recursos de CPU cuando se ejecute al mismo tiempo que la carga de trabajo "otra", y el 100% cuando la carga de trabajo "otra" no esté presente. La carga de trabajo "otra" ha de disponer del 30% de los recursos de CPU cuando se ejecute al mismo tiempo que la carga de

trabajo "prioritaria", y el 100% cuando la carga de trabajo "prioritaria" no esté presente. Es similar al Escenario 3, a excepción de que la utilización de CPU global cae por debajo del 100% en algunas ocasiones.

En los pasos siguientes se muestra cómo configurar la gestión de carga de trabajo de DB2 para lograr los objetivos listados anteriormente:

1. Cree las superclases de servicio "prioritaria" y "otra" y asigne cuotas de CPU de modo que se satisfaga la proporción de titularidad de CPU 70/30. Las cuotas de CPU pueden especificarse como 7 y 3, 14 y 6, o cualquier otra combinación que represente la proporción de titularidad de CPU 70%/30%. En el ejemplo siguiente, se asignan 7000 cuotas de CPU flexibles a la clase de servicio "prioritaria" ("favored" en el ejemplo) y se asignan 3000 cuotas de CPU fijas a la clase de servicio "otra" ("other" en el ejemplo).

```
create service class favored soft cpu shares 7000
create service class other hard cpu shares 3000
```

2. Cree las cargas de trabajo "prioritaria" ("favored" en el ejemplo) y "otra" ("other" en el ejemplo) tal como se muestra en el ejemplo siguiente:

```
create workload favored appl_name('favoredapp') service class favored
create workload other appl_name('otherapp') service class other
```

Como resultado de ello, el aspecto del sistema es similar al siguiente:

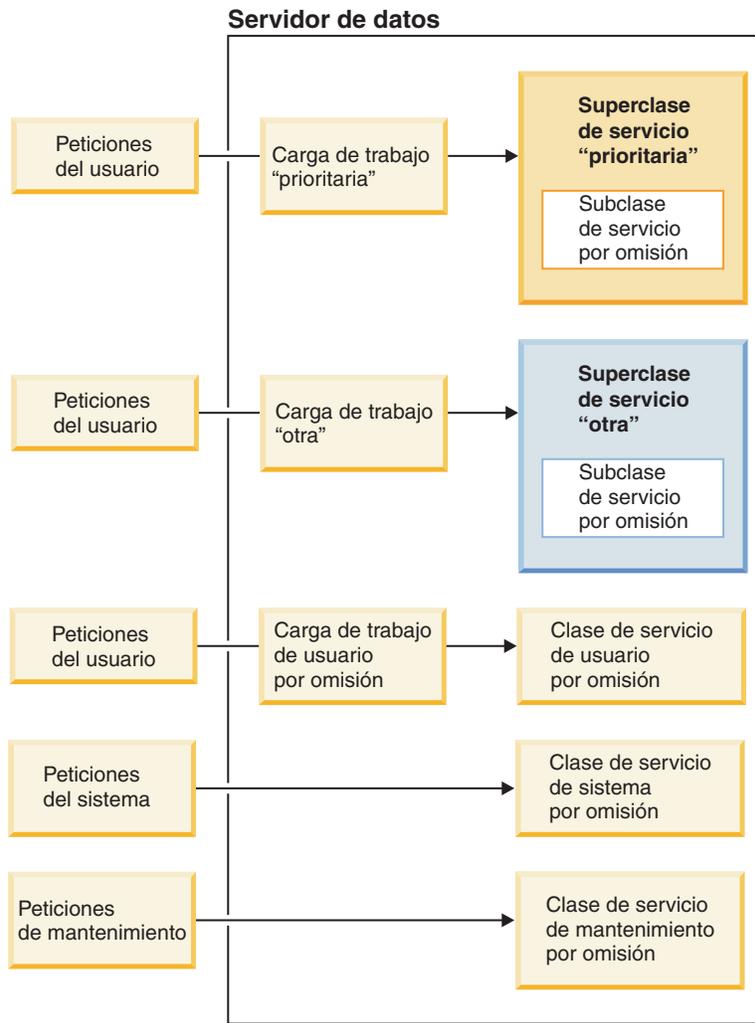


Figura 26. Configuración del servidor de datos

Para simplificar el ejemplo, imaginemos que sólo está ejecutándose una cantidad de trabajo inapreciable en las clases de servicio distintas de "prioritaria" y "otra".

3. Determine si los objetivos de cuota de CPU deseados se consiguen, midiendo éstos. El sistema puede encontrarse en varios estados en términos de su carga de trabajo:
  - a. Utilización de CPU del 100%
    - 1) El trabajo se ejecuta en la carga de trabajo "prioritaria" y en la carga de trabajo "otra"  
La utilización de CPU medida es del 70% para la clase de servicio "prioritaria" y del 30% para la clase de servicio "otra".
    - 2) El trabajo sólo se ejecuta en la carga de trabajo "prioritaria"  
La utilización de CPU medida es del 100% para la clase de servicio "prioritaria" y del 0% para la clase de servicio "otra".
    - 3) El trabajo sólo se ejecuta en la carga de trabajo "otra"  
La utilización de CPU medida es del 0% para la clase de servicio "prioritaria" y del 100% para la clase de servicio "otra".
  - b. Una clase de servicio utiliza una titularidad de CPU menor que la asignada

- 1) El trabajo se ejecuta en la carga de trabajo "prioritaria" y en la carga de trabajo "otra", en ambas. La carga de trabajo "prioritaria" utiliza el 50% en lugar del 70% total que le corresponde de su titularidad de CPU.  
La utilización de CPU medida de la carga de trabajo "prioritaria" es del 50%. La carga de trabajo "otra" sigue utilizando el 30% porque sus cuotas de CPU son fijas. La utilización de CPU total es del 80%.
- 2) El trabajo se ejecuta en la carga de trabajo "prioritaria" y en la carga de trabajo "otra", en ambas. La carga de trabajo "otra" utiliza el 10% en lugar del 30% total que le corresponde de su titularidad de CPU.  
Debido a que sus cuotas de CPU son flexibles, la utilización de CPU medida de la clase de servicio "prioritaria" se incrementa del 70% al 90%, porque reclama el 20% no utilizado al que la clase de servicio "otra" ha renunciado. La utilización de CPU de la clase de servicio "otra" es del 10%. La utilización de CPU total es del 100%.
- 3) El trabajo sólo se ejecuta en la carga de trabajo "prioritaria" y utiliza todas sus titularidades de CPU  
Debido a la ausencia de trabajo en ejecución en la clase de servicio "otra", la clase de servicio "prioritaria" puede utilizar su propia titularidad de CPU del 70% y el 30% no utilizado de recursos de CPU a los que ha renunciado la clase de servicio "otra". La utilización de CPU total es del 100%.
- 4) El trabajo sólo se ejecuta en la carga de trabajo "otra" y utiliza todas sus titularidades de CPU  
Debido a la ausencia de trabajo en ejecución en la clase de servicio "prioritaria", la clase de servicio "otra" puede utilizar su propia titularidad de CPU del 30% y el 70% no utilizado de recursos de CPU a los que ha renunciado la clase de servicio "prioritaria". La utilización de CPU total es del 100%.

#### **Escenario 4: crear un recinto de seguridad para lograr coherencia, incluso a expensas de que diversos recursos de CPU quedan desocupados**

En este escenario de uso de ejemplo, una carga de trabajo prioritaria nunca consume más recursos de CPU que los correspondientes a su proporción fija, con independencia de la disponibilidad de recursos de CPU no utilizados.

Utilicemos dos cargas de trabajo: "prioritaria" y "otra". La carga de trabajo "prioritaria" ha de disponer de un máximo del 70% de los recursos de CPU en cualquier momento, con independencia de la disponibilidad de recursos de CPU no utilizados. La carga de trabajo "otra" ha de disponer de un máximo del 30% de los recursos de CPU en cualquier momento, con independencia de la disponibilidad de recursos de CPU no utilizados.

En los pasos siguientes se muestra cómo configurar la gestión de carga de trabajo de DB2 para lograr los objetivos listados anteriormente:

1. Cree las clases de servicio "prioritaria" y "otra" y asigne a cada una de éstas un límite de CPU para que se aplique el máximo del 70% y del 30% de recursos de CPU a las clases de servicio "prioritaria" y "otra", respectivamente. En el siguiente ejemplo, los límites de CPU deben especificarse en forma de número comprendido entre el 1 y el 100, que representa el porcentaje máximo de utilización de CPU para esa clase de servicio:

```
create service class favored cpu limit 7000
create service class other cpu limit 3000
```

2. Cree las cargas de trabajo "prioritaria" ("favored" en el ejemplo) y "otra" ("other" en el ejemplo) tal como se muestra en el ejemplo siguiente:

```
create workload favored appl_name('favoredapp') service class favored
create workload other appl_name('otherapp') service class other
```

Como resultado de ello, el aspecto del sistema es similar al que se muestra en la Figura 26 en la página 192.

Para simplificar el ejemplo, imaginemos que sólo está ejecutándose una cantidad de trabajo inapreciable en las clases de servicio distintas de "prioritaria" y "otra".

3. Determine si se han aplicado correctamente los límites de CPU deseados, midiendo éstos. El sistema puede encontrarse en varios estados en términos de su carga de trabajo:
  - a. Utilización de CPU del 100%
    - 1) El trabajo se ejecuta en la carga de trabajo "prioritaria" y en la carga de trabajo "otra"  
La utilización de CPU medida es del 70% para la clase de servicio "prioritaria" y del 30% para la clase de servicio "otra".
  - b. Una o varias clases de servicio utilizan un límite de CPU menor al que les corresponde
    - 1) El trabajo se ejecuta en la carga de trabajo "prioritaria" y en la carga de trabajo "otra", en ambas. La carga de trabajo "prioritaria" utiliza el 50% en lugar del 70% total que le corresponde de su titularidad de límite de CPU.  
La utilización de CPU medida de la carga de trabajo "prioritaria" es del 50%. La carga de trabajo "otra" sigue utilizando el 30% porque es la titularidad de límite de CPU máxima. La utilización de CPU total es del 80%.
    - 2) El trabajo se ejecuta en la carga de trabajo "prioritaria" y en la carga de trabajo "otra", en ambas. La carga de trabajo "otra" utiliza el 10% en lugar del 30% total que le corresponde de su titularidad de límite de CPU.  
Debido a su límite de CPU, que estipula un máximo de utilización de CPU del 70%, la utilización de CPU medida de la clase de servicio "prioritaria" sigue en el 70%. El 20% no utilizado de recursos de CPU a los que ha renunciado la clase de servicio "otra" sigue sin reclamarse. La utilización de CPU de la clase de servicio "otra" es del 10%. La utilización de CPU total es del 80%.
    - 3) El trabajo sólo se ejecuta en la carga de trabajo "prioritaria" y utiliza la cantidad máxima de recursos de CPU que estipula su límite de CPU  
Debido a su límite de CPU, que estipula un máximo de utilización de CPU del 70%, la utilización de CPU medida de la clase de servicio "prioritaria" sigue en el 70%. El 30% no utilizado de recursos de CPU a los que ha renunciado la clase de servicio "otra" sigue sin reclamarse. La utilización de CPU de la clase de servicio "otra" es del 0%. La utilización de CPU total es del 70%.
    - 4) El trabajo sólo se ejecuta en la carga de trabajo "otra" y utiliza la cantidad máxima de recursos de CPU que estipula su límite de CPU  
Debido a su límite de CPU, que estipula un máximo de utilización de CPU del 30%, la utilización de CPU medida de la clase de servicio "otra" sigue en el 30%. El 70% no utilizado de recursos de CPU a los que ha

renunciado la clase de servicio "prioritaria" sigue sin reclamarse. La utilización de CPU de la clase de servicio "prioritaria" es del 0%. La utilización de CPU total es del 30%.

### **Escenario 5: Es necesario dividir las cargas de trabajo por usuario y seguir dando prioridad a un tipo de carga de trabajo respecto a otra para un usuario determinado**

En los escenarios anteriores, las clases de servicio se utilizaban para controlar la asignación de recursos de CPU entre cargas de trabajo distintas en función de las prioridades de la empresa. En algunos casos, puede que se desee controlar la cuota de recursos entre cargas de trabajo basándose en las prioridades de la empresa y también dar prioridad al trabajo de corta ejecución respecto al trabajo de larga ejecución para mejorar el rendimiento global o los tiempos de respuesta.

En este escenario de uso de ejemplo, un director general y otros cien empleados acceden a una base de datos por medio de una aplicación denominada mybizapp. Nuestro objetivo es asignar al director general una cuota del 10% total de los recursos de CPU y utilizar el 90% restante de los recursos de CPU para los cien empleados que acceden a la base de datos. Para cumplir este objetivo, cree una superclase de servicio para el directorio general, "ceo" en el ejemplo (ceo\_sc). A continuación, defina una carga de trabajo para correlacionar al director general ("ceo" en el ejemplo) con su superclase de servicio, tal como se muestra en los ejemplos siguientes:

```
create service class ceo_sc soft cpu shares 1000
create workload ceo_wl
  applname('mybizapp')
  session_user('ceo')
  service class ceo_sc
```

Cree una superclase de servicio para todos los demás usuarios (mybizapp\_sc). A continuación, defina una carga de trabajo (mybizapp\_wl) para correlacionar todos los demás usuarios de la aplicación mybizapp con la superclase de servicio que acaba de crearse mybizapp\_sc, tal como se muestra en el ejemplo siguiente:

```
create service class mybizapp_sc hard cpu shares 9000
create workload mybizapp_wl
  applname('mybizapp')
  service class mybizapp_sc
```

Dentro de la carga de trabajo de la aplicación mybizapp, existen consultas de alto coste y consultas de bajo coste. La organización empresarial ha determinado que las consultas de bajo coste, cuyo coste es inferior a 10.000 timerons, generalmente tienen una prioridad más alta que las consultas de alto coste, cuyo coste supera los 10.000 timerons. Por lo tanto, a las consultas de bajo coste ha de asignarse una cuota de recursos de CPU mayor que la de las consultas de alto coste, específicamente en una proporción de 2:1. Para lograr este objetivo, cree dos superclases de servicio (lowcost\_ssc y highcost\_ssc) dentro de la superclase de servicio mybizapp\_sc y asigne cuotas de CPU flexibles en una proporción de 2:1, tal como se muestra en el siguiente ejemplo:

```
create service class lowcost_ssc under mybizapp_sc soft cpu shares 2000
create service class highcost_ssc under mybizapp_sc soft cpu shares 1000
```

La proporción 2:1 de cuotas de CPU flexibles determina qué parte del 90% de recursos de CPU asignados a la superclase de servicio mybizapp\_sc ha de asignarse a las consultas de bajo coste y qué parte ha de asignarse a las consultas de alto coste.

Cree un conjunto de clases de trabajo (splitbycost\_wcs) y un conjunto de clases de trabajo (mybizapp\_was) para direccionar las consultas de bajo coste a la subclase de servicio lowcost\_ssc y las consultas de alto coste a la subclase de servicio highcost\_ssc, tal como se muestra en el siguiente ejemplo:

```
create work class set splitbycost_wcs
  (work class lowcost_wc work type dml for timeroncost from 0 to 10000,
   work class highcost_wc work type dml for timeroncost from 10001)

create work action set mybizapp_was for service class mybizapp_sc
  using work class set splitbycost_wcs
  (work action maplowcost_wa on work class lowcost_wc map activity
   to lowcost_ssc, work action maphighcost_wa on work class highcost_wc
   map activity to highcost_ssc)
```

Cuando el trabajo se ejecuta en cada una de las clases de servicio y consume su titularidad de recursos de CPU total, las configuraciones anteriores producen una división de los recursos de CPU, tal como se muestra en la siguiente figura:

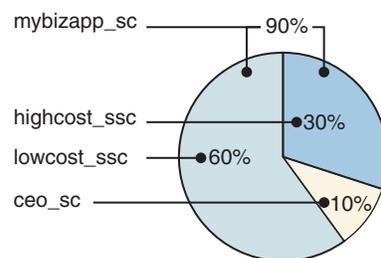


Figura 27. Titularidades de recursos de CPU

La superclase de servicio ceo\_sc tiene 1000 cuotas de CPU de las 10.000, lo que da como resultado una titularidad de recursos de CPU de 10%. La superclase de servicio mybizapp\_sc tiene 9000 cuotas de CPU de las 10.000, lo que da como resultado una titularidad de recursos de CPU de 90%. Del 90% de la titularidad de recursos de CPU de la superclase de servicio mybizapp\_sc, su subclase de servicio lowcost\_ssc tiene 2 cuotas de las 3, lo que da como resultado una titularidad de recursos de CPU del 66,7% de los recursos de CPU de su superclase de servicio, o el 60% de los recursos de CPU totales. La subclase de servicio highcost\_ssc tiene 1 cuota de 3, lo que da como resultado una titularidad de recursos de CPU del 33,3% de los recursos de CPU de su superclase de servicio, o el 30% de los recursos de CPU totales.

## Información adicional

Se proporciona completa información acerca de los temas siguientes:

- Cuotas de CPU fijas, consulte: “Cuotas de CPU fijas” en la página 200.
- Cuotas de CPU flexibles, consulte: “Cuotas de CPU flexibles” en la página 209.
- Límite de CPU, consulte: “Límite de CPU” en la página 215.
- Nivel de utilización de CPU mínimo para que se considere activa una clase de servicio, consulte: “Utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 230.
- Nivel de simultaneidad de asignación, consulte: “Nivel de simultaneidad de asignación” en la página 234.
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo” en la página 237.

## Comportamiento del asignador de gestión de carga de trabajo en entornos de varios miembros

Cuando el asignador de gestión de carga de trabajo está habilitado en instancias de DB2 de varios miembros, la planificación de recursos de CPU funciona en todos los miembros en un sistema principal determinado. Tenga en cuenta que el asignador de gestión de carga de trabajo también admite entornos LPAR compartidos (micropartición) en AIX.

### Entornos de base de datos de varios miembros

En instancias de DB2 de varios miembros, el asignador tiene el comportamiento siguiente:

- Se evalúa la cantidad relativa de cuotas de CPU y se asignan recursos de CPU en todos los miembros de la instancia en función del trabajo que está activo en esa instancia de ese sistema principal o LPAR en su conjunto. Por ejemplo, imaginemos que existe una base de datos con dos miembros cuyo trabajo se ejecuta en dos clases de servicio, A y B. A la clase de servicio A se le asignan 3500 cuotas de CPU flexibles y a la clase de servicio B se le asignan 6500 cuotas de CPU flexibles. La utilización de CPU para todo el sistema principal o LPAR es del 35% para la clase de servicio A y del 65% para la clase de servicio B, a pesar de que existe la posibilidad de que se ejecute una cantidad de trabajo relativamente superior en un miembro que en otro. El asignador de gestión de carga de trabajo examina ambos miembros cuando realiza la toma de decisiones relacionadas con la clase de servicio que ha de obtener más recursos de CPU en un momento determinado.

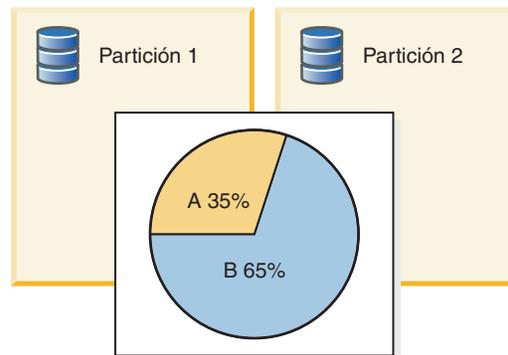


Figura 28. Entornos de bases de datos de varios miembros: gráfico circular de asignaciones de cuotas de CPU flexibles en dos miembros de base de datos

- El valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur** se aplica a cada sistema principal o LPAR. En las bases de datos de varios miembros, el nivel de simultaneidad de asignación que especifica el parámetro de configuración **wlm\_disp\_concur** se aplica a todos los miembros.
- Para las bases de datos de varios miembros, los límites de CPU se aplican individualmente a cada sistema principal o LPAR; cuando se utilizan varios miembros, cada partición puede utilizar la totalidad de un límite de CPU especificado siempre que la suma de todos los miembros sea inferior o igual al límite de CPU. Si el límite de CPU de la clase de servicio A es del 35% y existen dos miembros, el miembro 1 puede utilizar hasta el 35% si la suma de la utilización de CPU de los miembros 1 y 2 no excede el 35%.

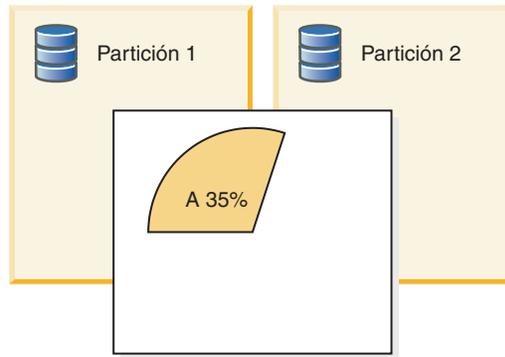


Figura 29. Entornos de bases de datos de varios miembros: gráfico circular del límite de CPU en dos miembros de base de datos

### Entornos de micropartición (LPAR compartidas)

El asignador de gestión de carga de trabajo da soporte a los entornos de micropartición en AIX. En los entornos de micropartición, el porcentaje de utilización de CPU para los controles y la supervisión del asignador de gestión de carga de trabajo se calcula en relación con los recursos de CPU que el sistema principal o el hipervisor ha asignado a la LPAR en el ciclo de planificación de recursos de CPU del asignador más reciente, mediante la utilización de los recursos de CPU de la titularidad (garantizados) para la LPAR como nivel de línea base. En aquellos casos en los que los recursos de CPU asignados a la LPAR son mayores que los recursos de CPU de la titularidad, el asignador calcula la utilización de CPU en relación con la cantidad más grande, y cuando la asignación de recursos de CPU es menor que los recursos de CPU de la titularidad, el asignador calcula la utilización de CPU en relación con los recursos de CPU de la titularidad.

**Nota:** El comportamiento del asignador de gestión de carga de trabajo es coherente con la forma en que AIX WLM realiza sus cálculos. Para las LPAR dedicadas en AIX, y para todos los entornos no AIX, la utilización de CPU se calcula en relación con la capacidad de CPU total de los núcleos físicos que están disponibles para el gestor de bases de datos DB2.

### Habilitación de un asignador de gestión de carga de trabajo

Para habilitar el asignador de gestión de carga de trabajo de DB2, establezca el valor del parámetro de configuración del gestor de bases de datos **wlm\_dispatcher** en YES. El asignador de gestión de carga de trabajo gestiona los recursos de CPU asignados a las clases de servicio de usuario y mantenimiento de DB2.

### Procedimiento

Para habilitar el asignador de gestión de carga de trabajo utilizando el procesador de línea de mandatos (CLP):

1. Conéctese a su instancia de DB2
 

```
ATTACH TO nombre_instancia
```
2. Emita el mandato **UPDATE DATABASE MANAGER CONFIGURATION** (o **UPDATE DBM CFG**). Establezca el valor del parámetro de configuración del gestor de bases de datos **wlm\_dispatcher** en YES.
 

```
UPDATE DBM CFG USING wlm_dispatcher yes
```
3. Opcional: Desconéctese de la instancia de DB2.
 

```
DETACH
```

## Resultados

El asignador de gestión de carga de trabajo está habilitado. Ahora puede especificar asignaciones de límite de CPU de los recursos de CPU para las clases de servicio de DB2.

## Qué hacer a continuación

Después de haber habilitado el asignador de gestión de carga de trabajo para que gestione los recursos de CPU, debe considerar la realización de las tareas siguientes:

- Crear o alterar una clase de servicio y configurar el límite de CPU mediante la sentencia `CREATE SERVICE CLASS` o `ALTER SERVICE CLASS`. Consulte: “Establecimiento de un límite de CPU” en la página 229.
- Habilitar las cuotas de CPU fijas y flexibles configurando el parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares`. Consulte en paso 1 en “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Crear o alterar una clase de servicio y configurar una cuota de CPU fija mediante la sentencia `CREATE SERVICE CLASS` o `ALTER SERVICE CLASS`. Consulte: “Habilitación y establecimiento de cuotas de CPU fijas” en la página 207.
- Crear o alterar una clase de servicio y configurar una cuota de CPU flexible mediante la sentencia `CREATE SERVICE CLASS` o `ALTER SERVICE CLASS`. Consulte: “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Establezca el porcentaje de utilización de CPU mínimo para que el asignador de gestión de carga de trabajo pueda considerar que las clases de servicio están activas configurando el parámetro de configuración del gestor de bases de datos `wlm_disp_min_util`. Consulte: “Establecimiento de la utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 233.
- Establezca el nivel de simultaneidad de asignación configurando el parámetro de configuración del gestor de bases de datos `wlm_disp_concur`. Consulte: “Establecimiento del nivel de simultaneidad de asignación” en la página 235.

## Cómo garantizar la máxima precisión de planificación del asignador de gestión de carga de trabajo de DB2

Para garantizar la máxima precisión de planificación para el asignador de gestión de carga de trabajo, el gestor de bases de datos de DB2 intenta asignar prioridades en tiempo real a la hebra de temporizador `db2w1mt` y a la hebra de planificación `db2w1mtm`. Para que la asignación de estas prioridades pueda realizarse correctamente, deben otorgarse determinados permisos al gestor de bases de datos DB2, en función del sistema operativo.

## Procedimiento

Aunque no pueda asignar prioridades en tiempo real a las hebras de temporizador y de planificación, el asignador de gestión de carga de trabajo de DB2 sigue teniendo la capacidad de poder establecer la prioridad del trabajo, pero no con la misma precisión o tan óptimamente como podría hacerlo. A continuación se proporcionan las tareas, ordenadas en función del sistema operativo, que tienen como objetivo garantizar la máxima precisión de planificación para el asignador de gestión de carga de trabajo. Seleccione el sistema operativo que está utilizando y siga las instrucciones que se facilitan.

- En los sistemas operativos AIX, el propietario de la instancia debe tener funciones CAP\_NUMA\_ATTACH y CAP\_PROPAGATE para definir una prioridad relativa más alta para los agentes de una clase de servicio utilizando AGENT PRIORITY. Para otorgar estas capacidades, inicie la sesión como usuario root y ejecute el mandato siguiente:

```
chuser capabilities=CAP_NUMA_ATTACH,CAP_PROPAGATE
```

- En Solaris 10 o posterior, el propietario de la instancia debe disponer del privilegio proc\_priocntl para poder establecer una prioridad relativa más alta para los agentes de una clase de servicio mediante AGENT PRIORITY. Para otorgar este privilegio, debe iniciarse la sesión como root y ejecutar el mandato siguiente:

```
usermod -K defaultpriv=basic,proc_priocntl db2user
```

En este ejemplo, se añade proc\_priocntl al conjunto de privilegios por omisión del usuario db2user.

Además, cuando el gestor de bases de datos DB2 se ejecuta en una zona no global de Solaris, el privilegio proc\_priocntl debe añadirse al conjunto de privilegios de límite de la zona. Para otorgar este privilegio a la zona, debe iniciarse la sesión como root y ejecutar el mandato siguiente:

```
global# zonecfg -z db2zone
zonecfg:db2zone> set limitpriv="default,proc_priocntl"
```

En este ejemplo, se añade proc\_priocntl al conjunto de privilegios de límite de la zona db2zone.

- En Solaris 9, no existe ningún recurso que permita al gestor de bases de datos DB2 incrementar la prioridad relativa de las hebras. Actualice a Solaris 10 o posterior para poder beneficiarse de esta función.

## Cuotas de CPU fijas

El asignador de gestión de carga de trabajo de DB2 puede gestionar los recursos de CPU mediante la utilización de titularidades basadas en cuotas que se asignan a las clases de servicio. Las cuotas de CPU fijas, cuando el administrador asigna éstas a una clase de servicio que contiene trabajo que se visualiza como trabajo de gran impacto o de prioridad más baja, impiden que esa clase de servicio pueda consumir más cuota de recursos de CPU de la que tiene asignada si existe trabajo en otras clases de servicio ejecutándose en el sistema.

### Introducción

Las cuotas de CPU fijas pueden asignarse a cualquier clase de servicio de usuario y de mantenimiento, pero no a la clase de servicio del sistema. Después de habilitar el asignador de gestión de carga de trabajo y de supervisar las cargas de trabajo existentes para determinar el alcance del consumo de recursos de CPU, y después de habilitar el atributo de las cuotas de CPU para las clases de servicio, puede asignar cuotas de CPU fijas a las clases de servicio que considera que son trabajo de baja prioridad en ejecución. El uso de cuotas de CPU fijas garantiza que estas clases de servicio limiten su consumo de CPU en presencia de otras cargas de trabajo, tanto limitando su impacto en el sistema como garantizando que la CPU restante se reserve para otras tareas de más alta prioridad.

En los siguientes apartados se describen más detalladamente las características y funciones de las cuotas de CPU fijas. Se ha incluido un apartado con diversos escenarios de uso que le ayudará a entender las características y funciones de las cuotas de CPU fijas con ejemplos de uso.

## Características y funciones

Cuando el sistema principal o la partición lógica (LPAR) se ejecuta al 100% de utilización de CPU, la asignación de recursos de CPU entre clases de servicio simplemente refleja sus porcentajes de cuota relativos. Por otra parte, cuando el sistema principal o la LPAR empieza a experimentar una ejecución que se sitúa por debajo de la utilización de CPU total, la reasignación de recursos es compleja y depende de si el atributo de las cuotas de CPU de cada clase de servicio activa se ha establecido como cuotas de CPU flexibles o fijas.

Una clase de servicio que tiene asignadas cuotas de CPU fijas no puede exceder su titularidad de recursos de CPU, indicada mediante su configuración de cuotas de CPU, para consumir cualquier recurso de CPU no utilizado que esté disponible en el sistema principal o LPAR. El asignador de gestión de carga de trabajo siempre respeta la titularidad de recursos de CPU, que se determina mediante la cantidad relativa de cuotas de CPU fijas que se han asignado, cuando el trabajo sigue ejecutándose en superclases de servicio que compiten por los recursos o en subclases de servicio que compiten por los recursos dentro de la misma superclase de servicio. Si no existen cargas de trabajo que compiten por los recursos o existe una carga de trabajo que compite por los recursos que temporalmente ha adquirido un estado de total desocupación, las clases de servicio que tienen cuotas de CPU fijas pueden reclamar los recursos de CPU no utilizados.

El establecimiento de cuotas de CPU fijas es más efectivo cuando se utilizan en aquellos casos en los que desea aplicar estrictamente la titularidad de recursos de CPU en una clase de servicio para impedir que el trabajo que se ejecuta en esta clase de servicio pueda interrumpir trabajo más importante que está en ejecución en el sistema principal o LPAR. Asigne cuotas de CPU a las clases de servicio que ejecutan consultas complejas o intensivas que, de otro modo, podrían degradar el rendimiento del trabajo de más alta prioridad debido a la contención por recursos tales como la E/S, la agrupación de almacenamientos intermedios o las antememorias de CPU.

Para habilitar el atributo de las cuotas de CPU, debe establecer el valor del parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares` en YES. El valor por omisión de este parámetro es NO. Después de haber habilitado este parámetro, a todas las clases de servicio nuevas y recientemente creadas se les asignarán 1000 cuotas de CPU fijas por omisión para garantizar inicialmente una distribución equitativa de los recursos de CPU. Puede asignar y ajustar las cuotas de CPU fijas mediante la utilización de las sentencias CREATE SERVICE CLASS y ALTER SERVICE CLASS. Para obtener completa información acerca de cómo habilitar y establecer las cuotas de CPU fijas, consulte: "Habilitación y establecimiento de cuotas de CPU fijas" en la página 207.

En función del número de cuotas de CPU que se asignan a una clase de servicio, el asignador de gestión de carga de trabajo calcula la parte de recursos de CPU para la que cada clase de servicio dispone de titularidad de uso. Para determinar la parte de recursos de CPU para la que cada superclase de servicio dispone de titularidad, puede utilizar la siguiente fórmula para convertir el número de cuotas de CPU de una superclase de servicio en particular en un porcentaje de recursos de CPU que el asignador de gestión de carga de trabajo asigna:

$$\% \text{ CPU}(\text{superclase}) = (\text{Número de cuotas de superclase} / \text{Número total de cuotas de todas las superclases activas}) \times 100$$

Para determinar la parte de recursos de CPU para la que cada subclase de servicio dispone de titularidad, puede utilizar la siguiente fórmula para convertir el

número de cuotas de CPU de una subclase de servicio en particular en un porcentaje de recursos de CPU que el asignador de gestión de carga de trabajo asigna:

$$\% \text{ CPU(subclase)} = \% \text{ CPU(superclase)} \times (\text{Número de cuotas de subclase} / \text{Número total de cuotas de todas las subclases activas en la superclase})$$

**Nota:** Se calcula el número total de cuotas de CPU (fijas y flexibles) de todas las superclases activas de todas las bases de datos y todos los miembros del sistema principal o LPAR.

## Escenarios de uso

### Escenario 1

En el panel A de la Figura 30 en la página 203, a la clase de servicio B se le han asignado cuotas de CPU fijas y a las clases de servicio A y C se les han asignado cuotas de CPU flexibles, cuyas cantidades se describen en la leyenda de la figura. En el gráfico circular se representa la proporción de recursos de CPU asignados para la que cada una de estas clases de servicio activas tiene titularidad, y cada clase de servicio utiliza su cuota completa de recursos de CPU, contabilizándose por lo tanto en este ejemplo una utilización de CPU del 100%. En el panel B, la clase de servicio A no tiene suficiente trabajo para utilizar por completo su titularidad de CPU, que desciende del 60% al 50% de utilización de CPU. El 10% no utilizado de los recursos de CPU, al que temporalmente ha renunciado la clase de servicio A, sólo puede reclamarlo la clase de servicio que compite por los recursos C en función de sus asignaciones de cuotas de CPU flexibles. La clase de servicio B no puede exceder su asignación de recursos de CPU del 30% en este ejemplo porque tiene asignadas cuotas de CPU fijas y no existe suficiente trabajo en ejecución en las clases de servicio A y C para que el asignador considere esas clases de servicio como activas. En el panel C se muestra la reasignación total de recursos de CPU para la clase de servicio C, que experimenta un incremento del 10% al 20% de los recursos de CPU disponibles totales.

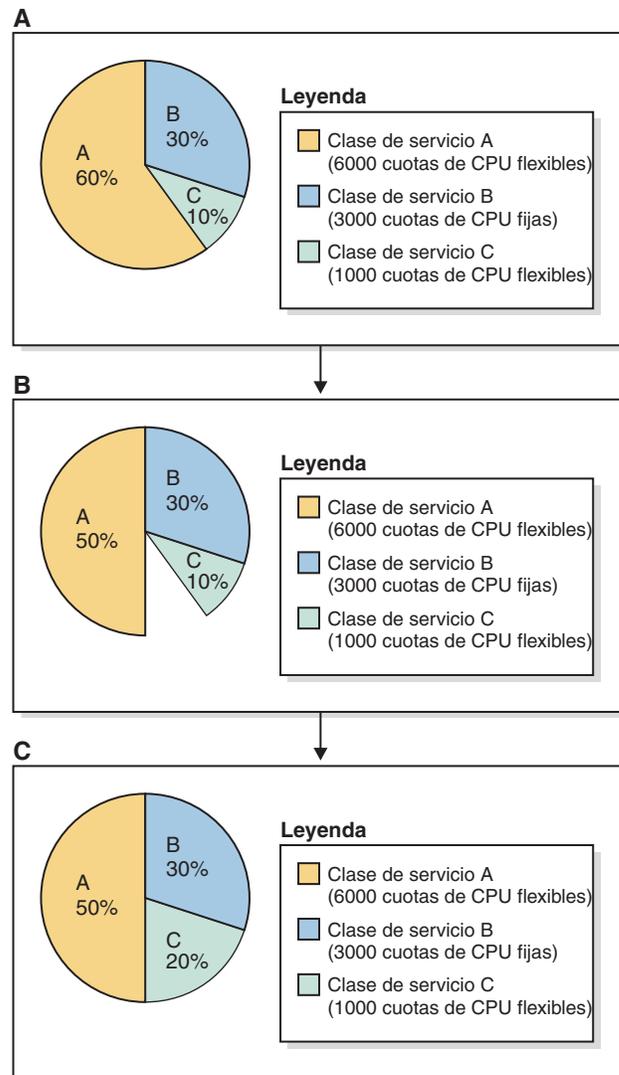


Figura 30. Gráficos circulares de las cuotas de CPU fijas y flexibles: Escenario 1

Si la clase de servicio A experimenta un incremento en su carga de trabajo, ésta incrementa efectivamente su demanda de recursos de CPU. En esta circunstancia, la clase de servicio C renuncia inmediatamente a favor de la clase de servicio A de todos los recursos de CPU reclamados, restaurándose de esta forma el estado de las asignaciones de recursos de CPU que se representaban por medio del gráfico circular del panel A.

**Nota:** Cuando las clases de servicio compiten por el consumo de recursos de CPU, el asignador de gestión de carga de trabajo gestiona las peticiones de las clases de servicio individuales que demandan recursos de CPU por orden, primera en llegar, primer en recibir servicio. Puesto que generalmente las peticiones de recursos de CPU son frecuentes y de corta duración en un sistema principal o LPAR ocupado, la reasignación de los recursos de CPU no utilizados da como resultado, con el transcurso del tiempo, una redistribución uniforme de los recursos de CPU que es proporcional a las asignaciones de cuotas de CPU relativas.

## Escenario 2

El uso de cuotas fijas puede dar como resultado que parte de los recursos de CPU sigan utilizándose por debajo de sus posibilidades en el servidor de bases de datos. La infrautilización de los recursos de CPU puede darse en aquellos casos en los que otras clases de servicio no han estado ejecutando una carga de trabajo lo suficientemente grande como para utilizar por completo sus asignaciones de recursos de CPU. La infrautilización de los recursos de CPU se deseará en aquellos casos en los que las cuotas fijas se utilizan para limitar una carga de trabajo intensiva que podría de otro modo interferir en el progreso de otras clases de servicio, incluso cuando la utilización de los recursos de CPU es completa. Por lo general, esta circunstancia se produce debido a la contención por recursos tales como la E/S o la antememoria de CPU.

En el panel A de la Figura 31 en la página 205, a las clases de servicio B y C se les han asignado cuotas de CPU fijas y a la clase de servicio A se le han asignado cuotas de CPU flexibles, cuyas cantidades se describen en la leyenda de la figura. En el gráfico circular se representa la proporción de recursos de CPU asignados para la que cada una de estas clases de servicio activas tiene titularidad, y cada clase de servicio utiliza su cuota completa de recursos de CPU, contabilizándose por lo tanto en este ejemplo una utilización de CPU del 100%. En el panel B, la clase de servicio A no tiene suficiente trabajo para utilizar por completo su titularidad de CPU, que desciende del 60% al 50% de utilización de CPU. El 10% no utilizado de los recursos de CPU, al que temporalmente ha renunciado la clase de servicio A, no pueden reclamarlo las clases de servicio que compite por los recursos B y C en función de sus asignaciones de cuotas de CPU fijas. Las clases de servicio B y C no pueden exceder sus asignaciones de recursos de CPU del 30% y del 10%, respectivamente, en este ejemplo porque ambas tienen asignadas cuotas de CPU fijas y no existe suficiente trabajo en ejecución en la clase de servicio A para que el asignador considere a ésta como activa (la utilización de CPU cae por debajo del nivel configurado para el parámetro de configuración del gestor de bases de datos `wlm_disp_min_util`, cuyo valor por omisión es del 5%). En el panel C se muestra que los recursos de CPU no utilizados no vuelven a asignarse y que los recursos de CPU siguen utilizándose por debajo de sus posibilidades en este escenario.

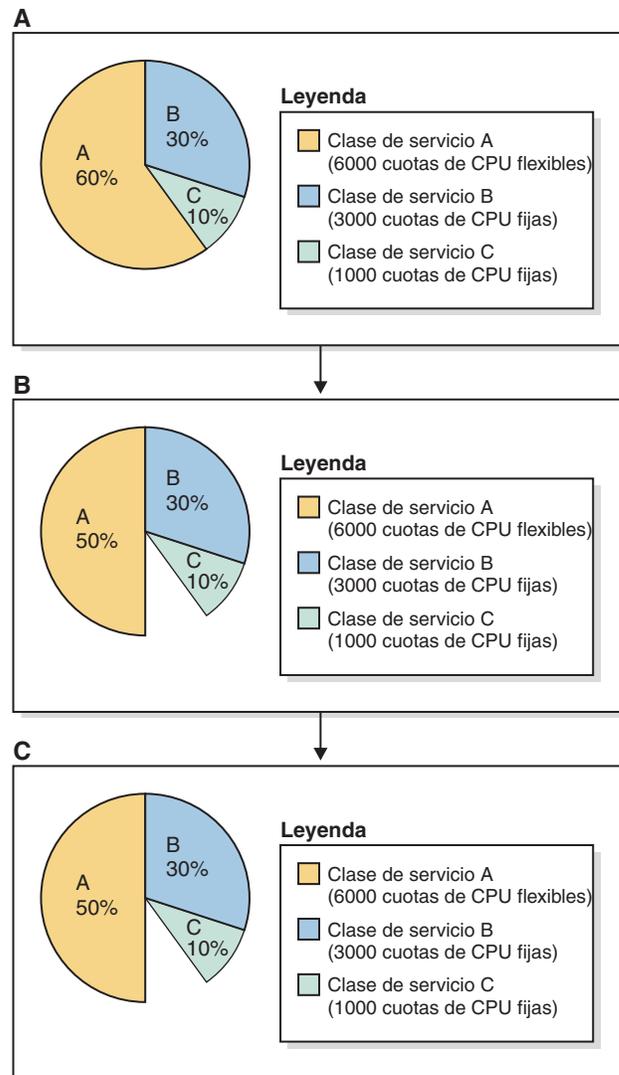


Figura 31. Gráficos circulares de las cuotas de CPU fijas y flexibles: Escenario 2

En este escenario se muestra que puede proteger el progreso del trabajo que está ejecutándose en una clase de servicio de alta prioridad de las interrupciones del trabajo que está ejecutándose en las clases de servicio de baja prioridad.

### Escenario 3

Las cuotas de CPU fijas ofrecen ventajas respecto a los límites de CPU de porcentaje fijo tradicionales. En ausencia de otras cargas de trabajo, la clase de servicio que tiene cuotas de CPU fijas tiene la flexibilidad de poder reclamar recursos de CPU no utilizados. Por lo tanto, una clase de servicio con una asignación de cuotas de CPU fijas no está limitada artificialmente cuando no existe otro trabajo en el sistema principal o LPAR que el que se produce para una clase de servicio con límites de CPU fijos.

En el panel A de la Figura 32 en la página 206, a las clases de servicio B y C se les han asignado cuotas de CPU fijas y a la clase de servicio A se le han asignado cuotas de CPU flexibles, cuyas cantidades se describen en la leyenda de la figura. En el gráfico circular se representa la proporción de recursos de CPU asignados para la que cada una de estas clases de servicio activas tiene titularidad, y cada

clase de servicio utiliza su cuota completa de recursos de CPU, contabilizándose por lo tanto en este ejemplo una utilización de CPU del 100%. En el panel B, la clase de servicio A no tiene ningún trabajo para utilizar por completo su titularidad de CPU, que desciende del 60% al 0% de utilización de CPU. El asignador considera a la clase de servicio A como no activa. El 60% no utilizado de los recursos de CPU, al que temporalmente ha renunciado la clase de servicio A, pueden ahora reclamarlo las clases de servicio que compite por los recursos B y C en función de sus asignaciones de cuotas de CPU fijas y una clase de servicio no activa. En esta circunstancia, las clases de servicio B y C pueden exceder sus asignaciones de recursos de CPU del 30% y del 10%, respectivamente, porque ambas tienen asignadas cuotas de CPU fijas y no existe suficiente trabajo en ejecución en la clase de servicio A para que el asignador considere a ésta como activa (la utilización de CPU cae por debajo del nivel configurado para el parámetro de configuración del gestor de bases de datos `wlm_disp_min_util`, cuyo valor por omisión es del 1%). En el panel C se muestra que a la clase de servicio B se le ha asignado el 75% ( $(3000 / (3000 + 1000)) \times 100$ ) de los recursos de CPU y a la clase de servicio C se le ha asignado el 25% ( $(1000 / (3000 + 1000)) \times 100$ ).

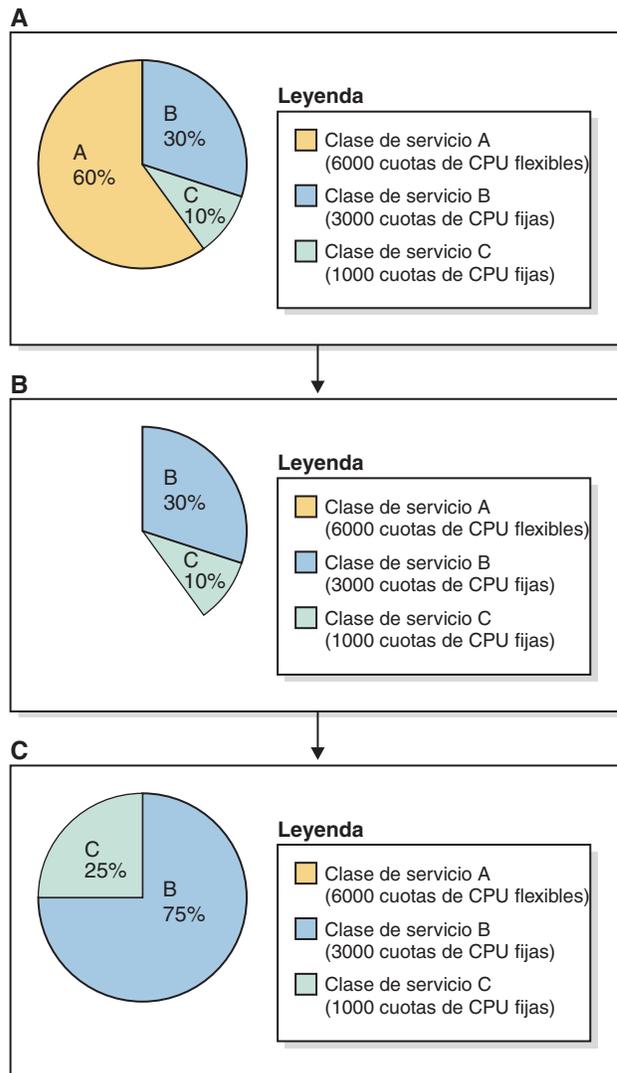


Figura 32. Gráficos circulares de las cuotas de CPU fijas y flexibles: Escenario 3

Si la clase de servicio A experimenta un incremento en su carga de trabajo, ésta incrementa efectivamente su demanda de recursos de CPU. En esta circunstancia, las clases de servicio B y C renuncian inmediatamente a favor de la clase de servicio A de todos los recursos de CPU reclamados, restaurándose de esta forma el estado de las asignaciones de recursos de CPU que se representaban por medio del gráfico circular del panel A.

En este escenario se muestra que, aunque puede proteger el progreso del trabajo que está ejecutándose en las clases de servicio de alta prioridad de las interrupciones del trabajo que está ejecutándose en las clases de servicio de baja prioridad, cuando ya no existe trabajo de alta prioridad (como durante el horario de trabajo de menor actividad), las clases de servicio de baja prioridad con cuotas de CPU fijas todavía tendrán la flexibilidad de poder reclamar los recursos de CPU no utilizados.

**Nota:** Si la clase de servicio B o la clase de servicio C no utiliza su asignación de recursos de CPU total y el asignador sigue considerando a ésta como trabajo en ejecución activo, la otra clase de servicio con cuotas de CPU fijas no puede beneficiarse de ello y utilizar más que lo que tiene asignado.

### Información adicional

Se proporciona completa información acerca de los siguientes temas del asignador de gestión de carga de trabajo:

- Asignador de gestión de carga de trabajo, consulte: “Asignador de gestión de carga de trabajo” en la página 177.
- Cuotas de CPU flexibles, consulte: “Cuotas de CPU flexibles” en la página 209.
- Límite de CPU, consulte: “Límite de CPU” en la página 215.
- Nivel de utilización de CPU mínimo para que se considere activa una clase de servicio, consulte: “Utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 230.
- Nivel de simultaneidad de asignación, consulte: “Nivel de simultaneidad de asignación” en la página 234.
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo” en la página 237.

### Habilitación y establecimiento de cuotas de CPU fijas

Habilite los atributos de las cuotas de CPU estableciendo el valor del parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares` en YES. Establezca el atributo de las cuotas de CPU fijas mediante las sentencias CREATE SERVICE CLASS o ALTER SERVICE CLASS. Por lo general deberá especificar cuotas de CPU fijas en las clases de servicio que ejecutan tareas de baja prioridad o de gran impacto.

### Antes de empezar

**Nota:** Para que el valor de las cuotas de CPU fijas sea efectivo, debe habilitar el asignador de gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos `wlm_dispatcher`. Esto puede realizarse antes o después de haber habilitado y establecido las cuotas de CPU fijas. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.

## Acerca de esta tarea

En esta tarea, los atributos de las cuotas de CPU se habilitan mediante el método del procesador de línea de mandatos.

En esta tarea, el atributo de las cuotas de CPU fijas se establece para una clase de servicio mediante la sentencia `CREATE SERVICE CLASS` o `ALTER SERVICE CLASS`, por medio del procesador de línea de mandatos.

### Restricciones

Puede asignar cuotas de CPU flexibles sólo a las clases de servicio que el asignador de gestión de carga de trabajo pueda controlar, que son las clases de servicio de mantenimiento y de usuario. El asignador de gestión de carga de trabajo no puede controlar las asignaciones de recursos de CPU para las clases de servicio del sistema.

## Procedimiento

Vaya al paso 2 si los atributos de las cuotas de CPU ya se han habilitado.

Para habilitar los atributos de las cuotas de CPU:

1. Mediante la utilización del procesador de línea de mandatos (CLP), emita el mandato **UPDATE DATABASE MANAGER CONFIGURATION** (o **UPDATE DBM CFG**) y establezca el valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_cpu\_shares** en YES; el valor del parámetro se actualiza inmediatamente estableciendo primero una conexión con su instancia de DB2:

```
attach to nombre_instancia
update dbm cfg using wlm_disp_cpu_shares yes
detach
```

Vaya al paso 3 si desea alterar un valor de cuotas de CPU fijas de una clase de servicio existente.

Para crear una nueva clase de servicio y establecer el atributo de las cuotas de CPU fijas:

2. Emita la sentencia `CREATE SERVICE CLASS` para crear una nueva clase de servicio y establecer el valor de las cuotas de CPU fijas en 5000 en este ejemplo:  

```
create service class nombre_clase_servicio hard cpu shares 5000
```

Para alterar el valor de las cuotas de CPU fijas para una clase de servicio existente:

3. Emita la sentencia `ALTER SERVICE CLASS` para alterar el valor de las cuotas de CPU fijas por 15000 en este ejemplo:  

```
alter service class nombre_clase_servicio hard cpu shares 15000
```

## Resultados

Los atributos de las cuotas de CPU se han habilitado y ha establecido o alterado el valor de las cuotas de CPU fijas para la clase de servicio especificada. Las cuotas de CPU fijas asignadas a las clases de servicio pueden reclamar, bajo condiciones restringidas, recursos de CPU a otras clases de servicio que han renunciado a éstos debido a su menor demanda de carga de trabajo.

## Qué hacer a continuación

Después de haber habilitado el asignador de gestión de carga de trabajo para que gestione los recursos de CPU y después de haber establecido los atributos de las cuotas de CPU y el atributo de las cuotas de CPU fijas, considere la realización de las tareas siguientes:

- Para que el valor de las cuotas de CPU fijas sea efectivo, debe habilitar el asignador de gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos **wlm\_dispatcher**. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.
- Establezca el nivel de simultaneidad de asignación configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**. Consulte: “Establecimiento del nivel de simultaneidad de asignación” en la página 235.
- Crear o alterar una clase de servicio y configurar una cuota de CPU flexible mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Crear o alterar una clase de servicio y configurar el límite de CPU mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Establecimiento de un límite de CPU” en la página 229.
- Establezca el porcentaje de utilización de CPU mínimo para que el asignador de gestión de carga de trabajo pueda considerar que las clases de servicio están activas configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util**. Consulte: “Establecimiento de la utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 233.

## Cuotas de CPU flexibles

El asignador de gestión de carga de trabajo de DB2 puede gestionar los recursos de CPU mediante la utilización de titularidades basadas en cuotas que se asignan a las clases de servicio. Las cuotas de CPU flexibles, cuando el administrador asigna éstas a una clase de servicio, permiten a esa clase de servicio consumir más cuota de recursos de CPU de la que tiene asignada si éstas no se utilizan. Cuando se utilizan junto con otras clases de servicio limitadas mediante cuotas de CPU fijas, esto proporciona un tratamiento de preferencia a esa clase de servicio con relación a los recursos de CPU.

### Introducción

Las cuotas de CPU flexibles pueden asignarse a cualquier clase de servicio de usuario y de mantenimiento, pero no a la clase de servicio del sistema. Después de habilitar el asignador de gestión de carga de trabajo y de supervisar las cargas de trabajo existentes para determinar el alcance del consumo de recursos de CPU, puede asignar cuotas de CPU flexibles a las clases de servicio que se consideren de alta prioridad. Las cuotas de CPU flexibles alcanzan su máxima eficacia cuando se utilizan en cargas de trabajo de prioridad alta, porque permiten que una carga de trabajo consuma más que su titularidad especificada en caso de que haya recursos de CPU inactivos en el sistema. No se recomienda el uso de cuotas de CPU flexibles si desea restringir el consumo de CPU de las tareas de prioridad más baja o de gran impacto; en ese caso, utilice cuotas de CPU fijas.

En los siguientes apartados se describen más detalladamente las características y funciones de las cuotas de CPU flexibles. Se ha incluido un apartado con diversos escenarios de uso que le ayudará a entender las características y funciones de las cuotas de CPU flexibles con ejemplos de uso.

## Características y funciones

Cuando el sistema principal o la partición lógica (LPAR) se ejecuta al 100% de utilización de CPU, la asignación de recursos de CPU entre clases de servicio simplemente refleja sus porcentajes de cuota relativos. Por otra parte, cuando el sistema principal o la LPAR empieza a experimentar una ejecución que se sitúa por debajo de la utilización de CPU total, la reasignación de recursos es compleja y depende de si el atributo de las cuotas de CPU de cada clase de servicio activa se ha establecido como cuotas de CPU flexibles o fijas.

Una clase de servicio que tiene asignadas cuotas de CPU flexibles puede exceder su titularidad de recursos de CPU, indicada mediante su configuración de cuotas de CPU, para consumir cualquier recurso de CPU no utilizado que esté disponible en el sistema principal o LPAR. Cuando dos o más clases de servicio tienen cuotas flexibles y pasan a estar disponibles recursos de CPU no utilizados con suficiente demanda de recursos de CPU procedente de cada clase de servicio para consumir la capacidad de reserva, la asignación de los recursos de CPU a las clases de servicio que compiten por los recursos se realiza de forma proporcional de acuerdo con la cuota relativa de cada clase de servicio activa. El establecimiento de cuotas de CPU flexibles es más efectivo para el trabajo de alta prioridad que desea que pueda reclamar temporalmente cualquier recurso de CPU de reserva que pase a estar disponible. Asimismo, el establecimiento de cuotas de CPU flexibles es más efectivo para las cargas de trabajo que constan de consultas cortas que se espera que tendrán un impacto relativamente modesto en los recursos de la base de datos, fuera de su ámbito de consumo de CPU inmediato.

Para habilitar el atributo de las cuotas de CPU, debe establecer el valor del parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares` en YES. El valor por omisión de este parámetro es NO. Puede asignar y ajustar las cuotas de CPU flexibles mediante las sentencias CREATE SERVICE CLASS y ALTER SERVICE CLASS. Para obtener completa información acerca de cómo habilitar y establecer las cuotas de CPU flexibles, consulte: "Habilitación y establecimiento de cuotas de CPU flexibles" en la página 213.

En función del número de cuotas de CPU que se asignan a una clase de servicio, el asignador de gestión de carga de trabajo calcula la parte de recursos de CPU para la que cada clase de servicio dispone de titularidad de uso. Para determinar la parte de recursos de CPU para la que cada superclase de servicio dispone de titularidad, puede utilizar la siguiente fórmula para convertir el número de cuotas de CPU de una superclase de servicio en particular en un porcentaje de recursos de CPU que el asignador de gestión de carga de trabajo asigna:

$$\% \text{ CPU}(\text{superclase}) = (\text{Número de cuotas de superclase} / \text{Número total de cuotas de todas las superclases activas}) \times 100$$

Para determinar la parte de recursos de CPU para la que cada subclase de servicio dispone de titularidad, puede utilizar la siguiente fórmula para convertir el número de cuotas de CPU de una subclase de servicio en particular en un porcentaje de recursos de CPU que el asignador de gestión de carga de trabajo asigna:

$$\% \text{ CPU}(\text{subclase}) = \% \text{ CPU}(\text{superclase}) \times (\text{Número de cuotas de subclase} / \text{Número total de cuotas de todas las subclases activas en la superclase})$$

**Nota:** Se calcula el número total de cuotas de CPU (fijas y flexibles) de todas las superclases activas de todas las bases de datos y todos los miembros del sistema principal o LPAR.

## Escenario de uso

En el panel A de la Figura 33 en la página 212, a las clases de servicio A, B y C se les han asignado cuotas de CPU flexibles, cuyas cantidades se describen en la leyenda de la figura. En el gráfico circular se representa la proporción de recursos de CPU asignados para la que cada una de estas clases de servicio activas tiene titularidad, y cada clase de servicio utiliza su cuota completa de recursos de CPU, contabilizándose por lo tanto en este ejemplo una utilización de CPU del 100%. En el panel B, la clase de servicio A no tiene suficiente trabajo para utilizar por completo su titularidad de CPU, que desciende del 60% al 50% de utilización de CPU. El 10% no utilizado de los recursos de CPU, al que temporalmente ha renunciado la clase de servicio A, pueden reclamarlo proporcionalmente las clases de servicio que compiten por los recursos B y C en función de sus asignaciones de cuotas de CPU flexibles relativas. En el panel C se representa la reasignación proporcional de recursos de CPU entre las clases de servicio B y C, donde la clase de servicio B obtiene el 7,5% ( $10\% \times (3000/4000)$ ) y la clase de servicio C obtiene el 2,5% ( $10\% \times (1000/4000)$ ) del 10% no utilizado al que ha renunciado la clase de servicio A.

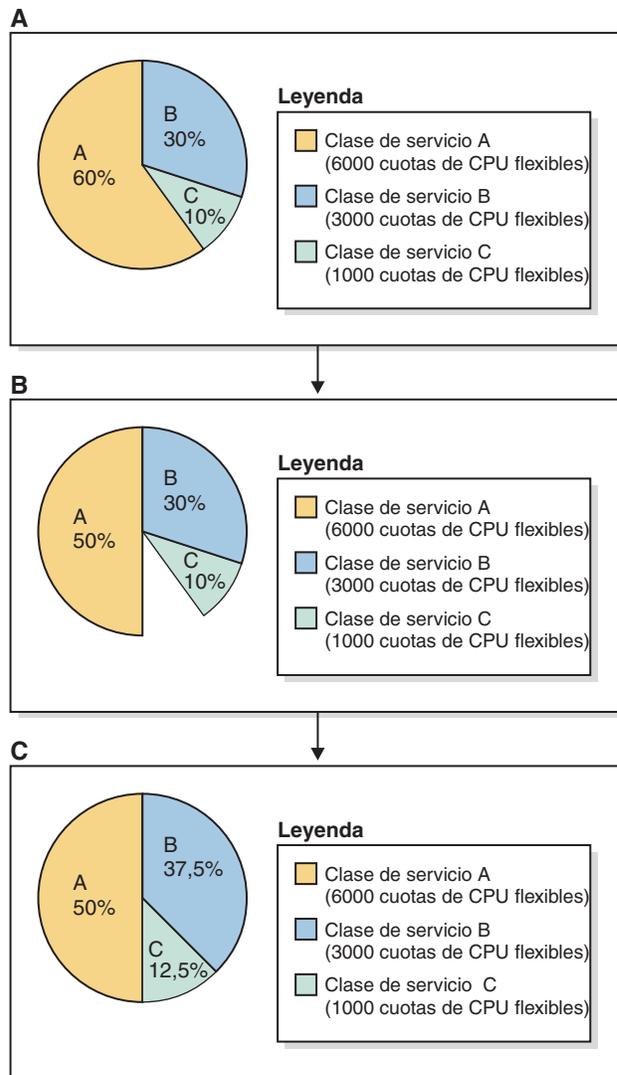


Figura 33. Gráficos circulares de las cuotas de CPU flexibles

Si la clase de servicio A experimenta un incremento en su carga de trabajo, ésta incrementa efectivamente su demanda de recursos de CPU. En esta circunstancia, las clases de servicio B y C renuncian inmediatamente a favor de la clase de servicio A de todos los recursos de CPU reclamados, restaurándose de esta forma el estado de las asignaciones de recursos de CPU que se representaban por medio del gráfico circular del panel A.

**Nota:** Cuando las clases de servicio compiten por el consumo de recursos de CPU, el asignador de gestión de carga de trabajo gestiona las peticiones de las clases de servicio individuales que demandan recursos de CPU por orden, primera en llegar, primer en recibir servicio. Puesto que generalmente las peticiones de recursos de CPU son frecuentes y de corta duración en un sistema principal o LPAR ocupado, la reasignación de los recursos de CPU no utilizados da como resultado, con el transcurso del tiempo, una redistribución uniforme de los recursos de CPU que es proporcional a las asignaciones de cuotas de CPU relativas.

## Información adicional

Se proporciona completa información acerca de los siguientes temas del asignador de gestión de carga de trabajo:

- Asignador de gestión de carga de trabajo, consulte: “Asignador de gestión de carga de trabajo” en la página 177.
- Cuotas de CPU fijas, consulte: “Cuotas de CPU fijas” en la página 200.
- Límite de CPU, consulte: “Límite de CPU” en la página 215.
- Nivel de utilización de CPU mínimo para que se considere activa una clase de servicio, consulte: “Utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 230.
- Nivel de simultaneidad de asignación, consulte: “Nivel de simultaneidad de asignación” en la página 234.
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo” en la página 237.

## Habilitación y establecimiento de cuotas de CPU flexibles

Habilite los atributos de las cuotas de CPU estableciendo el valor del parámetro de configuración del gestor de bases de datos `wlm_disp_cpu_shares` en YES. Establezca el atributo de las cuotas de CPU flexibles mediante las sentencias CREATE SERVICE CLASS o ALTER SERVICE CLASS. Puede establecer las cuotas de CPU flexibles en las clases de servicio de alta prioridad cuando existe restricción de recursos de CPU.

## Antes de empezar

**Nota:** Para que el valor de las cuotas de CPU flexibles sea efectivo, debe habilitar el asignador de gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos `wlm_dispatcher`. Esto puede realizarse antes o después de haber habilitado y establecido las cuotas de CPU flexibles. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.

## Acerca de esta tarea

En esta tarea, los atributos de las cuotas de CPU se habilitan mediante el método del procesador de línea de mandatos.

En esta tarea, el atributo de las cuotas de CPU flexibles se establece para una clase de servicio mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS, por medio del procesador de línea de mandatos.

### Restricciones

Puede asignar cuotas de CPU flexibles sólo a las clases de servicio que el asignador de gestión de carga de trabajo pueda controlar, que son las clases de servicio de mantenimiento y de usuario. El asignador de gestión de carga de trabajo no puede controlar las asignaciones de recursos de CPU para las clases de servicio del sistema.

## Procedimiento

Vaya al paso 2 si los atributos de las cuotas de CPU ya se han habilitado. Para habilitar los atributos de las cuotas de CPU:

1. Mediante la utilización del procesador de línea de mandatos (CLP), emita el mandato **UPDATE DATABASE MANAGER CONFIGURATION** (o **UPDATE DBM CFG**) y establezca el valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_cpu\_shares** en YES; el valor del parámetro se actualiza inmediatamente estableciendo primero una conexión con su instancia de DB2:

```
attach to nombre_instancia
update dbm cfg using wlm_disp_cpu_shares yes
detach
```

Vaya al paso 3 si desea alterar un valor de cuotas de CPU flexibles de una clase de servicio existente.

Para crear una nueva clase de servicio y establecer el atributo de las cuotas de CPU flexibles:

2. Emita la sentencia **CREATE SERVICE CLASS** para crear una nueva clase de servicio y establecer el valor de las cuotas de CPU flexibles en 5000 en este ejemplo:

```
create service class nombre_clase_servicio soft cpu shares 5000
```

Para alterar el valor de las cuotas de CPU flexibles para una clase de servicio existente:

3. Emita la sentencia **ALTER SERVICE CLASS** para alterar el valor de las cuotas de CPU flexibles por 15000 en este ejemplo:

```
alter service class nombre_clase_servicio soft cpu shares 15000
```

## Resultados

Los atributos de las cuotas de CPU se han habilitado y ha establecido o alterado el valor de las cuotas de CPU flexibles para la clase de servicio especificada. Las cuotas de CPU flexibles asignadas a las clases de servicio pueden reclamar recursos de CPU a otras clases de servicio que han renunciado a éstos debido a su menor demanda de carga de trabajo.

**Nota:** Las cuotas de CPU flexibles son más efectivas cuando se han asignado a clases de servicio de alta prioridad y las clases de servicio de alta prioridad se han protegido de las interrupciones de las clases de servicio de baja prioridad a las que se han asignado cuotas de CPU fijas.

## Qué hacer a continuación

Después de haber habilitado los atributos de las cuotas de CPU y después de haber establecido el atributo de las cuotas de CPU flexibles, considere la realización de las tareas siguientes:

- Para que el valor de las cuotas de CPU flexibles sea efectivo, debe habilitar el asignador de gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos **wlm\_dispatcher**. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.
- Establezca el nivel de simultaneidad de asignación configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**. Consulte: “Establecimiento del nivel de simultaneidad de asignación” en la página 235.
- Crear o alterar una clase de servicio y configurar una cuota de CPU fija mediante la sentencia **CREATE SERVICE CLASS** o **ALTER SERVICE CLASS**. Consulte: “Habilitación y establecimiento de cuotas de CPU fijas” en la página 207.

- Crear o alterar una clase de servicio y configurar el límite de CPU mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Establecimiento de un límite de CPU” en la página 229.
- Establezca el porcentaje de utilización de CPU mínimo para que el asignador de gestión de carga de trabajo pueda considerar que las clases de servicio están activas configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util**. Consulte: “Establecimiento de la utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 233.

## Límite de CPU

El asignador de gestión de carga de trabajo de DB2 puede forzar las titularidades de límites de CPU fijos que pueden asignarse a las superclases y a las subclases de servicio. Al aplicar un límite de CPU, puede limitar la CPU que consume una clase de servicio a una cantidad fija en el sistema, independientemente de otros trabajos que estén en ejecución en el gestor de bases de datos de DB2. Esto hace que el resto de los recursos de la CPU estén disponibles para otros consumidores que los necesiten. Cuando los límites de CPU se utilizan junto con las cuotas de CPU, siempre se recurre a la condición más limitante o restrictiva.

### Introducción

Un límite de CPU puede asignarse a cualquier clase de servicio de usuario y de mantenimiento, pero no a la clase de servicio del sistema. Después de haber habilitado el asignador de gestión de carga de trabajo y de haber supervisado las cargas de trabajo existentes para determinar el alcance del consumo de recursos de CPU, puede asignar límites de CPU a las clases de servicio sobre las que desea que haya un límite estricto en cuanto a consumo de CPU.

Las cuotas de CPU le permiten controlar la titularidad de recursos de CPU de cargas de trabajo individuales cuando la carga de trabajo global del sistema principal o LPAR es considerable y sin embargo no gastar recursos de CPU cuando la carga de trabajo global es ligera. No obstante, existen cargas de trabajo cuya titularidad de recursos de CPU siempre deseará limitar, a pesar de que la carga de trabajo global del sistema principal o LPAR sea ligera. Por ejemplo, si varios departamentos comparten el coste de la compra de un servidor de bases de datos, puede que cada departamento desee asegurarse de que los demás departamentos no hacen uso de más titularidades de recursos de CPU que las que tienen asignadas, a pesar de que exista la posibilidad de que la configuración seleccionada dé como resultado una infrautilización de los recursos de CPU del sistema principal o LPAR. Las cuotas de CPU no le proporcionan este grado de control, mientras que un límite de CPU sí lo hace.

En los siguientes apartados se describen más detalladamente las características y funciones del límite de CPU. Se ha incluido un apartado con diversos escenarios de uso que le ayudará a entender las características y funciones del límite CPU con ejemplos de uso.

### Características y funciones

El límite de CPU le permite establecer un límite fijo para la titularidad de recursos de CPU por trabajo en una clase de servicio. Si se establece un límite de CPU en todas las clases de servicio, puede conservar una parte de los recursos de CPU para realizar trabajo con independencia de cualquier otro trabajo que esté en ejecución en el gestor de bases de datos DB2. La configuración de un límite de CPU en una clase de servicio efectivamente proporciona un recinto de seguridad

para las cargas de trabajo en las que puede conseguir un equilibrio en el consumo de recursos de CPU entre cargas de trabajo, pero a expensas de no conseguir ocasionalmente una utilización total de los recursos de CPU.

El límite de CPU también es útil cuando se dispone de varias instancias de DB2 en el sistema principal o LPAR. Cuando el asignador de gestión de carga de trabajo se utiliza en el ámbito de la instancia, la asignación de recursos de CPU de cualquier clase de servicio se calcula a partir de las cuotas de esa clase de servicio en relación con las cuotas de todas las demás clases de servicio de la instancia. Sin embargo, un límite de CPU se expresa en forma de porcentaje de los recursos de CPU del sistema principal o LPAR, con independencia de la cantidad de instancias de DB2 que existan en ese sistema principal o LPAR. Mediante la aplicación de límites de CPU a las superclases y de cuotas a las subclases, puede utilizar el límite de CPU para controlar la titularidad de recursos de CPU absoluta de cada superclase, y por extensión la instancia, y utilizar cuotas para controlar las titularidades de recursos de CPU relativas de las subclases de servicio que se ejecutan en esas superclases.

Un límite de CPU puede configurarse en el ámbito de la superclase de servicio, donde representa un límite de porcentaje de titularidad de recursos de CPU en el sistema principal o LPAR por medio de todas las subclases de esa superclase, o bien en el ámbito de la subclase, donde representa un límite de porcentaje de titularidad de recursos de CPU en el sistema o LPAR por medio de esa subclase en particular.

Para habilitar el atributo del límite de CPU, debe habilitar el asignador de gestión de carga de trabajo estableciendo el valor del parámetro de configuración del gestor de bases de datos `wlm_dispatcher` en ON. El valor por omisión de este parámetro es OFF. Mediante la habilitación del asignador de gestión de carga de trabajo, el control de los recursos de CPU por medio del atributo del límite de CPU pasa a estar disponible por omisión. Puede asignar y ajustar los límites de CPU mediante la utilización de las sentencias CREATE SERVICE CLASS y ALTER SERVICE CLASS. Para obtener completa información acerca de cómo establecer límites de CPU, consulte: “Establecimiento de un límite de CPU” en la página 229.

El asignador de gestión de carga de trabajo siempre respeta las asignaciones de límite de CPU o de cuotas de CPU más restrictivas al asignar recursos de CPU a las clases de servicio. Por ejemplo, cuando se ha establecido un límite de CPU en el ámbito de la superclase y en el ámbito de la subclase, se aplica el límite de CPU más restrictivo. De forma similar, si una clase de servicio alcanza su límite de CPU antes de que haya utilizado toda su titularidad de recursos de CPU basados en cuotas, el asignador respeta el límite de CPU.

## Escenarios de uso

### Límite de CPU y varias superclases

En este conjunto de ejemplos de uso se expone el comportamiento del límite de CPU en un entorno de varias superclases.

En la Figura 34 en la página 217 se muestra un sistema principal o LPAR que se ha configurado con dos superclases, A y B. Para que la ilustración pueda servir de ayuda para describir los conceptos básicos, imaginemos que existe trabajo en ejecución inapreciable en las clases de servicio de usuario, mantenimiento y

sistema por omisión. Para los siguientes escenarios, sólo existe una única instancia de DB2 con una base de datos y un único miembro en este sistema principal o LPAR.

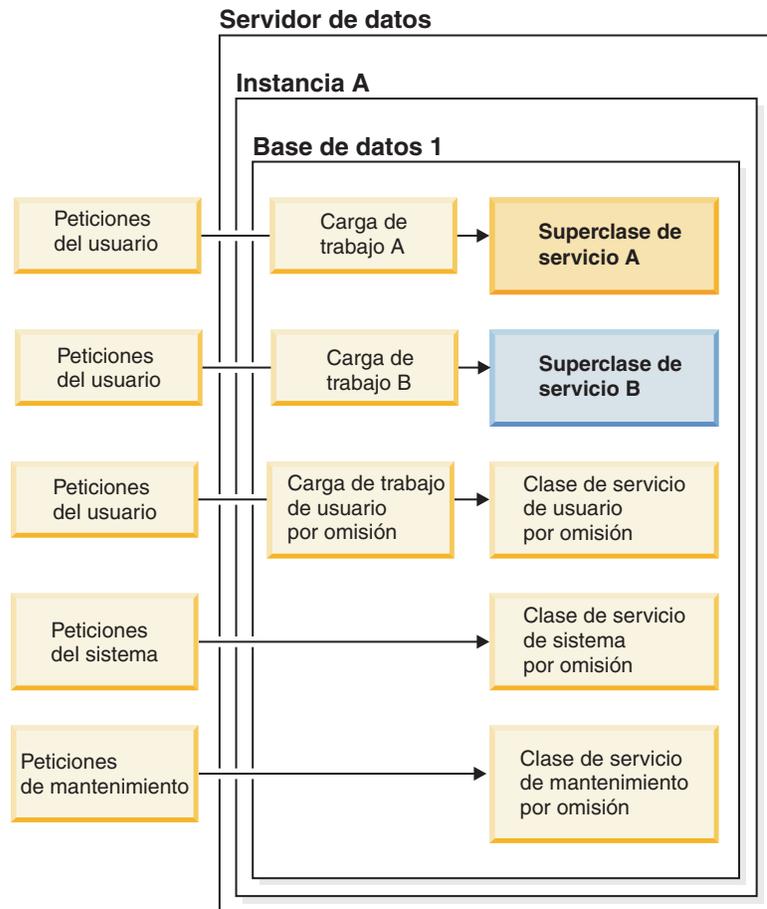


Figura 34. Configuración del servidor de datos: varias superclases

### Límite de CPU y varias superclases: Escenario 1

En este ejemplo, el panel A de la Figura 35 en la página 218 muestra que la clase de servicio A tiene un límite de CPU del 30% y que la clase de servicio B no tiene ningún límite de CPU. Al principio de este escenario, la clase de servicio A tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 30% y la clase de servicio B tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 70%. Las clases de servicio A y B tienen ambas el valor por omisión de 1000 cuotas de CPU flexibles.

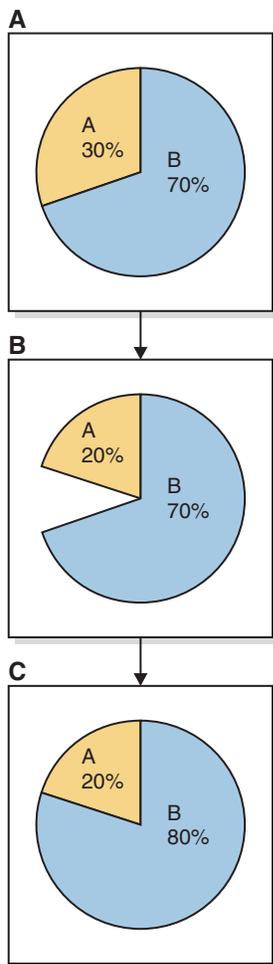


Figura 35. Límite de CPU y varias superclases: Escenario 1

En el panel B se muestra que la clase de servicio A experimenta una reducción en la demanda de recursos de CPU del 30% al 20%. La clase de servicio B tiene una demanda de recursos de CPU más que suficiente para reclamar los recursos de CPU a los que temporalmente ha renunciado la clase de servicio A, tal como se muestra en el panel C.

#### Límite de CPU y varias superclases: Escenario 2

En este ejemplo, el panel A de la Figura 36 en la página 219 muestra nuevamente que la clase de servicio A tiene un límite de CPU del 30% y que la clase de servicio B no tiene ningún límite de CPU. Al principio de este escenario, la clase de servicio A tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 30% y la clase de servicio B tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 70%. Las clases de servicio A y B tienen ambas el valor por omisión de 1000 cuotas de CPU flexibles.

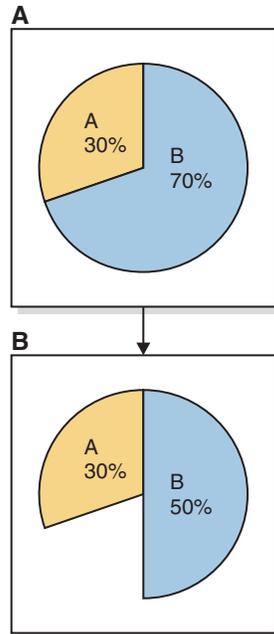


Figura 36. Límite de CPU y varias superclases: Escenario 2

En el panel B se muestra que la clase de servicio B experimenta una reducción en la demanda de recursos de CPU del 70% al 50%. Debido a su límite de CPU, la clase de servicio A no puede consumir el 20% de los recursos de CPU a los que ha renunciado temporalmente la clase de servicio B. La utilización de CPU total del sistema principal o LPAR sigue siendo el 80%.

### Límite de CPU y varias superclases: Escenario 3

En este ejemplo, el panel A de la Figura 37 en la página 220 muestra nuevamente que la clase de servicio A tiene un límite de CPU del 30% y que la clase de servicio B no tiene ningún límite de CPU. Al principio de este escenario, la clase de servicio A tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 30% y la clase de servicio B tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 70%. Las clases de servicio A y B tienen ambas el valor por omisión de 1000 cuotas de CPU flexibles.

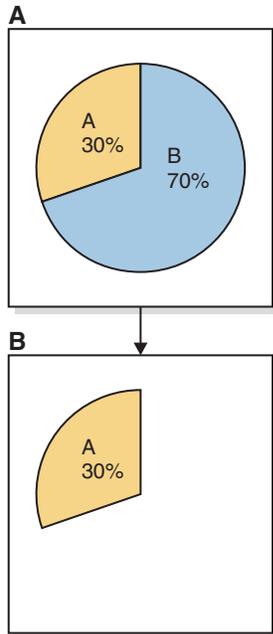


Figura 37. Límite de CPU y varias superclases: Escenario 3

En el panel B se muestra que la clase de servicio B experimenta una reducción en la demanda de recursos de CPU del 70% al 0%. Debido a su límite de CPU, la clase de servicio A no puede consumir el 70% de los recursos de CPU a los que ha renunciado temporalmente la clase de servicio B. La utilización de CPU total del sistema principal o LPAR sigue siendo el 30%.

### Límite de CPU y varias subclases

En el siguiente conjunto de ejemplos de uso se expone el comportamiento del límite de CPU en un entorno de varias subclases.

En la Figura 38 en la página 221 se muestra un sistema principal o LPAR que se ha configurado con dos superclases, A y B. La superclase de servicio A contiene las subclases de servicio A1 y A2. Para que la ilustración pueda servir de ayuda para describir los conceptos básicos, imaginemos que existe trabajo en ejecución inapreciable en las clases de servicio de usuario, mantenimiento y sistema por omisión. Para los siguientes escenarios, sólo existe una única instancia de DB2 con una base de datos y una única partición lógica en este sistema principal o LPAR.

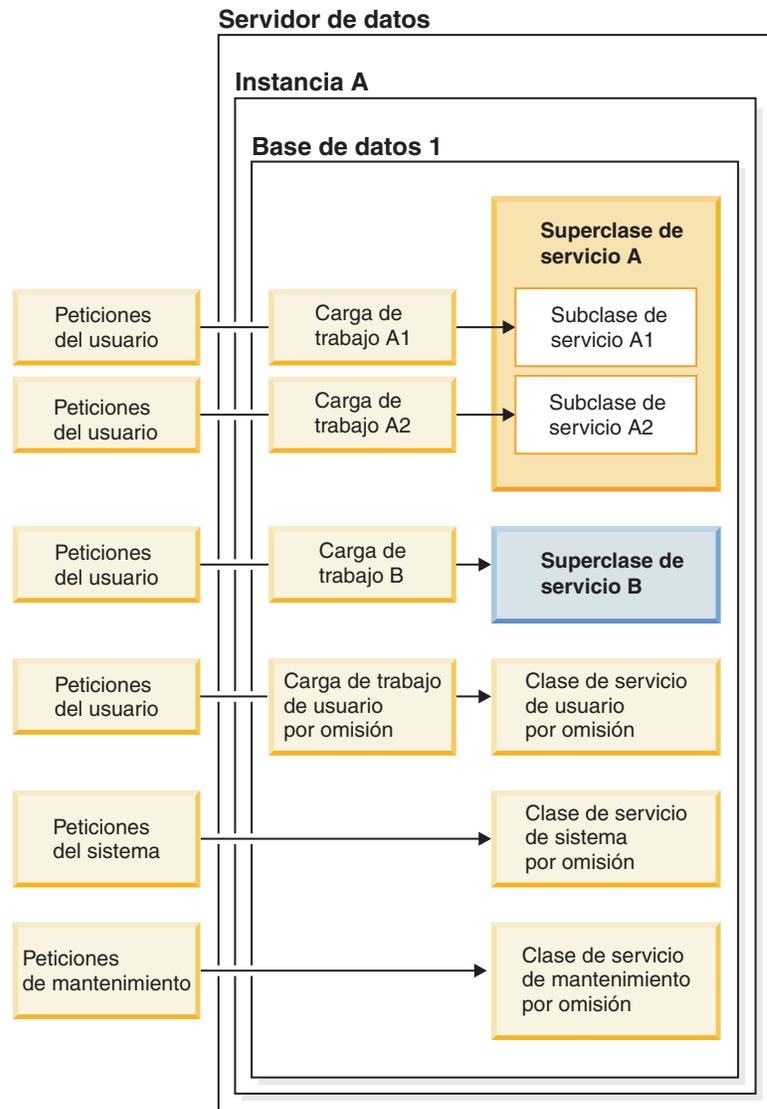


Figura 38. Configuración del servidor de datos: varias subclases

### Límite de CPU y varias subclases: Escenario 1

En este ejemplo, el panel A de la Figura 39 en la página 222 muestra que la superclase de servicio A tiene un límite de CPU del 50% y que la subclase de servicio A1 tiene un límite de CPU del 20%. La superclase de servicio B no tiene ningún límite de CPU. Al principio de este escenario, la subclase de servicio A1 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 20% y la subclase de servicio A2 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 30%, con lo que la superclase de servicio A dispone de una utilización de recursos de CPU total del 50%. La superclase de servicio B tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 50%. Las superclases de servicio A y B tienen ambas el valor por omisión de 1000 cuotas de CPU flexibles.

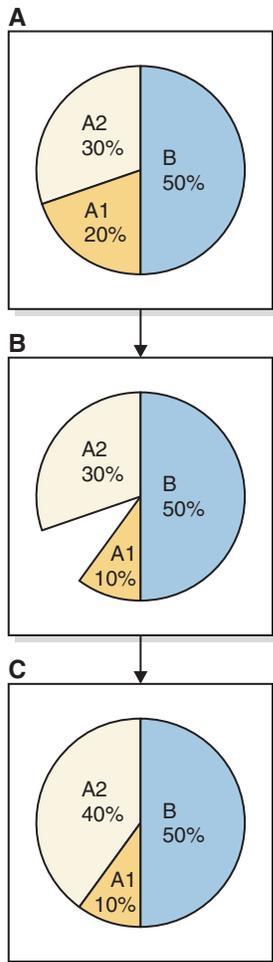


Figura 39. Límite de CPU y varias subclases: Escenario 1

En el panel B se muestra que la subclase de servicio A1 experimenta una reducción en la demanda de recursos de CPU del 20% al 10%. La subclase de servicio A2 y la superclase de servicio B tienen, cada una de ellas, una demanda de recursos de CPU más que suficiente para reclamar toda la cantidad de recursos de CPU no utilizados a los que temporalmente ha renunciado la clase de servicio A1. Sin embargo, la subclase de servicio A2 tiene asignados todos los recursos de CPU a los que se ha renunciado e incrementa su utilización de CPU del 30% al 40%, tal como se representa en el panel C. Este resultado de asignación de recursos de CPU se debe a que las superclases de servicio A y B ya están compartiendo los recursos de CPU totales con una división equitativa de 50%/50% resultante de las 1000 cuotas de CPU flexibles que se han asignado a cada una de las superclases.

#### Límite de CPU y varias subclases: Escenario 2

En este ejemplo, el panel A de la Figura 40 en la página 223 muestra nuevamente que la superclase de servicio A tiene un límite de CPU del 50% y que la subclase de servicio A1 tiene un límite de CPU del 20%. La superclase de servicio B no tiene ningún límite de CPU. Al principio de este escenario, la subclase de servicio A1 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 20% y la subclase de servicio A2 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU al 30%, con lo que la superclase de servicio A dispone de una utilización de recursos de CPU total del 50%. La superclase de servicio B tiene, como mínimo, suficiente trabajo para llevar su

utilización de recursos de CPU al 50%. Las superclases de servicio A y B tienen ambas el valor por omisión de 1000 cuotas de CPU flexibles.

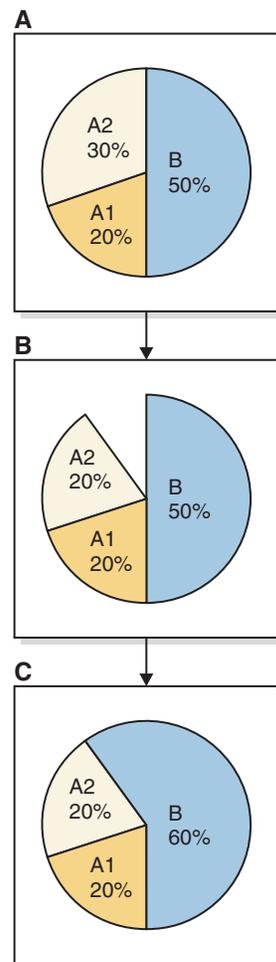


Figura 40. Límite de CPU y varias subclases: Escenario 2

En el panel B se muestra que la subclase de servicio A2 experimenta una reducción en la demanda de recursos de CPU del 30% al 20%. Debido a que la subclase de servicio A1 tiene un límite de CPU del 20%, la subclase de servicio A1 no puede exceder su utilización de CPU actual del 20%. La superclase de servicio A reduce su utilización de CPU a un total del 40%. La superclase de servicio B tiene una demanda de recursos de CPU suficiente para reclamar los recursos de CPU a los que temporalmente ha renunciado la subclase de servicio A2, incrementándose la utilización de CPU de la superclase de servicio B del 50% al 60%, tal como se muestra en el panel C.

### Límite de CPU y varias instancias de DB2

En el siguiente conjunto de ejemplos de uso se expone el comportamiento del límite de CPU en un entorno de varias instancias de DB2.

En la Figura 41 en la página 224 se muestra un sistema principal o LPAR que se ha configurado con dos instancias de DB2, la instancia A y la instancia B. Cada instancia contiene una base de datos, las bases de datos 1 y 2. Cada instancia contiene una superclase de servicio, las superclases de servicio A y B. La superclase

de servicio A contiene las subclases de servicio A1 y A2. La superclase de servicio B contiene las subclases de servicio B1 y B2. Para seguir conservando la simplicidad del escenario, las peticiones de usuario para cada base de datos se direccionan sólo a las dos subclases de servicio y no se direcciona ningún trabajo a la subclase de servicio por omisión. Asimismo, como ayuda para describir los conceptos básicos, imaginemos que existe trabajo en ejecución inapreciable en las clases de servicio de usuario, mantenimiento y sistema por omisión. Para los siguientes escenarios, sólo existen dos instancias de DB2 con una base de datos en cada instancia y un único miembro en este sistema principal o LPAR.

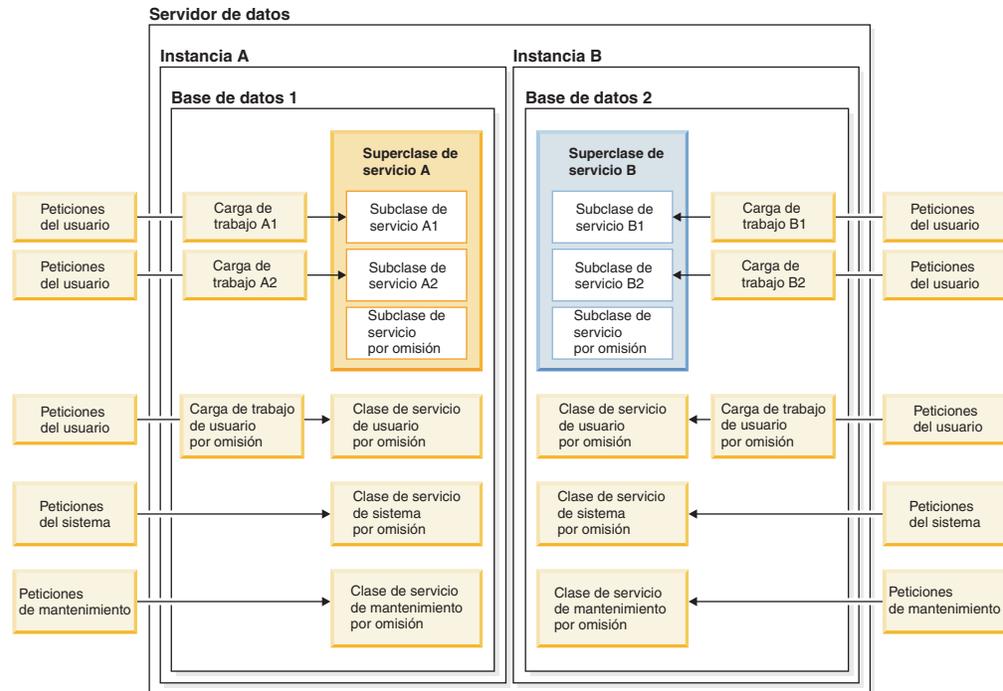


Figura 41. Configuración del servidor de datos: varias instancias de DB2

### Límite de CPU y varias instancias de DB2: Escenario 1

En este ejemplo, la instancia A va a recibir como máximo el 50% de los recursos de CPU totales del sistema principal o LPAR y la instancia B va a recibir como máximo el 50% de los recursos de CPU totales del sistema principal o LPAR. Dentro de la instancia A, el 80% de los recursos de CPU de la instancia van a asignarse a la subclase de servicio A1 y el 20% a la subclase de servicio A2 mediante la asignación de 8000 cuotas de CPU flexibles para A1 y 2000 cuotas de CPU flexibles para A2. Cuando la subclase de servicio A1 no consume su 80% total, deseamos que la subclase de servicio A2 reclame cualquier recurso de CPU no utilizado, y viceversa, para maximizar la utilización de los recursos de CPU dentro de la instancia A. Dentro de la instancia B, el 60% de los recursos de CPU de la instancia van a asignarse a la subclase de servicio B1 y el 40% a la subclase de servicio B2 mediante la asignación de 6000 cuotas de CPU flexibles para B1 y 4000 cuotas de CPU flexibles para B2. Como con las subclases de servicio A1 y A2, deseamos maximizar la utilización de CPU dentro de la instancia B haciendo que la subclase de servicio B1 reclame cualquier recurso de CPU no utilizado cuando la subclase de servicio B2 no utilice su titularidad total de recursos de CPU, y viceversa.

Para configurar las condiciones adecuadas para este escenario tal como se describía en el párrafo anterior, limite los recursos de CPU para cada instancia creando un límite de CPU del 50% en la superclase de servicio A, en la base de datos 1, en la instancia A, y creando un límite de CPU del 50% en la superclase de servicio B, en la base de datos 2, en la instancia B. Cuando ambas instancias utilicen todas sus titularidades de recursos de CPU, se considerará que el sistema principal o LPAR tiene una utilización de CPU del 100%. Si cualquiera de las dos instancias no utiliza toda su titularidad de CPU, la otra instancia no puede reclamar los recursos de CPU no utilizados.

Para cada superclase de servicio de cada instancia, el asignador de gestión de carga de trabajo divide los recursos de CPU disponibles entre las subclases utilizando sus asignaciones de cuotas de CPU flexibles relativas. Mediante la utilización de las cuotas de CPU flexibles asignadas a las subclases de servicio, la única circunstancia en la que una instancia no hace un uso completo de su titularidad de recursos de CPU se produce cuando cada subclase de la instancia no tiene suficiente trabajo en ejecución dentro de ésta para conseguir la utilización de CPU total de su titularidad.

En el panel A de la Figura 42 en la página 226 se muestra que la superclase de servicio A tiene un límite de CPU del 50% y que la superclase de servicio B tiene un límite de CPU del 50%. La subclase de servicio A1 tiene el 80% de los recursos de CPU de la instancia A (40% de los recursos de CPU del sistema principal o LPAR); la subclase de servicio A2 tiene el restante 20% (10% de los recursos de CPU del sistema principal o LPAR). La subclase de servicio B1 tiene el 60% de los recursos de CPU de la instancia B (30% de los recursos de CPU del sistema principal o LPAR); la subclase de servicio B2 tiene el restante 40% (20% de los recursos de CPU del sistema principal o LPAR). Al principio de este escenario, la subclase de servicio A1 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU del sistema principal o LPAR al 40% y la subclase de servicio A2 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU del sistema principal o LPAR al 10%, con lo que la superclase de servicio A dispone de una utilización de recursos de CPU total del 50%. La subclase de servicio B1 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU del sistema principal o LPAR al 30% y la subclase de servicio B2 tiene, como mínimo, suficiente trabajo para llevar su utilización de recursos de CPU del sistema principal o LPAR al 20%, con lo que la superclase de servicio B dispone de una utilización de recursos de CPU total del 50%.

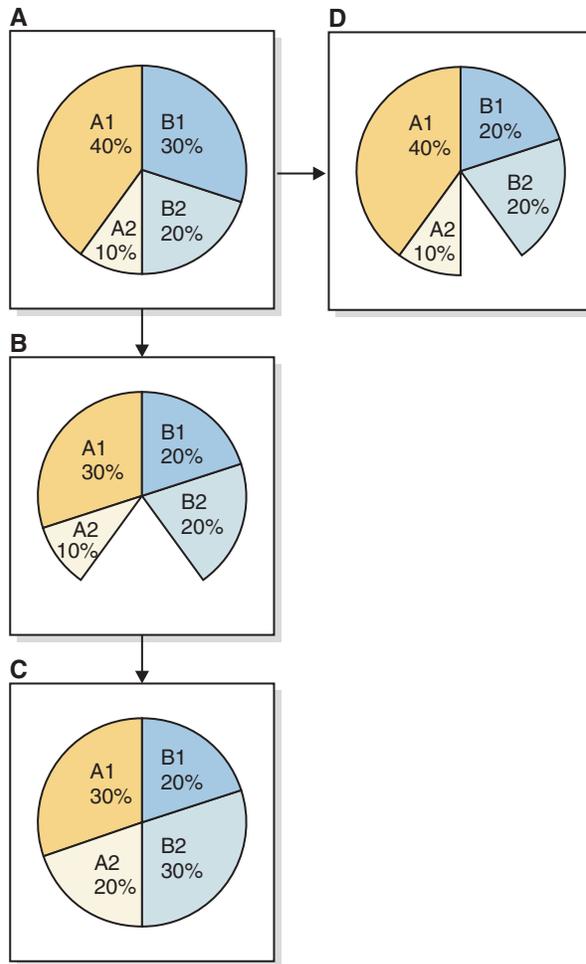


Figura 42. Límite de CPU y varias instancias de DB2: Escenario 1

En el panel B se muestra que la subclase de servicio A1 experimenta una reducción en la demanda de recursos de CPU del 40% al 30% y que la subclase de servicio B1 experimenta una reducción en la demanda de recursos de CPU del 30% al 20%. Como se da por supuesto que la subclase de servicio A2 tiene suficiente trabajo en ejecución para reclamar cualquier recurso de CPU no utilizado que pase a estar disponible, A2 incrementa su utilización de CPU del 10% al 20% porque A2 tiene asignadas cuotas de CPU flexibles, tal como se muestra en el panel C. Lo mismo se aplica a la subclase de servicio B2, que incrementa su utilización de CPU del 20% al 30%, tal como se muestra en el panel C.

Veamos ahora la condición inicial original tal como se describe en el panel A para un ejemplo distinto. Si la subclase de servicio B1 reduce su demanda de CPU del 30% al 20% y la subclase de servicio B2 no tiene suficiente trabajo en ejecución para exceder una demanda de CPU del 20%, la superclase de servicio B de la instancia B no utiliza su titularidad de CPU total del 50% y sigue con una utilización de CPU del 40%, tal como se muestra en el panel D. El resultado es que el sistema principal o LPAR tiene una utilización de CPU de sólo el 90% de los recursos de CPU totales.

### Límite de CPU y varias instancias de DB2: Escenario 2

En este escenario de ejemplo se muestra qué sucede cuando la suma de los límites de CPU de las subclases de servicio no excede el límite de CPU de la superclase de servicio padre.

Mediante la utilización de unas condiciones iniciales similares a las de “Límite de CPU y varias instancias de DB2: Escenario 1”, cambiemos sólo las cuotas de CPU flexibles que se han asignado a las subclases de servicio A1 y A2 en los límites de CPU del 40% y del 10%, respectivamente, de los recursos de CPU totales para el sistema principal o LPAR, tal como se muestra en el panel A de la Figura 43 panel A. Cuando ambas subclases de servicio utilizan los recursos de CPU hasta sus límites de CPU asignados, la utilización de CPU total es del 50% para la superclase de servicio A, haciendo que sea redundante la restricción adicional de un límite de CPU del 50% en la superclase de servicio A.

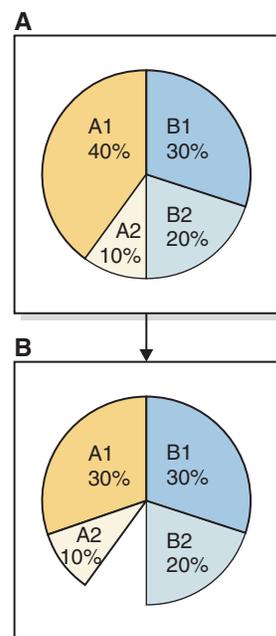


Figura 43. Límite de CPU y varias instancias de DB2: Escenario 2

En el panel B se muestra que la subclase de servicio A1 experimenta una reducción en la demanda de recursos de CPU del 40% al 30% debido a una reducción en la cantidad de trabajo en ejecución en la subclase de servicio. En estas circunstancias, el asignador no puede asignar los recursos de CPU no utilizados, a los que temporalmente ha renunciado la subclase de servicio A1, a la subclase de servicio A2. La subclase de servicio A2 sigue ejecutando cargas de trabajo en su límite de CPU del 10% de los recursos de CPU del sistema principal o LPAR. Esta situación hace que la instancia A no pueda utilizar su titularidad de recursos de CPU total del 50%.

### Límite de CPU y varias instancias de DB2: Escenario 3

En este escenario de ejemplo se muestra qué sucede cuando la suma de los límites de CPU de las subclases de servicio excede el límite de CPU de la superclase de servicio padre.

Mediante la utilización de unas condiciones iniciales similares a las de “Límite de CPU y varias instancias de DB2: Escenario 1”, cambiemos sólo las cuotas de CPU

flexibles que se han asignado a las subclases de servicio A1 y A2 en los límites de CPU del 40% y del 40%, respectivamente, de los recursos de CPU totales para el sistema principal o LPAR. En este ejemplo, el total de los límites de CPU asignados para las subclases de servicio A1 y A2 es del 80%, que excede el límite de CPU del 50% asignado a la superclase de servicio A. El asignador de gestión de carga de trabajo impide que la superclase de servicio A exceda su límite de CPU del 50%. La cantidad de recursos de CPU que se asigna a cada subclase de servicio en la superclase A la determinan las cuotas de CPU que se han asignado a las subclases. No se han asignado explícitamente cuotas de CPU para las subclases de servicio A1 y A2, pero cada una de estas subclases tiene las 1000 cuotas de CPU flexibles que se han asignado al crearse las subclases, con lo que cada subclase dispone de una titularidad de CPU igual. El asignador asigna una división equitativa del 50% de los recursos de CPU del sistema principal o LPAR totales de la titularidad a la superclase de servicio A. El resultado es que el 25% se asigna a la subclase de servicio A1 y el 25% se asigna a la subclase de servicio A2, tal como se muestra en el panel A de la Figura 44.

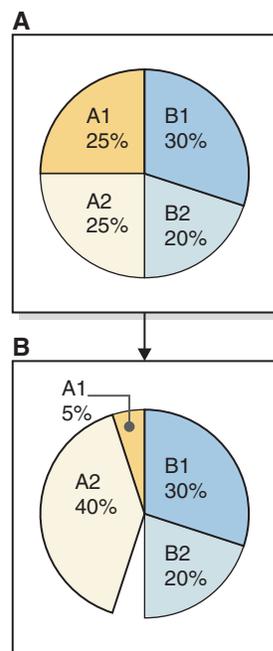


Figura 44. Límite de CPU y varias instancias de DB2: Escenario 3

En el panel B se muestra que la subclase de servicio A1 experimenta una reducción en la demanda de recursos de CPU del 25% al 5% debido a una reducción en la cantidad de trabajo en ejecución en la subclase de servicio. Debido a la asignación de cuotas de CPU flexibles, el asignador puede asignar los recursos de CPU no utilizados, a los que temporalmente ha renunciado la subclase de servicio A1, hasta el 40% que reclama la subclase de servicio A2. La subclase de servicio A2 no puede exceder su límite de CPU del 40% de los recursos de CPU del sistema principal o LPAR. Esta situación hace que la instancia A no pueda utilizar su titularidad de recursos de CPU total del 50%.

**Nota:** Las limitaciones de este enfoque de gestión del trabajo en varias instancias en el mismo sistema principal o LPAR son que éste le limita a una única superclase de servicio en cada instancia. Para los sistemas operativos que tienen un gestor de carga de trabajo en el que puede integrarse la gestión de carga de trabajo de DB2,

una alternativa consiste en correlacionar las clases de servicio de DB2 de cada instancia con una clase de servicio en un WLM de sistema operativo (SO) (como AIX WLM y Linux WLM) y asignar los límites fijos del WLM de SO en cada clase de servicio del SO para establecer límites superiores en la utilización de recursos de CPU de cada instancia.

## Información adicional

Se proporciona completa información acerca de los siguientes temas del asignador de gestión de carga de trabajo:

- Asignador de gestión de carga de trabajo, consulte: “Asignador de gestión de carga de trabajo” en la página 177.
- Nivel de simultaneidad de asignación, consulte: “Nivel de simultaneidad de asignación” en la página 234.
- Nivel de utilización de CPU mínimo para que se considere activa una clase de servicio, consulte: “Utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 230.
- Cuotas de CPU fijas, consulte: “Cuotas de CPU fijas” en la página 200.
- Cuotas de CPU flexibles, consulte: “Cuotas de CPU flexibles” en la página 209.
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo” en la página 237.

## Establecimiento de un límite de CPU

Establezca el atributo del límite de CPU mediante la sentencia `CREATE SERVICE CLASS` para crear una nueva clase de servicio y un límite de CPU, y utilice la sentencia `ALTER SERVICE CLASS` para alterar el valor del límite de CPU para una clase de servicio existente. El límite de CPU se utiliza para limitar la utilización de CPU de una clase de servicio bajo todas las circunstancias durante el control del asignador de gestión de carga de trabajo.

## Antes de empezar

Debe habilitar el asignador de gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos `wlm_dispatcher`, si todavía no lo ha hecho. Mediante la habilitación del asignador de gestión de carga de trabajo, el control de los recursos de CPU por medio del atributo del límite de CPU pasa a estar disponible por omisión. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.

## Acerca de esta tarea

En esta tarea, el límite de CPU se establece para una clase de servicio mediante la sentencia `CREATE SERVICE CLASS` o `ALTER SERVICE CLASS`, por medio del procesador de línea de mandatos.

### Restricciones

Puede asignar límites de CPU sólo a las clases de servicio que el asignador de gestión de carga de trabajo pueda controlar, que son las clases de servicio de mantenimiento y de usuario. El asignador de gestión de carga de trabajo no puede controlar las asignaciones de recursos de CPU para las clases de servicio del sistema.

## Procedimiento

Para crear una nueva clase de servicio y establecer el límite de CPU (vaya al paso 2 si desea alterar un valor de límite de CPU de una clase de servicio existente):

1. Emita la sentencia CREATE SERVICE CLASS para crear una nueva clase de servicio y establecer el límite de CPU en el 25% en este ejemplo:

```
create service class nombre_clase_servicio cpu limit 25
```

Para alterar el valor del límite de CPU para una clase de servicio existente:

2. Emita la sentencia ALTER SERVICE CLASS para alterar el límite de CPU por el 50% en este ejemplo:

```
alter service class nombre_clase_servicio cpu limit 50
```

## Resultados

Ha establecido o alterado el valor del límite de CPU para la clase de servicio especificada. El asignador de gestión de carga de trabajo no puede asignar recursos de CPU a esta clase de servicio por encima del límite establecido.

## Qué hacer a continuación

Después de haber habilitado el asignador de gestión de carga de trabajo para que gestione los recursos de CPU y después de haber establecido el límite de CPU, considere la realización de las tareas siguientes:

- Establezca el nivel de simultaneidad de asignación configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**. Consulte: “Establecimiento del nivel de simultaneidad de asignación” en la página 235.
- Habilitar las cuotas de CPU fijas y flexibles configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_cpu\_shares**. Consulte en paso 1 en “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Crear o alterar una clase de servicio y configurar una cuota de CPU fija mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU fijas” en la página 207.
- Crear o alterar una clase de servicio y configurar una cuota de CPU flexible mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Establezca el porcentaje de utilización de CPU mínimo para que el asignador de gestión de carga de trabajo pueda considerar que las clases de servicio están activas configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util**. Consulte: “Establecimiento de la utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 233.

## Utilización de recursos de CPU mínima para que se considere activa la clase de servicio

Mediante el establecimiento del parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util**, puede controlar el nivel mínimo de utilización de recursos de CPU que el asignador de gestión de carga de trabajo utilizará para considerar que la clase de servicio está activamente comprometida en la ejecución

del trabajo. En la planificación de la asignación de recursos de CPU que realiza el asignador, se tienen en cuenta las cuotas de CPU de las clases de servicio activas únicamente.

Al gestionar las asignaciones de recursos de CPU basados en cuotas, el asignador de gestión de carga de trabajo considera una clase de servicio como activa y tiene en cuenta toda asignación de cuotas de CPU en la asignación de la planificación de recursos de CPU global si las peticiones de la base de datos que se ejecutan en esa clase de servicio están utilizando cualquier cantidad de los recursos de CPU. En determinadas circunstancias, es deseable poder ejercer algún control sobre qué cantidad de actividad de CPU generada por la clase de servicio se necesita para que el asignador de gestión de carga de trabajo incluya las cuotas de CPU de la clase de servicio durante la planificación de los recursos de CPU.

En el escenario de ejemplo que se representa en el panel A de la Figura 45 en la página 232, la clase de servicio A contiene una carga de trabajo de transacción de alta prioridad que sólo se ejecuta durante el día en horario laboral, mientras que las clases de servicio B y C contienen trabajos por lotes de baja prioridad en curso que se ejecutan de día y de noche. La clase de servicio A se ha protegido de interrupciones mediante la asignación de cuotas de CPU fijas a las clases de servicio B y C. Por la noche, cuando no existe carga de trabajo de transacción en la clase de servicio A (tal como se muestra en el panel B), las clases de servicio B y C pueden utilizar todos los recursos de CPU y realizar un progreso mucho más rápido (tal como se muestra en el panel C).

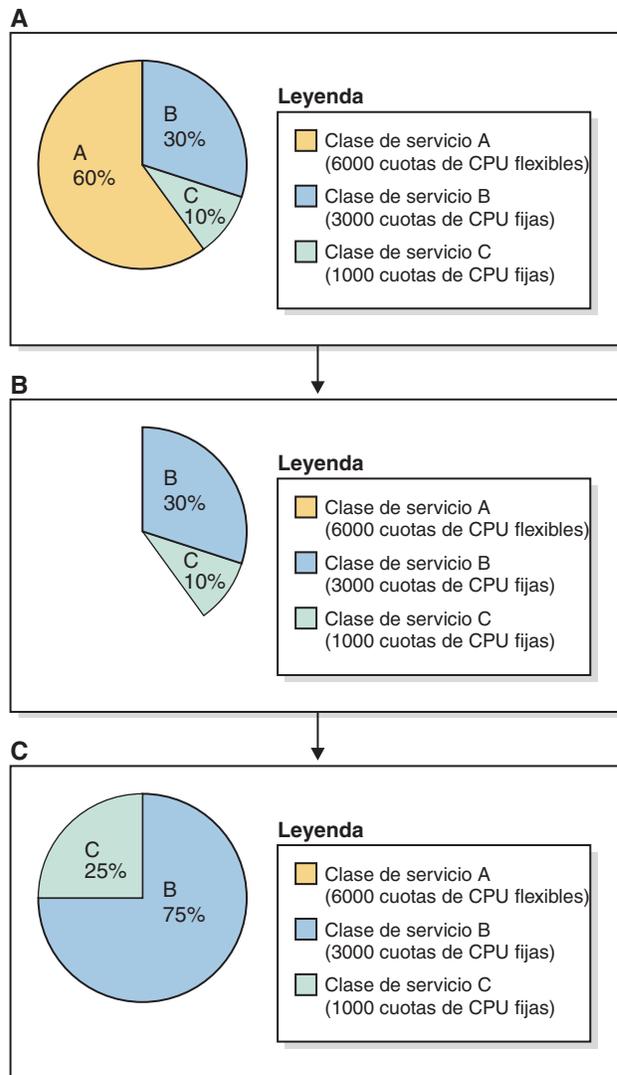


Figura 45. Ejemplo de utilización de CPU mínima: cuotas de CPU fijas y flexibles

Veamos ahora qué sucede si una pequeña parte del trabajo de transacción por lotes sigue produciéndose durante la noche en la clase de servicio A. En este caso, el asignador de gestión de carga de trabajo considera a la clase de servicio A como activa, y los recursos de CPU no utilizados a los que ha renunciado temporalmente la clase A no están disponibles para las clases de servicio B y C. Las titularidades de recursos de CPU para las clases de servicio B y C se muestran en el panel B como una pequeña fracción en el gráfico de utilización de CPU que representa la actividad de la clase de servicio A (que no se muestra en el gráfico circular del panel B). El resultado es un progreso mucho más lento durante la noche para las clases de servicio B y C que el que tiene lugar en el escenario original que se muestra en el panel C.

El usuario dispone de máxima flexibilidad para gestionar sus cargas de trabajo de DB2 mediante la opción que le permite establecer un porcentaje de utilización de CPU alcanzado el cual o después de haberse superado el cual una clase de servicio se considera como activa en el sistema principal o LPAR. Cuando una clase de servicio se considera como no activa, su asignación de cuotas de CPU no se tiene en cuenta en los cálculos de titularidad de recursos de CPU, permitiendo de esta forma que las clases de servicio, en particular las que tienen asignadas cuotas de

CPU fijas, puedan reclamar los recursos de CPU no utilizados. Este porcentaje mínimo de utilización de CPU se especifica mediante la configuración del valor de porcentaje del parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util**. Este parámetro de configuración puede establecerse en un valor de porcentaje comprendido entre el 0 y el 100, y el valor por omisión es 5. Para que esta configuración de parámetro sea eficaz, debe habilitarse el asignador de gestión de carga de trabajo estableciendo el valor del parámetro de configuración del gestor de bases de datos **wlm\_dispatcher** en YES.

Revisemos el escenario de realización de una pequeña parte del trabajo por lotes descrito anteriormente. Con el valor de porcentaje del parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util** establecido ahora en un valor ligeramente más alto que el correspondiente a la pequeña fracción de utilización de CPU que tenía lugar durante la noche para la clase de servicio A, podemos estar más seguros de que el progreso de los trabajos por lotes que se ejecutan durante la noche de las clases de servicio B y C ha mejorado considerablemente, de forma similar al del escenario original en el que la utilización de CPU era como la que se muestra en el panel C.

## Información adicional

Se proporciona completa información acerca de los siguientes temas del asignador de gestión de carga de trabajo:

- Asignador de gestión de carga de trabajo, consulte: “Asignador de gestión de carga de trabajo” en la página 177.
- Nivel de simultaneidad de asignación, consulte: “Nivel de simultaneidad de asignación” en la página 234.
- Cuotas de CPU fijas, consulte: “Cuotas de CPU fijas” en la página 200.
- Cuotas de CPU flexibles, consulte: “Cuotas de CPU flexibles” en la página 209.
- Límite de CPU, consulte: “Límite de CPU” en la página 215.
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo” en la página 237.

## Establecimiento de la utilización de recursos de CPU mínima para que se considere activa la clase de servicio

Establezca la utilización de recursos de CPU mínima para una clase de servicio estableciendo el valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util** en un porcentaje. Se considera que las clases de servicio que utilizan recursos de CPU equivalentes o superiores a ese porcentaje están activas en el sistema principal o en la partición lógica (LPAR), y las cuotas de CPU de las clases de servicio activas forman parte de los cálculos de asignación de CPU.

## Procedimiento

Para establecer el porcentaje de utilización de CPU mínima para que se considere que las clases de servicio están activas:

Mediante la utilización del procesador de línea de mandatos (CLP), emita el mandato **UPDATE DATABASE MANAGER CONFIGURATION** (o **UPDATE DBM CFG**) y establezca el valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util** en el 6%; el valor del parámetro se actualiza inmediatamente estableciendo primero una conexión con su instancia de DB2:

```
attach to nombre_instancia
update dbm cfg using wlm_disp_min_util 6
detach
```

## Resultados

Se ha establecido el porcentaje de utilización de CPU mínima para que se considere que las clases de servicio están activas. Se considera que las clases de servicio que tienen un porcentaje de utilización de CPU inferior a este mínimo están desocupadas, y sus asignaciones de CPU pueden volver a asignarse potencialmente a las clases de servicio que reclaman recursos de CPU de reserva.

## Qué hacer a continuación

Después de haber establecido el porcentaje de utilización de CPU mínima para que el asignador de la gestión de carga de trabajo considere que las clases de servicio están activas, considere la realización de las tareas siguientes:

- Para que el valor de utilización de CPU mínima sea efectivo, debe habilitar el asignador de la gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos **wlm\_dispatcher**. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.
- Establezca el nivel de simultaneidad de asignación configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**. Consulte: “Establecimiento del nivel de simultaneidad de asignación” en la página 235.
- Crear o alterar una clase de servicio y configurar el límite de CPU mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Establecimiento de un límite de CPU” en la página 229.
- Habilitar las cuotas de CPU fijas y flexibles configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_cpu\_shares**. Consulte en paso 1 en “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Crear o alterar una clase de servicio y configurar una cuota de CPU fija mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU fijas” en la página 207.
- Crear o alterar una clase de servicio y configurar una cuota de CPU flexible mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.

## Nivel de simultaneidad de asignación

El asignador de gestión de carga de trabajo controla el número de hebras que el gestor de bases de datos permitirá que se asignen simultáneamente al sistema operativo. A este número se le conoce como nivel de simultaneidad de asignación. Puede establecer el nivel de simultaneidad de asignación mediante el parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**. Aquí se proporciona información detallada.

## Características y funciones

El número de agentes en ejecución simultánea que el asignador de gestión de carga de trabajo gestiona generalmente es un múltiplo bajo del número de CPU que hay en el sistema. La finalidad es establecer un nivel de simultaneidad lo suficientemente grande para garantizar que los recursos de CPU del sistema

principal o LPAR se utilizan por completo y no se hace un uso superior, a la vez que se contabilizan las latencias de planificación del sistema operativo a medida que las hebras conmutan del estado activo al estado no activo y viceversa. Este valor óptimo garantiza la máxima eficacia y otorga al asignador de gestión de carga de trabajo el máximo control sobre la asignación de recursos de CPU. Si el valor de simultaneidad configurado es demasiado pequeño, no existirá suficiente trabajo que utilice por completo los recursos de CPU disponibles. Si el valor de simultaneidad es demasiado grande, el asignador tendrá menos control sobre el trabajo que se ejecuta en el sistema, lo que podría sumarse a la contención por los recursos.

Mediante el establecimiento del parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur**, puede especificar cómo ha de establecer el propio gestor de bases de datos DB2 el nivel de simultaneidad (COMPUTED), o bien puede establecer manualmente el nivel de simultaneidad en un valor fijo.

Como directriz general, el nivel de simultaneidad de asignación óptimo es cuatro veces el número de CPU disponibles en el hardware del servidor. Esta directriz generalmente es suficiente para la mayoría de las plataformas y es el nivel que obtendrá como valor por omisión (COMPUTED es el valor por omisión del parámetro de configuración). Para obtener completa información acerca de cómo establecer el nivel de simultaneidad de asignación, consulte: “Establecimiento del nivel de simultaneidad de asignación”.

El valor especificado del parámetro de configuración de base de datos **wlm\_disp\_concur** se aplica a cada sistema principal o LPAR. En entornos de bases de datos de varios miembros, el nivel de simultaneidad especificado se comparte entre las particiones lógicas.

## Ajuste

Después de haber habilitado el asignador de gestión de carga de trabajo, supervise el sistema para determinar si se ha producido una caída del rendimiento y la utilización de CPU. Si cae cualquiera de las dos métricas, incremente el valor del parámetro de configuración en incrementos del número de procesadores disponibles.

## Información adicional

Se proporciona completa información acerca de los temas siguientes:

- Asignador de gestión de carga de trabajo, consulte: “Asignador de gestión de carga de trabajo” en la página 177.
- Cuotas de CPU flexibles, consulte: “Cuotas de CPU flexibles” en la página 209.
- Cuotas de CPU fijas, consulte: “Cuotas de CPU fijas” en la página 200.
- Límite de CPU, consulte: “Límite de CPU” en la página 215.
- Nivel de utilización de CPU mínimo para que se considere activa una clase de servicio, consulte: “Utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 230.
- Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo” en la página 237.

## Establecimiento del nivel de simultaneidad de asignación

Establezca el nivel de simultaneidad de asignación estableciendo el valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur** en

COMPUTED o en un valor establecido manualmente. El nivel de simultaneidad de asignación ha de establecerse en un valor óptimo que garantice la máxima eficiencia y que proporcione al asignador de gestión de carga de trabajo el máximo control sobre las asignaciones de recursos de CPU.

## Procedimiento

Para establecer el nivel de simultaneidad de asignación:

Mediante la utilización del procesador de línea de mandatos (CLP), emita el mandato **UPDATE DATABASE MANAGER CONFIGURATION** (o **UPDATE DBM CFG**) y establezca el valor del parámetro de configuración del gestor de bases de datos **wlm\_disp\_concur** en uno de los siguientes valores posibles:

- COMPUTED (valor por omisión)
- *valor\_establecido\_manualmente*

En el ejemplo siguiente se establece manualmente el nivel de simultaneidad de asignación en 16, que es 4 veces las 4 CPU disponibles en el sistema, y que se actualiza inmediatamente estableciendo primero una conexión con su instancia de DB2:

```
attach to nombre_instancia
update dbm cfg using wlm_disp_concur 16
detach
```

## Resultados

Se ha establecido en nivel de simultaneidad de asignación. Si se establece en COMPUTED, el nivel de simultaneidad lo determina el gestor de bases de datos DB2. Si se establece manualmente, puede que deba ajustar el nivel de simultaneidad para establecer un valor óptimo que garantice la máxima eficiencia y que proporcione al asignador de gestión de carga de trabajo máximo control sobre las asignaciones de recursos de CPU.

## Qué hacer a continuación

Después de haber establecido el nivel de simultaneidad de asignación, considere la realización de las tareas siguientes:

- Para que el valor del nivel de simultaneidad de asignación sea efectivo, debe habilitar el asignador de gestión de carga de trabajo configurando el parámetro de configuración del gestor de bases de datos **wlm\_dispatcher**. Consulte: “Habilitación de un asignador de gestión de carga de trabajo” en la página 198.
- Crear o alterar una clase de servicio y configurar el límite de CPU mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Establecimiento de un límite de CPU” en la página 229.
- Habilitar las cuotas de CPU fijas y flexibles configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_cpu\_shares**. Consulte en paso 1 en “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Crear o alterar una clase de servicio y configurar una cuota de CPU fija mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU fijas” en la página 207.

- Crear o alterar una clase de servicio y configurar una cuota de CPU flexible mediante la sentencia CREATE SERVICE CLASS o ALTER SERVICE CLASS. Consulte: “Habilitación y establecimiento de cuotas de CPU flexibles” en la página 213.
- Establezca el porcentaje de utilización de CPU mínimo para que el asignador de gestión de carga de trabajo pueda considerar que las clases de servicio están activas configurando el parámetro de configuración del gestor de bases de datos **wlm\_disp\_min\_util**. Consulte: “Establecimiento de la utilización de recursos de CPU mínima para que se considere activa la clase de servicio” en la página 233.

Después de haber habilitado el asignador de gestión de carga de trabajo y de haber establecido el nivel de simultaneidad de asignación, supervise el sistema para determinar si se ha producido una caída del rendimiento y la utilización de CPU. Si cae cualquiera de las dos métricas, incremente el valor del parámetro de configuración **wlm\_disp\_concur** en incrementos del número de procesadores disponibles. Para obtener información adicional acerca de la supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo, consulte: “Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo”.

## Supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo

La supervisión y ajuste del rendimiento del asignador de gestión de carga de trabajo pueden realizarse con las funciones de tabla y elementos de supervisión que se proporcionan con el gestor de bases de datos DB2. Aquí se proporciona información detallada.

### Introducción

Para supervisar y ajustar el rendimiento del asignador de gestión de carga de trabajo y obtener los mejores resultados, necesita las herramientas adecuadas. Se proporcionan funciones de tabla y elementos de supervisión que pueden ayudarle a supervisar el rendimiento del asignador. Después de haber analizado los datos de supervisión recopilados tal como se describe aquí, puede ajustar el nivel de simultaneidad del asignador o redistribuir las titularidades de CPU ajustando las cuotas de CPU y los límites de CPU de la clase de servicio para ajustar el rendimiento del asignador.

En los apartados siguientes se describen los tipos de cargas de trabajo que han de tenerse en cuenta, pues son distintos en cuanto a que pueden supervisarse mejor para ofrecer los datos adecuados que se deben analizar, y las mediciones del rendimiento, que son las más adecuadas para el tipo de carga de trabajo en particular que está considerándose.

### Tipos de cargas de trabajo

Desde la perspectiva de la medición del rendimiento con el fin de ajustar la configuración del asignador para obtener del sistema el mejor rendimiento posible de la carga de trabajo, existen dos cargas de trabajo que han de tenerse en cuenta, las cargas de trabajo por lotes y de transacción. Cada tipo de carga de trabajo tiene mediciones características del rendimiento que son más adecuadas para determinar el rendimiento del sistema bajo ese tipo de carga de trabajo. Utilice las mediciones del rendimiento que mejor caractericen el tipo de carga de trabajo que experimenta el sistema.

## **Por lotes**

Una carga de trabajo por lotes tiene una o varias aplicaciones que se conectan con la base de datos, y cada aplicación envía actividad tras actividad o transacción tras transacción sin ninguna pausa. La medición del rendimiento más importante de esta carga de trabajo es con qué rapidez se completa el conjunto completo de actividades o de transacciones. La velocidad de proceso del gestor de bases de datos es el principal factor que determina la rapidez con la que se completa el conjunto completo de actividades o de transacciones.

## **De transacción**

En una carga de trabajo de transacción, existe un usuario en el terminal que envía una actividad o transacción a la base de datos, éste espera una respuesta, analiza la respuesta y decide si ha de enviar o no una actividad o transacción de seguimiento. Para este tipo de carga de trabajo, la medición más importante de su rendimiento es con qué rapidez el usuario recupera un resultado individual. La velocidad de proceso del gestor de bases de datos para procesar una única actividad o transacción para cada usuario individual del sistema es el principal factor que determina la rapidez con la que el usuario promedio puede recuperar un resultado individual. La rapidez con la que el gestor de bases de datos puede procesar todas las actividades o transacciones de un usuario en el transcurso de un periodo de tiempo determinado no es la medición más importante porque ésta depende más del comportamiento del usuario que del rendimiento del gestor de bases de datos.

## **Mediciones del rendimiento**

Puede utilizar las siguientes mediciones del rendimiento para determinar cuál es el rendimiento del sistema bajo un tipo de carga de trabajo en particular.

### **Promedio de rendimiento**

El promedio de rendimiento es el número medio de finalizaciones del servicio por unidad de tiempo. Si el servicio es una transacción o unidad de trabajo (UOW), el promedio de rendimiento de UOW es el número de finalizaciones de unidades de trabajo por unidad de tiempo. Generalmente se presenta en forma de transacciones por segundos o de transacciones por minuto. El promedio de rendimiento es una medición útil del rendimiento del sistema cuando el tipo de trabajo que se mide es una carga de trabajo por lotes.

El promedio de rendimiento de actividad es el número medio de finalizaciones de actividad por unidad de tiempo. En un sistema en el que se ejecutan principalmente unidades de trabajo de larga ejecución que contienen muchas actividades individuales, es más fácil medir el progreso de las actividades dentro de la carga de trabajo midiendo el rendimiento de la actividad que medir el progreso de la carga de trabajo midiendo el rendimiento de la UOW.

### **Promedio de tiempo de respuesta**

El promedio de tiempo de respuesta es la cantidad de tiempo media que se necesita para completar un único servicio a partir del momento en el que se ha solicitado el servicio. Si el servicio es una transacción o unidad de trabajo (UOW), el promedio de tiempo de respuesta de UOW es la cantidad de tiempo que se necesita para completar una UOW a partir del momento en el que se ha solicitado. El promedio de tiempo de respuesta es una medición útil del rendimiento del

sistema cuando el tipo de trabajo que se mide es una carga de trabajo de transacción. La aproximación más exacta al promedio de tiempo de respuesta de UOW es la estadística **uow\_lifetime\_avg** que está disponible desde las funciones de tabla `MON_SAMPLE_SERVICE_CLASS_METRICS` y `MON_SAMPLE_WORKLOAD_METRICS`, las funciones de tabla `WLM_GET_SERVICE_SUBCLASS_STATS` y `WLM_GET_WORKLOAD_STATS` y los grupos de datos lógicos del supervisor de sucesos `event_scstats` y `event_wlstats` de los que se informa en el supervisor de sucesos de estadísticas de WLM. Existe un formato de información de la vida útil de la UOW más sofisticado en el histograma `UowLifetime`, que también está disponible en el supervisor de sucesos.

El promedio de tiempo de respuesta de actividad es la cantidad de tiempo media que se necesita para que una única actividad devuelva sus resultados a partir del momento en el que se ha iniciado la actividad. La aproximación más exacta al promedio de tiempo de respuesta de actividad es la estadística **coord\_act\_lifetime\_avg** que está disponible desde las funciones de tabla `WLM_GET_SERVICE_SUBCLASS_STATS` y `WLM_GET_WORKLOAD_STATS` y los grupos de datos lógicos del supervisor de sucesos `event_scstats` y `event_wlstats`. Este número se mide en cada miembro y se restablece cuando un miembro se desactiva o se llama al procedimiento `WLM_COLLECT_STATS`. La razón por la que puede ser una aproximación es que para un tipo de actividad, una actividad de cursor, la actividad puede devolver algunos resultados antes de que se haya finalizado y depende de que el usuario acabe de leer el conjunto de resultados y cierre el cursor para que se considere que la actividad se ha completado. Existe un formato de la vida útil de actividad más sofisticado en el histograma `CoordActLifetime`, que también está disponible en el supervisor de sucesos.

### Utilización de CPU

Otra métrica que es útil al ajustar el asignador de gestión de carga de trabajo, con independencia del tipo de carga, es la utilización de CPU. La utilización de CPU es la fracción de tiempo que los recursos de CPU están ocupados en el sistema principal o LPAR. La utilización de CPU es la métrica que el asignador de gestión de carga de trabajo utiliza para asignar los recursos de CPU a cualquiera de las clases de servicio. La utilización de CPU también es la métrica que puede utilizar para verificar que la configuración del asignador de gestión de carga de trabajo funciona como esperaba. Puede medir la utilización de CPU para los mismos intervalos que los elementos de supervisor **uow\_throughput**, **uow\_lifetime\_avg** y **act\_throughput** mediante las funciones de tabla `MON_SAMPLE_SERVICE_CLASS_METRICS` y `MON_SAMPLE_WORKLOAD_METRICS`, las funciones de tabla `WLM_GET_SERVICE_SUBCLASS_STATS` y `WLM_GET_WORKLOAD_STATS` y los grupos de datos lógicos del supervisor de sucesos `event_scstats` y `event_wlstats` que recopila y de los que informa el supervisor de sucesos de estadísticas de WLM y que éste recopila.

**Nota:** Si las mediciones de utilización de CPU no son las esperadas para las clases de servicio que ha creado, compruebe si existen cargas de trabajo en ejecución en las clases de servicio de usuario y mantenimiento por omisión porque estas cargas de trabajo no se han asignado explícitamente a las clases de servicio que ha creado. El hecho de olvidarse de incluir las cargas de trabajo que se ejecutan en estas clases de servicio por omisión, a cada una de las cuales se asignan 1000 cuotas de CPU fijas por omisión al habilitarse por primera vez las cuotas de CPU, puede justificar las mediciones de la utilización de CPU que no son las que se esperaban.

La utilización de CPU de la que se informa por medio de las funciones de tabla y los supervisores de sucesos son los recursos de CPU que consume el trabajo que se ejecuta sólo en las clases de servicio de usuario y mantenimiento. El trabajo que el asignador no gestiona no contabiliza en la utilización de CPU.

El trabajo que el asignador de gestión de carga de trabajo no gestiona incluye:

- Trabajo que realizan las aplicaciones o productos de middleware, distintos del gestor de bases de datos DB2, que ejecutan parte de su trabajo fuera del gestor de bases de datos DB2
- Trabajo que realizan las entidades que se ejecutan en la clase de servicio del sistema de DB2
- Trabajo que realizan otras instancias de DB2
- Trabajo del gestor de bases de datos no DB2 que se realiza en procesos en modalidad delimitada (FMP), como los procedimientos almacenados delimitados
- Trabajo del gestor de bases de datos no DB2 que se realiza en rutinas fiables

Para obtener la utilización de CPU para estos otros consumidores de recursos de CPU, debe utilizarse la supervisión de nivel de sistema operativo (nivel de SO), como la que se proporciona con los gestores de carga de trabajo de SO.

### **Velocidad de CPU**

La velocidad de CPU es la estadística que determina si existe o no contención por un recurso y el grado de esa contención. Cuando todo el acceso a un recurso es mutuamente exclusivo y existen varios peticionarios de ese recurso que están a la espera de poder acceder a éste al mismo tiempo, debe existir alguna forma que permita colocar en cola las peticiones de acceso o que haga que los peticionarios desistan. Cuando está permitido formar una cola, el tiempo que un peticionario necesita para obtener y acabar de utilizar un recurso puede exceder el tiempo empleado en utilizar simplemente el recurso. La velocidad es la relación entre el tiempo empleado en utilizar simplemente el recurso y el tiempo total empleado en la espera y utilización del recurso. Se mide en una escala del cero al 100%. Cuando existe gran cantidad de contención por un recurso, la velocidad se desplaza hacia el cero. Cuando no existe ninguna contención por un recurso, no existe tiempo en cola; la velocidad de CPU alcanza su máximo valor del 100%.

Cuando se ha habilitado el asignador de gestión de carga de trabajo, puede medir la velocidad de CPU mediante las funciones de tabla `MON_SAMPLE_SERVICE_CLASS_METRICS` y `MON_SAMPLE_WORKLOAD_METRICS`, las funciones de tabla `WLM_GET_WORKLOAD_STATS` y `WLM_GET_SERVICE_SUBCLASS_STATS` y los grupos de datos lógicos del supervisor de sucesos `event_scstats` y `event_wlstats` que recopila y de los que informa el supervisor de sucesos de estadísticas de WLM. Un valor de velocidad de CPU bajo indica que existe contención por los recursos de CPU en el sistema principal o LPAR e indica que el asignador de gestión de carga de trabajo puede ser efectivo en el desplazamiento de los recursos de CPU hacia las clases de servicio de alta prioridad, alejándolos de las clases de servicio de baja prioridad. Una velocidad de CPU alta indica que el asignador de gestión de carga de trabajo tendrá un efecto limitado en la mejora del rendimiento de la carga de trabajo porque cada una de las peticiones de servicio de recursos de CPU ya está siendo atendida sin ningún retardo.

---

## Cancelación de actividades

Si una actividad consume demasiados recursos, o tarda mucho en ejecutarse, puede cancelarla. Cancelar una actividad es menos radical que forzar la aplicación que envió la actividad. Una actividad cancelada devuelve el error SQL4725N al usuario, pero no finaliza la conexión ni afecta a otras actividades del usuario. Forzar la aplicación finaliza la conexión y las actividades del usuario.

### Acerca de esta tarea

Solo puede cancelar explícitamente una actividad si una actividad coordinadora está actualmente trabajando en una petición para la actividad. Si cancela una actividad en estado IDLE (es decir, no se está procesando ninguna petición), la actividad adquiere el estado de CANCEL\_PENDING y se cancela cuando se recibe la petición siguiente. Por ejemplo, si intenta cancelar una actividad CURSOR entre búsquedas, no se devuelve el error SQL4725N al usuario hasta la siguiente búsqueda después de la cancelación.

Todas las actividades del usuario son susceptibles de cancelación, incluso el programa de utilidad de carga y los procedimientos almacenados.

### Procedimiento

1. Identifique la actividad que desea cancelar. Puede utilizar la función de tabla WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para identificar las actividades que se ejecutan en una aplicación. También puede utilizar la función de tabla MON\_GET\_ACTIVITY\_DETAILS\_COMPLETE para ver detalles adicionales sobre una actividad en particular si la información de WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES no es suficiente para identificar el trabajo que las actividades están realizando.
2. Cancele la actividad utilizando el procedimiento almacenado WLM\_CANCEL\_ACTIVITY. El procedimiento almacenado toma los argumentos siguientes: *application\_handle*, *uow\_id* y *activity\_id*. Para conocer un ejemplo de cómo utilizar este procedimiento almacenado, consulte “Caso de ejemplo: Identificación de actividades que tardan demasiado tiempo en completarse” en la página 380.



---

## Capítulo 4. Supervisión e intervención

El tercer dominio de la gestión de carga de trabajo es la supervisión, que debe realizarse sobre la marcha.

La finalidad principal de la supervisión es validar la salud y la eficiencia del sistema y de las cargas de trabajo individuales que se ejecutan en éste. Mediante la utilización de funciones de tabla, puede acceder a los datos de operación en tiempo real, como una lista de las apariciones de carga de trabajo en ejecución y las actividades que se ejecutan en una clase de servicio o el promedio de los tiempos de respuesta. Mediante la utilización de los supervisores de sucesos puede capturar información de actividad detallada y agregar estadísticas de actividad para el análisis histórico.

La consulta de la información de agregado es el primer paso que ha de realizarse al crear una estrategia de supervisión. Los agregados proporcionan una buena imagen de la actividad general del servidor de datos y también son rentables, pues no necesitará recopilar información acerca de cada actividad que podría interesarle. Podrá recopilar más información detallada a medida que adquiera un conocimiento más profundo del ámbito de sus necesidades de supervisión.

Las tareas de supervisión típicas que puede realizar son:

- Análisis de la carga de trabajo del sistema como ayuda para diseñar la configuración de la gestión de carga de trabajo de DB2 inicial.
- Seguimiento e investigación del comportamiento del sistema para obtener tipos de información de operación que le permitan:
  - Analizar la degradación del rendimiento del sistema
  - Diagnosticar las actividades que necesitan un tiempo excesivo para completarse
  - Investigar la contención del agente
  - Aislar las consultas que ofrecen un pobre rendimiento

Dispondrá de información acerca de las actividades, las clases de servicio, las cargas de trabajo, las clases de trabajo, las colas de umbrales y las violaciones de umbral.

- Ejercer control sobre el entorno de ejecución, para cancelar las actividades en cola que ha determinado que causarán problemas o cancelar las actividades en ejecución que ha diagnosticado que tendrán un impacto negativo en el sistema.

---

### Supervisión en tiempo real con funciones de tabla

Los datos de supervisión en tiempo real incluyen información acerca del trabajo que actualmente se ejecuta en el sistema, así como estadísticas y métricas para el trabajo que se ha realizado en el sistema, que pueden ayudarle a determinar los patrones de uso y la asignación de recursos y a identificar áreas de problemas. Puede utilizar las funciones de tabla de DB2 para obtener esta información de operación.

Las funciones de tabla cuyos nombres empiezan por *WLM\_* son funciones de tabla de la gestión de carga de trabajo de DB2. Estas funciones de tabla proporcionan acceso a un conjunto de datos relacionados con la gestión de la carga de trabajo, como las estadísticas de gestión de carga de trabajo, en forma de tabla de DB2

virtual para la que puede emitir una sentencia SELECT. Esto le permite grabar aplicaciones para consultar datos y analizar éstos como si se encontraran en una tabla física en el servidor de datos. Las funciones de tabla de la gestión de carga de trabajo de DB2 se califican con el nombre de esquema SYSPROC.

Las funciones de tabla cuyos nombres empiezan por *MON\_* son funciones de métricas de supervisión. Las métricas de supervisión proporcionan datos de supervisión acerca de la salud y el rendimiento de la consulta del servidor de datos de DB2, que pueden utilizarse como datos de entrada para una herramienta de otro proveedor o bien junto con script adicional que puede proporcionar para analizar las métricas devueltas. Aquí sólo se incluyen las funciones de métricas de supervisión relacionadas con la gestión de carga de trabajo de DB2. Las funciones de tabla de métricas del supervisor son similares a las funciones de tabla de estadísticas de la gestión de carga de trabajo. Ambas devuelven elementos que describen el trabajo que ha tenido lugar en el sistema. Las diferencias clave entre estas funciones de tabla de métricas de supervisión y las funciones de tabla de la gestión de carga de trabajo de DB2 son:

- Las funciones de tabla de la gestión de carga de trabajo de DB2 proporcionan datos de naturaleza más estadística, como valores calculados en forma de promedios, marcas de límite superior, desviaciones estándar, etc. A su vez, las funciones de tabla de métricas de supervisión proporcionan un conjunto mucho más completo de datos de supervisión en bruto.
- Los datos de los que informan las funciones de estadísticas de DB2 se restablecen cuando los datos se envían al supervisor de sucesos estadísticos. Este restablecimiento es necesario para que valores tales como las marcas de límite superior tengan un significado respecto a un intervalo de recopilación específico. Los datos de los que informan las funciones de métricas de supervisión también los captura un supervisor de sucesos estadísticos, pero nunca se restablecen. Los datos de los que informan las interfaces de supervisión se acumulan desde el momento en el que se activa una base de datos hasta el momento en el que se desactiva.

Algunas funciones de tabla devuelven conjuntos de información acerca del trabajo que actualmente se ejecuta en un sistema:

*Tabla 49. Funciones de tabla que muestran el trabajo que actualmente se ejecuta en el sistema*

Objetos para los que se recopila información	Funciones e información devuelta
Apariciones de carga de trabajo	<p>La función de tabla <code>WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES</code> devuelve una lista de las instancias de carga de trabajo, en los miembros de carga de trabajo, que se han asignado a una clase de servicio. Para cada aparición, existe información acerca del estado actual y de los atributos de conexión utilizados para asignar la carga de trabajo a la clase de servicio, y estadísticas de actividad que indican el volumen de la actividad y los índices de éxito. Para ver un ejemplo de cómo utilizar esta función de tabla, consulte: “Ejemplo: Investigación del uso de agentes por parte de una clase de servicio” en la página 101.</p> <p>También está disponible la función de tabla <code>WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97</code>, en desuso.</p>

Tabla 49. Funciones de tabla que muestran el trabajo que actualmente se ejecuta en el sistema (continuación)

Objetos para los que se recopila información	Funciones e información devuelta
Actividades de aparición de carga de trabajo	<p>La función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES devuelve una lista de las actividades actuales que están asociadas a una aparición de carga de trabajo. Para cada actividad, existe información acerca del estado actual de la actividad (por ejemplo, en ejecución o en cola), el tipo de actividad (por ejemplo, LOAD, READ o DDL) y la hora a la que se ha iniciado la actividad. Para ver ejemplos de cómo utilizar esta función de tabla, consulte: “Ejemplo: Agregación de datos utilizando funciones de tabla de gestión de carga de trabajo de DB2” en la página 254 y “Caso de ejemplo: Identificación de actividades que tardan demasiado tiempo en completarse” en la página 380.</p> <p>También está disponible la función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97, en desuso.</p>
Agentes de clase de servicio	<p>La función de tabla WLM_GET_SERVICE_CLASS_AGENTS devuelve una lista de los agentes de base de datos que se asocian a una clase de servicio o a un descriptor de contexto de aplicación. La información devuelta también muestra el estado actual del agente, la acción que el agente está realizando y el estado de esa acción. Para ver un ejemplo de cómo utilizar esta función de tabla, consulte: “Ejemplo: Investigación del uso de agentes por parte de una clase de servicio” en la página 101.</p> <p>También está disponible la función de tabla WLM_GET_SERVICE_CLASS_AGENTS_V97, en desuso.</p>
Actividades	<p>La función de tabla MON_GET_ACTIVITY_DETAILS devuelve métricas y otra información detallada en forma de documento XML sobre una actividad específica que se identifica mediante su descriptor de contexto de aplicación, ID de unidad de trabajo e ID de actividad. Uno de los detalles que se devuelven es el tipo de actividad; en función de ese tipo, se devuelve un conjunto de datos adicionales. Por ejemplo, para las actividades de SQL, se proporcionan estimaciones de coste e información acerca del texto de la sentencia, los datos del paquete y las filas devueltas o modificadas. También se proporcionan detalles acerca del nivel de aislamiento y de los recursos de procesador, entre otros.</p> <p>También está disponible la función de tabla WLM_GET_ACTIVITY_DETAILS, en desuso. Para ver un ejemplo en el que se muestra cómo utilizar esta función de tabla, consulte “Ejemplo: Supervisión del comportamiento actual del sistema en diferentes niveles utilizando las funciones de tabla de la gestión de carga de trabajo de DB2” en la página 250.</p>

Algunas funciones de tabla devuelven datos de supervisión para todas las peticiones que se han ejecutado en el sistema agregado por subclase de servicio y objetos de carga de trabajo:

Tabla 50. Funciones de tabla que muestran datos de supervisión agregados por objetos de la gestión de carga de trabajo de DB2

Objetos para los que se agregan datos	Funciones e información devuelta
Cargas de trabajo	<p>Las funciones de tabla MON_GET_WORKLOAD y MON_GET_WORKLOAD_DETAILS devuelven métricas para una o más cargas de trabajo. Las métricas que devuelve esta función representan la acumulación de todas las métricas de todas las apariciones de carga de trabajo que utilizan la misma definición de carga de trabajo.</p> <p>La función de tabla MON_GET_WORKLOAD devuelve las métricas utilizadas más comúnmente en formato basado en columnas, y es un método eficiente para recuperar métricas base.</p> <p>La función de tabla MON_GET_WORKLOAD_DETAILS devuelve el conjunto completo de métricas disponibles en forma de documento XML, que proporciona máxima flexibilidad para dar formato a la salida. La salida basada en XML puede analizarla directamente un analizador XML, o puede convertirse en formato relacional mediante la función XMLTABLE.</p>
Subclases de servicio	<p>Las funciones de tabla MON_GET_SERVICE_SUBCLASS y MON_GET_SERVICE_SUBCLASS_DETAILS devuelven métricas para una o más subclases de servicio. Las métricas que devuelven estas funciones de tabla representan la acumulación de todas las métricas para todas las peticiones que se han ejecutado en la subclase de servicio indicada.</p> <p>La función de tabla MON_GET_SERVICE_SUBCLASS devuelve las métricas utilizadas más comúnmente en formato basado en columnas, y es un método eficiente para recuperar métricas base.</p> <p>La función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS devuelve el conjunto completo de métricas disponibles en forma de documento XML, que proporciona máxima flexibilidad para dar formato a la salida. La salida basada en XML puede analizarla directamente un analizador XML o puede convertirse en formato relacional mediante la función XMLTABLE.</p>
Conexiones	<p>La función de tabla MON_GET_CONNECTION devuelve datos que se agregan a través de conexiones de usuario al sistema.</p>
Unidades de trabajo	<p>La función de tabla MON_GET_UNIT_OF_WORK devuelve datos que se agregan para la unidad actual de trabajo en una conexión de usuario.</p>

## Información estadística

También existe información estadística general para diversos objetos. Puede utilizar esta información estadística para distintas finalidades, como para verificar si los cambios realizados en la configuración de la gestión de carga de trabajo de DB2 han producido el efecto esperado. Si crea una nueva clase de trabajo para clasificar las actividades READ, por ejemplo, puede verificar si las actividades READ se han clasificado correctamente bajo la nueva clase de trabajo. También puede utilizar funciones de tabla para reconocer rápidamente determinados problemas relacionados con el sistema. Por ejemplo, puede utilizar funciones de tabla para determinar un valor aceptable para el promedio de vida útil de actividad y

reconocer cuándo este valor excede su rango normal, lo que posiblemente indicará un problema para el que deberán realizarse más investigaciones.

En la siguiente tabla se indican las estadísticas que puede obtener mediante la utilización de las funciones de tabla. Todas las funciones de tabla de estadísticas devuelven las estadísticas que se han acumulado desde la última vez que restableció las estadísticas.

*Tabla 51. Funciones de tabla que muestran información estadística*

<b>Objetos para los que se devuelven estadísticas</b>	<b>Funciones y estadísticas devueltas</b>
Superclases de servicio	La función de tabla WLM_GET_SERVICE_SUPERCLASS_STATS muestra estadísticas de resumen de los miembros de base de datos del nivel de superclase de servicio, es decir, marcas de límite superior para conexiones simultáneas, que son útiles para determinar la actividad de carga de trabajo máxima.
Subclases de servicio	<p>La función de tabla WLM_GET_SERVICE_SUBCLASS_STATS muestra estadísticas de resumen de los miembros de base de datos del nivel de subclase de servicio (todas las actividades ejecutadas en las subclases de servicio). Las estadísticas incluyen números de actividades completadas y promedios de tiempo de ejecución. Esta información es útil cuando se consulta la salud general del sistema y la distribución de las actividades de las clases de servicio y los miembros de base de datos. Para ver ejemplos de cómo utilizar esta función de tabla, consulte “Ejemplo: Obtención de estadísticas puntuales de clases de servicio” en la página 253, “Ejemplo: Agregación de datos utilizando funciones de tabla de gestión de carga de trabajo de DB2” en la página 254, “Ejemplo: Análisis de una ralentización de sistema relacionada con la clase de servicio” en la página 99 y “Caso de ejemplo: Investigación de una ralentización del sistema relacionada con la carga de trabajo” en la página 376.</p> <p>También está disponible la función de tabla WLM_GET_SERVICE_SUBCLASS_STATS_V97, en desuso.</p>
Cargas de trabajo	<p>La función de tabla WLM_GET_WORKLOAD_STATS muestra estadísticas de resumen de los miembros de base de datos del nivel de carga de trabajo. Incluyen marcas de límite superior para las apariciones de carga de trabajo simultáneas y números de actividades completadas. Esta información es útil cuando se supervisa la salud general del sistema o cuando se desean obtener detalles para identificar áreas de problemas. Para ver un ejemplo de cómo utilizar esta función de tabla, consulte “Caso de ejemplo: Investigación de una ralentización del sistema relacionada con la carga de trabajo” en la página 376.</p> <p>También está disponible la función de tabla WLM_GET_WORKLOAD_STATS_V97, en desuso.</p>
Conjuntos de acciones de trabajo	La función de tabla WLM_GET_WORK_ACTION_SET_STATS muestra estadísticas de resumen de los miembros de base de datos del nivel de conjunto de acciones de trabajo, es decir, el número de actividades de cada clase de trabajo a las que se han aplicado las acciones de trabajo correspondientes. Esta información es útil para entender la efectividad de un conjunto de acciones de trabajo y para entender los tipos de actividades que se ejecutan en el sistema. Para ver un ejemplo de cómo utilizar esta función de tabla, consulte “Ejemplo: Análisis de cargas de trabajo por tipo de actividad” en la página 66.

Tabla 51. Funciones de tabla que muestran información estadística (continuación)

Objetos para los que se devuelven estadísticas	Funciones y estadísticas devueltas
Colas de umbrales	La función de tabla WLM_GET_QUEUE_STATS muestra estadísticas de resumen de los miembros de base de datos para las colas utilizadas para los umbrales. Las estadísticas incluyen los números actuales y totales de las actividades en cola y el tiempo total que han permanecido en una cola. Esta información es útil cuando desea consultar la actividad en cola actual o cuando desea validar si ha definido un umbral correctamente. Una colocación en cola excesiva podría indicar que un umbral es demasiado restrictivo, y una colocación en cola muy pequeña podría indicar que un umbral no es lo suficientemente restrictivo o no es necesario.

Las estadísticas sólo son útiles si el período de tiempo durante el que se recopilan reviste significado. La recopilación de estadísticas durante un período de tiempo muy prolongado, y durante cualquier período de tiempo cuando se utiliza el procedimiento almacenado WLM\_COLLECT\_STATS, podría ser menos útil si es difícil identificar cambios en las tendencias o áreas de problemas porque existen demasiados datos antiguos. Por ello, puede restablecer las estadísticas en cualquier momento.

Gracias a la carga de trabajo por omisión y a las clases de servicio de usuario por omisión, las funciones de supervisión están disponibles desde el momento en el que instala el servidor de datos de DB2. Pueden ayudarle a empezar a identificar orígenes de actividades, que puede utilizar para crear cargas de trabajo y las clases de servicio a las que puede asignar éstas.

## Ejemplo: Utilización de funciones de tabla de la gestión de carga de trabajo de DB2

La supervisión en tiempo real de la gestión de carga de trabajo de DB2 proporciona gran cantidad de datos. En el ejemplo de este tema se muestra cómo puede empezar a utilizar la información.

En este caso, sólo existen la carga de trabajo por omisión y la clase de servicio. Utilice este ejemplo para saber cómo puede utilizar las funciones de tabla para entender qué, exactamente, está ejecutándose en el servidor de datos. Siga estos pasos:

1. Utilice la función de tabla de estadísticas de superclase de servicio para que se muestren todas las superclases de servicio. Después de haber instalado o actualizado a DB2 9.5 o posterior, se definen tres superclases por omisión: una para las actividades de mantenimiento, otra para las actividades del sistema y otra para las actividades del usuario. SYSDEFAULTUSERCLASS es la clase de servicio que nos interesa.

```
SELECT VARCHAR(SERVICE_SUPERCLASS_NAME,30) AS SUPERCLASS
      FROM TABLE(WLM_GET_SERVICE_SUPERCLASS_STATS('',-1)) AS T

SUPERCLASS
-----
SYSDEFAULTSYSTEMCLASS
SYSDEFAULTMAINTENANCECLASS
SYSDEFAULTUSERCLASS
```

3 registro(s) seleccionado(s).

2. Utilice la función de tabla de estadísticas de subclase de servicio para que se muestren estadísticas para todas las subclases de servicio de la superclase SYSDEFAULTUSERCLASS. Para cada subclase de servicio verá el volumen actual de las peticiones que están procesándose, el número de actividades cuya ejecución se ha completado y la distribución general de las actividades en los miembros (que posiblemente indicarán un problema si la distribución es desigual). Opcionalmente, puede obtener estadísticas adicionales, incluidos el promedio de vida útil para las actividades, el promedio de tiempo que las actividades han permanecido en cola, etc. Puede obtener estadísticas opcionales para una subclase de servicio especificando la palabra clave COLLECT AGGREGATE ACTIVITY DATA en la sentencia ALTER SERVICE CLASS para habilitar la recopilación de estadísticas de actividad.

```
SELECT VARCHAR(SERVICE_SUPERCLASS_NAME, 20) AS SUPERCLASS,
       VARCHAR(SERVICE_SUBCLASS_NAME, 20) AS SUBCLASS,
       COORD_ACT_COMPLETED_TOTAL,
       COORD_ACT_ABORTED_TOTAL,
       COORD_ACT_REJECTED_TOTAL,
       CONCURRENT_ACT_TOP
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(
 'SYSDEFAULTUSERCLASS', 'SYSDEFAULTSUBCLASS', -1))
AS T
```

SUPERCLASS	SUBCLASS	COORD_ACT_COMPLETED_TOTAL	COORD_ACT_ABORTED_TOTAL	COORD_ACT_REJECTED_TOTAL	CONCURRENT_ACT_TOP
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0	0	1

1 registro(s) seleccionado(s).

3. Para una subclase de servicio determinada, utilice la función de tabla de información de aparición de carga de trabajo para obtener una lista de las apariciones de una carga de trabajo que se han correlacionado con la subclase de servicio. La función de tabla visualiza todos los atributos de conexión, que puede utilizar para identificar el origen de las actividades. Esta información puede ser bastante útil para determinar definiciones de carga de trabajo personalizadas en el futuro. Por ejemplo, quizás una aparición de carga de trabajo específica aquí indicada tiene un gran volumen de trabajo de una aplicación, tal como muestra el recuento de actividades completadas.

```
SELECT APPLICATION_HANDLE,
       VARCHAR(WORKLOAD_NAME, 30) AS WORKLOAD,
       VARCHAR(SESSION_AUTH_ID, 20) AS SESSION_AUTH_ID,
       VARCHAR(APPLICATION_NAME, 20) AS APPL_NAME
FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES(
 'SYSDEFAULTUSERCLASS', 'SYSDEFAULTSUBCLASS', -1))
AS T
```

APPLICATION_HANDLE	WORKLOAD	SESSION_AUTH_ID	APPL_NAME
431	SYSDEFAULTUSERWORKLOAD	SWALKTY	db2bp

1 registro(s) seleccionado(s).

- a. Para esa aplicación, utilice la función de tabla de información de actividades de aparición de carga de trabajo para que se muestren las actividades actuales de los miembros de base de datos que se han creado desde la conexión de la aplicación. Puede utilizar esta información para diversas finalidades, incluida la identificación de las actividades que podrían causar problemas en el servidor de datos.

```
SELECT APPLICATION_HANDLE,
       LOCAL_START_TIME,
       UOW_ID,
       ACTIVITY_ID,
       ACTIVITY_TYPE
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(431,-1)) AS T
```

APPLICATION_HANDLE	LOCAL_START_TIME	UOW_ID	ACTIVITY_ID	ACTIVITY_TYPE
431	2008-06-17-12.49.46.854259	11	1	READ_DML

1 registro(s) seleccionado(s)

- b. Para cada actividad, recupere información más detallada utilizando la función de tabla de detalles de actividad. Puede que los datos muestren que algunas sentencias de SQL devuelven gran número de filas, que algunas actividades han estado inactivas durante un plazo de tiempo prolongado o que algunas consultas en ejecución tienen un coste estimado extremadamente elevado. En estos casos, puede tener sentido definir umbrales para identificar y evitar un potencial daño en el comportamiento en el futuro.

```
SELECT VARCHAR(NAME, 20) AS NAME,
       VARCHAR(VALUE, 40) AS VALUE
FROM TABLE(WLM_GET_ACTIVITY_DETAILS(431,11,1,-1))
      AS T WHERE NAME IN ('UOW_ID', 'ACTIVITY_ID', 'STMT_TEXT')
```

NAME	VALUE
UOW_ID	1
ACTIVITY_ID	1
STMT_TEXT	select * from syscat.tables

3 registro(s) seleccionado(s).

## Ejemplo: Supervisión del comportamiento actual del sistema en diferentes niveles utilizando las funciones de tabla de la gestión de carga de trabajo de DB2

La gestión de carga de trabajo de DB2 proporciona una serie de funciones de tabla que puede utilizar para obtener datos sobre su configuración de la gestión de carga de trabajo.

La instalación de DB2 Versión 9.5 o posterior crea a un conjunto de cargas de trabajo y clases de servicio por omisión. Antes de decidir cómo implementar su propia la solución de la gestión de carga de trabajo de DB2, puede utilizar las funciones de tabla para observar el trabajo que se realiza en el sistema en términos de apariciones de la carga de trabajo por omisión, clases de servicio y actividades.

Puede empezar obteniendo la lista de apariciones de la carga de trabajo en una clase de servicio. Para ello, utilice la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES. En el ejemplo siguiente, se proporciona una serie vacía para *service\_superclass\_name* y *service\_subclass\_name*, y se proporciona -2 (un carácter comodín) para *member*:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(COORD_MEMBER),1,4) AS COORDMEMB,
       SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHNDL,
       SUBSTR(CHAR(WORKLOAD_NAME),1,22) AS WORKLOAD_NAME,
       SUBSTR(CHAR(WORKLOAD_OCCURRENCE_ID),1,6) AS WLO_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES(' ', ' ', -2)) AS SCINFO
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB, APPHNDL, WORKLOAD_NAME, WLO_ID
```

Supongamos que el sistema tiene cuatro miembros de base de datos y que hay dos aplicaciones que realizan actividades en la base de datos cuando emite la consulta. Los resultados serían parecidos a los siguientes:

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	COORDMEMB	APPHNDL	WORKLOAD_NAME	WLO_ID
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0	1	SYSDEFAULTUSERWORKLOAD	1
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0	2	SYSDEFAULTUSERWORKLOAD	2

Los resultados indican que se han asignado ambas apariciones de la carga de trabajo a la carga de trabajo SYSDEFAULTUSERWORKLOAD. Los resultados también muestran que ambas apariciones de la carga de trabajo se han asignado a la subclase de servicio SYSDEFAULTSUBCLASS en la superclase de servicio SYSDEFAULTUSERCLASS, y ambas apariciones de la carga de trabajo proceden del mismo miembro coordinador (miembro 0).

A continuación, también puede utilizar de nuevo la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES para determinar los atributos de conexión de las apariciones de la carga de trabajo:

```
SELECT SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHNDL,
       SUBSTR(CHAR(WORKLOAD_NAME),1,22) AS WORKLOAD_NAME,
       SUBSTR(CHAR(WORKLOAD_OCCURRENCE_ID),1,6) AS WLO_ID,
       SUBSTR(CHAR(SYSTEM_AUTH_ID),1,9) AS SYSAUTHID,
       SUBSTR(CHAR(APPLICATION_NAME),1,15) AS APPLNAME
FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES(' ', ' ', 0)) AS SCINFO
ORDER BY APPHNDL, WORKLOAD_NAME, WLO_ID
```

APPHNDL	WORKLOAD_NAME	WLO_ID	SYSAUTHID	APPLNAME
1	SYSDEFAULTUSERWORKLOAD	1	LYNN	accountspay
2	SYSDEFAULTUSERWORKLOAD	2	KATE	businessobjects

A continuación, puede utilizar la función de tabla WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para mostrar las actividades actuales de una de las apariciones de la carga de trabajo:

```
SELECT SUBSTR(CHAR(COORD_MEMBER),1,5) AS COORD,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(UOW_ID),1,5) AS UOWID,
       SUBSTR(CHAR(ACTIVITY_ID),1,5) AS ACTID,
       SUBSTR(CHAR(PARENT_UOW_ID),1,8) AS PARUOWID,
       SUBSTR(CHAR(PARENT_ACTIVITY_ID),1,8) AS PARACTID,
       SUBSTR(CHAR(ACTIVITY_TYPE),1,9) AS ACTTYPE,
       SUBSTR(CHAR(NESTING_LEVEL),1,7) AS NESTING
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(1, -2)) AS WLOACTS
ORDER BY MEMB, UOWID, ACTID
```

COORD	MEMB	UOWID	ACTID	PARUOWID	PARACTID	ACTTYPE	NESTING
0	0	1	3	-	-	CALL	0
0	0	1	5	1	3	READ_DML	1
0	1	1	5	-	-	READ_DML	1
0	2	1	5	-	-	READ_DML	1
0	3	1	5	-	-	READ_DML	1

Los resultados de la consulta muestran que la aparición de la carga de trabajo 1 se ejecuta en dos actividades. Una actividad es un procedimiento almacenado (que se indica mediante el tipo de actividad CALL), mientras que la otra actividad es una actividad de DML que realiza una lectura (por ejemplo una sentencia SELECT). La actividad de DML que está anidada en la llamada de procedimiento almacenado. Se puede observar que la actividad de DML está anidada porque el identificador de unidad de trabajo principal y el identificador de actividad principal de la actividad de DML coinciden con el identificador de unidad de trabajo y el identificador de actividad de la actividad CALL. También puede observar que la

actividad de DML se ejecuta en los miembros de base de datos 0, 1, 2 y 3. La información de identificador principal sólo está disponible en el miembro coordinador.

Puede obtener más información sobre una determinada actividad actualmente en ejecución utilizando la función de tabla MON\_GET\_ACTIVITY\_DETAILS. Esta función de tabla devuelve un documento XML cuyos elementos describen la actividad. En este ejemplo, se utiliza la función XMLTABLE para devolver una tabla de resultados de la salida de código XML.

```

SELECT D.APP_HANDLE,
       D.MEMBER,
       D.COORD_MEMBER,
       D.LOCAL_START_TIME,
       D.UOW_ID,
       D.ACTIVITY_ID,
       D.PARENT_UOW_ID,
       D.PARENT_ACTIVITY_ID,
       D.ACTIVITY_TYPE,
       D.NESTING_LEVEL,
       D.INVOCATION_ID,
       D.ROUTINE_ID
FROM TABLE(MON_GET_ACTIVITY_DETAILS(65592, 1, 1, -2)) AS ACTDETAILS,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
          '$details/db2_activity_details' PASSING XMLPARSE(DOCUMENT
          ACTDETAILS.DETAILS) as "details"
COLUMNS "APP_HANDLE"          BIGINT    PATH 'application_handle',
         "MEMBER"              BIGINT    PATH 'member',
         "COORD_MEMBER"       BIGINT    PATH 'coord_member',
         "LOCAL_START_TIME"    VARCHAR(26) PATH 'local_start_time',
         "UOW_ID"              BIGINT    PATH 'uow_id',
         "ACTIVITY_ID"         BIGINT    PATH 'activity_id',
         "PARENT_UOW_ID"       BIGINT    PATH 'parent_uow_id',
         "PARENT_ACTIVITY_ID" BIGINT    PATH 'parent_activity_id',
         "ACTIVITY_TYPE"       VARCHAR(10) PATH 'activity_type',
         "NESTING_LEVEL"       BIGINT    PATH 'nesting_level',
         "INVOCATION_ID"       BIGINT    PATH 'invocation_id',
         "ROUTINE_ID"          BIGINT    PATH 'routine_id'
) AS D;

```

APP_HANDLE	MEMBER	COORD_MEMBER	LOCAL_START_TIME
UOW_ID	ACTIVITY_ID	PARENT_UOW_ID	
PARENT_ACTIVITY_ID	ACTIVITY_TYPE	NESTING_LEVEL	INVOCATION_ID
ROUTINE_ID			
65592	1	1	
2009-04-07-18.39.42.549197	-	1	
1	-	READ_DML	0
65592	0	1	
2009-04-07-18.39.42.552763	-	1	
1	-	READ_DML	0

2 registro(s) seleccionado(s).

Las funciones de tabla mencionadas anteriormente proporcionan una descripción general del trabajo que se ejecuta en el sistema. La información que estas funciones de tabla proporcionan en relación con el estado del trabajo se limita a un estado de actividad, como EXECUTING. Si desear investigar con más detalle para descubrir lo que sucede exactamente en una clase de servicio en un instante específico, puede ejecutar la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS.

En el ejemplo siguiente, se llama a WLM\_GET\_SERVICE\_CLASS\_AGENTS proporcionando 1 a *application\_handle* y -2 (un carácter comodín) a *member*:

```

SELECT SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHANDLE,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(AGENT_TID),1,9) AS AGENT_TID,
       SUBSTR(CHAR(AGENT_TYPE),1,11) AS AGENTTYPE,

```

```

SUBSTR(AGENT_STATE,1,10) AS AGENTSTATE,
SUBSTR(REQUEST_TYPE,1,14) AS REQTYPE,
SUBSTR(CHAR(UOW_ID),1,6) AS UOW_ID,
SUBSTR(CHAR(ACTIVITY_ID),1,6) AS ACT_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS(' ', ' ', 1, -2)) AS SCDETAILS
ORDER BY APPHANDLE, MEMB, AGENT_TID

```

APPHANDLE	MEMB	AGENT_TID	AGENTTYPE	AGENTSTATE	REQTYPE	UOW_ID	ACT_ID
1	0	3	COORDINATOR	ACTIVE	FETCH	1	5
1	0	4	PDBSUBAGENT	ACTIVE	SUBSECTION:1	1	5
1	1	2	PDBSUBAGENT	ACTIVE	SUBSECTION:2	1	5

Los resultados muestran un agente coordinador y un subagente en el miembro 0 y un subagente en el miembro 1 que opera en nombre de una actividad con un identificador de unidad de trabajo 1 y un identificador de actividad 5. La información del agente coordinador indica que la petición es una petición de captación.

## Ejemplo: Obtención de estadísticas puntuales de clases de servicio

Cada actividad se correlaciona con una clase de servicio antes de su ejecución. Para supervisar el sistema puede utilizar las funciones de tabla de estadísticas de clases de servicio y consultar todas las clases de servicio en todos los miembros de base de datos para obtener estadísticas puntuales.

Puede utilizar la sentencia siguiente para obtener estadísticas de clases de servicio, como el promedio de vida útil de las actividades. Si proporciona una serie vacía a un argumento para la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS, significa que dicho argumento no debe restringir el resultado. El valor del último argumento, *member*, es -2 (un carácter comodín), que significa que deben devolverse los datos de todas las peticiones de la base de datos.

**Nota:** La información de vida útil sólo se devuelve para aquellas clases de servicio definidas con COLLECT AGGREGATE ACTIVITY DATA.

```

SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
CAST(COORD_ACT_LIFETIME_AVG / 1000 AS DECIMAL(9,3)) AS AVGLIFETIME,
CAST(COORD_ACT_LIFETIME_STDDEV / 1000 AS DECIMAL(9,3)) AS STDDEVLIFETIME,
SUBSTR(CAST(LAST_RESET AS VARCHAR(30)),1,16) AS LAST_RESET
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(' ', ' ', -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB

```

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	AVGLIFETIME	STDDEVLIFETIME	LAST_RESET
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	691.242	34.322	2006-07-24-11.44
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	644.740	22.124	2006-07-24-11.44
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	612.431	43.347	2006-07-24-11.44
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	593.451	28.329	2006-07-24-11.44

También puede utilizar la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS para obtener la marca de límite superior de simultaneidad de actividades de coordinador que se ejecutan en la clase de servicio en cada miembro:

```

SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
CONCURRENT_ACT_TOP AS ACTHIGHWATERMARK
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(' ', ' ', -2)) AS SCSTATS

```

ORDER BY SUPERCLASS\_NAME, SUBCLASS\_NAME, MEMB

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	ACTHIGHWATERMARK
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 0	0	10
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 1	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 2	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS 3	0	0

Al revisar el promedio de vida útil y el número de actividades completadas, puede utilizar la salida de la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS para obtener una vista resumida de la carga de trabajo en cada miembro de base de datos. Una variación significativa en las marcas de límite superior y los promedios que devuelve una función de tabla puede indicar un cambio en la carga de trabajo del sistema.

## Ejemplo: Agregación de datos utilizando funciones de tabla de gestión de carga de trabajo de DB2

Puede realizar diferentes agregaciones en los datos de la tabla en una configuración de gestión de carga de trabajo de DB2 para supervisar el sistema e identificar posibles problemas.

A continuación se muestra un ejemplo de agregación de datos que puede realizar para identificar problemas.

### Identificación de aumentos repentinos del número de consultas que se ejecutan en una carga de trabajo

Supongamos que tiene una carga de trabajo denominada WL1. Puede identificar una situación en la que un gran número de consultas se ejecutan en la carga de trabajo mostrando el número total de actividades de coordinador no anidadas en ejecución para la carga de trabajo en todo el sistema:

```
SELECT SUBSTR(WORKLOAD_NAME,1,22) AS WLNAME,
COUNT(*) AS TOTAL_EXE_ACT
FROM TABLE(WLM_GET_SERVICE_CLASS WORKLOAD_OCCURRENCES(' ', ' ', -2)) AS APPS,
TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(APPS.APPLICATION_HANDLE, -2)) AS APPACTS
WHERE WORKLOAD_NAME = 'WL1' AND
APPS.DBPARTITIONNUM = APPS.COORD_PARTITION_NUM AND
ACTIVITY_STATE = 'EXECUTING' AND
NESTING_LEVEL = 0
GROUP BY WORKLOAD_NAME
```

WLNAME	TOTAL_EXE_ACT
WL1	5

## Ejemplo: Determinación de las actividades que un umbral de WLM coloca en cola y su orden en la cola

Puede utilizar una función de tabla del gestor de carga de trabajo (WLM) de DB2, WLM\_GET\_SERVICE\_CLASS\_AGENTS, para ver qué aplicaciones o actividades ha colocado en cola el umbral de WLM y el orden de esas aplicaciones o actividades en la cola.

Cualquier agente que el umbral de WLM haya puesto en cola tendrá el valor WLM\_QUEUE en la columna EVENT\_OBJECT en la fila correspondiente que se devuelve para el agente. Además, la columna AGENT\_STATE\_LAST\_UPDATE\_TIME contendrá la hora a la que el agente entró en el estado WLM\_QUEUE, que coincide con la hora a la que el agente entró en la cola. Con esta información, puede construirse una vista sencilla para enumerar todas las

aplicaciones o actividades que el umbral de WLM ha puesto en cola, y para obtener la hora de entrada en la cola de cada una de ellas.

Para crear una vista que utilice la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS para obtener la información de cola de WLM para los ejemplos que se proporcionan aquí, ejecute la sentencia siguiente:

```
CREATE VIEW WLM_QUEUE_INFO (APPLICATION_HANDLE, UOW_ID, ACTIVITY_ID,
    THRESHOLD_NAME, QUEUE_ENTRY_TIME, MEMBER)
AS SELECT APPLICATION_HANDLE, UOW_ID, ACTIVITY_ID,
    VARCHAR(EVENT_OBJECT_NAME, 128), AGENT_STATE_LAST_UPDATE_TIME,
    MEMBER FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS(NULL,NULL,NULL,-2))
AS T WHERE EVENT_OBJECT = 'WLM_QUEUE'
```

**Nota:** Para los umbrales TOTALSCPARTITIONCONNECTIONS, **id\_unidad\_trabajo** e **id\_actividad** no son aplicables (NA) y, por lo tanto, se muestran como NULL.

Esta vista puede utilizarse para responder fácilmente a preguntas como éstas:

- ¿Cuántas aplicaciones o actividades están actualmente en cola por acción del umbral de WLM?
- ¿En qué orden están las aplicaciones o actividades en la cola del umbral de WLM?

## Ejemplo 1

Para hacer un recuento de las aplicaciones que están en cola por cada umbral de colas, ejecute la sentencia siguiente:

```
SELECT VARCHAR(THRESHOLD_NAME, 30) AS THRESHOLD, COUNT(*)
    AS QUEUED_ENTRIES FROM WLM_QUEUE_INFO GROUP BY THRESHOLD_NAME
```

A continuación se muestra un ejemplo de la salida que se obtiene después de ejecutar la sentencia anterior:

THRESHOLD	QUEUED_ENTRIES
-----	-----
TH1	3

1 registro(s) seleccionado(s).

## Ejemplo 2

Para enumerar todas las actividades que están en cola en el umbral TH1 CONCURRENTDBCOORDACTIVITIES y mostrar el orden de las actividades en la cola según la hora de entrada en la cola, ejecute la sentencia siguiente:

```
SELECT QUEUE_ENTRY_TIME, APPLICATION_HANDLE, UOW_ID, ACTIVITY_ID FROM
    WLM_QUEUE_INFO WHERE THRESHOLD_NAME = 'TH1' ORDER BY QUEUE_ENTRY_TIME ASC
```

A continuación se muestra un ejemplo de la salida que se obtiene después de ejecutar la sentencia anterior:

QUEUE_ENTRY_TIME	APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID
-----	-----	-----	-----
2009-11-09-18.08.32.583286	145	1	2
2009-11-09-18.08.42.589623	146	1	1
2009-11-09-18.08.54.607083	147	1	1

3 registro(s) seleccionado(s).

---

## Supervisión histórica con supervisores de sucesos de WLM

La gestión de carga de trabajo de DB2 utiliza supervisores de sucesos para capturar información que podría ser de utilidad en el futuro o para el análisis histórico.

Puede utilizar tres supervisores de sucesos. Cada supervisor de sucesos sirve para una finalidad distinta:

### Supervisor de sucesos de actividad

Este supervisor captura información acerca de actividades individuales de una clase de servicio, carga de trabajo o clase de trabajo o actividades que han infringido un umbral. La cantidad de datos que se capturan para cada actividad puede configurarse y debe tenerse en cuenta al determinar la cantidad de espacio de disco y el período de tiempo necesario de conservación de los datos del supervisor. Un uso común de los datos de actividad es utilizar éstos como entrada para herramientas como **db2adv**is o para utilizar planes de acceso (del programa de utilidad de Explain) como ayuda para determinar el uso de tablas, columnas e índices para un conjunto de consultas.

Puede recopilar información acerca de una actividad especificando COLLECT ACTIVITY DATA para la clase de servicio, carga de trabajo o acción de trabajo a la que pertenece este tipo de actividad o un umbral que ese tipo de actividad podría infringir. La información se recopila cuando se ha completado la actividad, con independencia de si ésta se ha completado correctamente.

Tenga en cuenta que si un supervisor de sucesos de actividades está activo cuando se desactiva la base de datos, los registros de actividad anotados en la cola se descartarán. Para asegurarse de que obtiene todos los registros del supervisor de sucesos de actividades y de que no se descarta ninguno, primero desactive explícitamente el supervisor de sucesos de actividades, antes de desactivar la base de datos. Cuando un supervisor de sucesos de actividades se desactiva explícitamente, todos los registros de actividad anotados en la cola se procesan antes de que se desactive el supervisor de sucesos.

### Supervisor de sucesos de violaciones de umbral

Este supervisor captura información cuando se ha infringido un umbral. Indica qué umbral se ha infringido, la actividad que ha dado lugar a la infracción y qué acción estaba realizándose cuando se produjo.

Si especifica COLLECT ACTIVITY DATA para el umbral y se ha creado y está activo un supervisor de sucesos de actividades, también se recopilará información acerca de las actividades que han infringido el umbral, pero esta información se recopilará cuando finalice la actividad (correcta o incorrectamente).

Puede obtener detalles acerca de un umbral consultando la vista SYSCAT.THRESHOLDS.

### Supervisor de sucesos de estadísticas

Este supervisor sirve de alternativa de baja actividad general para capturar información de actividad detallada mediante la recopilación de datos de agregado (por ejemplo, el número de actividades que se han completado y el promedio de tiempo de ejecución). Los datos de agregado incluyen histogramas de diversas mediciones de actividad, entre las que se incluyen la vida útil, el tiempo en cola, el tiempo de ejecución y el coste estimado.

Puede utilizar los histogramas para entender la distribución de valores, identificar valores atípicos y calcular estadísticas adicionales, como los promedios y las desviaciones estándar. Por ejemplo, los histogramas pueden ayudarle a entender la variación en la vida útil que experimentan los usuarios. El promedio de tiempo de vida útil por sí solo no refleja qué experimenta un usuario si existe un alto grado de variabilidad. Consulte “Recopilación de las estadísticas de gestión de carga de trabajo utilizando un supervisor de sucesos de estadísticas” en la página 288 para obtener una descripción de cómo enviar estadísticas al supervisor de sucesos.

En la siguiente figura se muestran las distintas opciones de supervisión que están disponibles para acceder a la información de carga de trabajo: funciones de tabla para acceder a las estadísticas en tiempo real y detalles de actividad e información histórica capturada como agregados eficientes o como detalles acerca de actividades individuales:

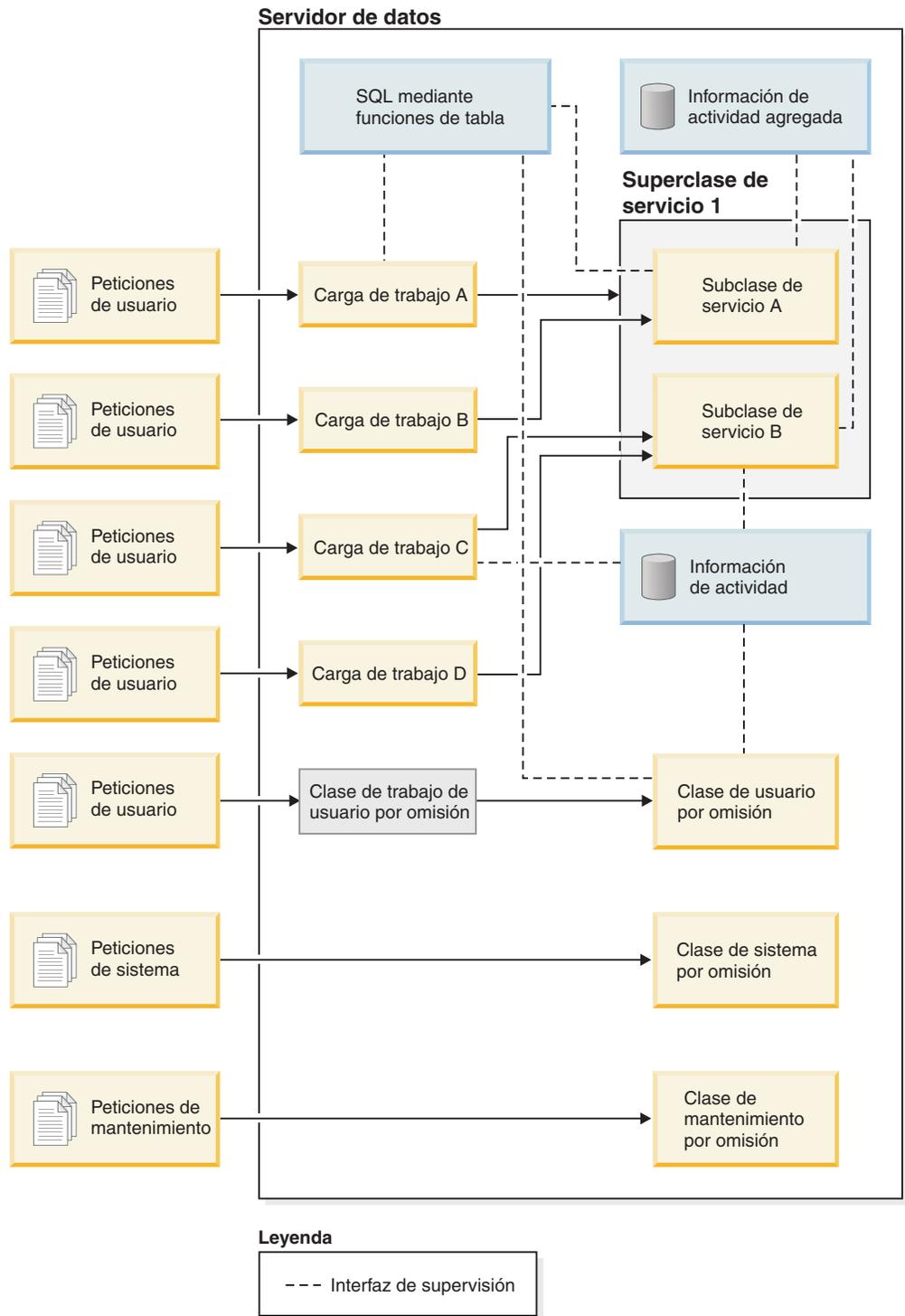


Figura 46. Gestión de carga de trabajo con supervisión

A diferencia de los supervisores de sucesos de transacción, sentencia y conexión, los supervisores de sucesos de violaciones de umbral, de actividad y de estadísticas no tienen condiciones de sucesos (es decir, condiciones que se especifican en la palabra clave WHERE de la sentencia CREATE EVENT MONITOR). En lugar de ello, estos supervisores de sucesos dependen de los atributos de las clases de servicio, cargas de trabajo, clases de trabajo y umbrales

para determinar si estos objetos han de enviar su información de actividad o información de agregado a estos supervisores.

Por lo general, los supervisores de sucesos graban datos en tablas o en archivos. Debe podar estas tablas o archivos periódicamente porque no se podan automáticamente.

Puede utilizar el script `wlmevmon.ddl` del directorio `sql1lib/misc` para crear y habilitar tres supervisores de sucesos, que se denominan `DB2ACTIVITIES`, `DB2STATISTICS` y `DB2THRESHOLDVIOLATIONS`. Si es necesario, modifique el script para cambiar el espacio de tablas u otros parámetros.

## Ejemplo

**Ejemplo: Identificar las consultas con un coste estimado grande utilizando el supervisor de sucesos de estadísticas:** Sospecha que la carga de trabajo de la base de datos en ocasiones incluye consultas grandes, poco rentables, posiblemente debido a una pobre optimización de las propias consultas. Desea identificar estas consultas para poder evitar que generen un consumo excesivo de recursos del sistema, con el objetivo a largo plazo de, quizás, volver a grabar algunas de las consultas para mejorar el rendimiento. El supervisor de sucesos de estadísticas le proporciona un método de baja actividad general que le permite medir el coste estimado de las consultas, que podrá utilizar para determinar cuál debería ser el coste estimado máximo aceptable para una consulta en el servidor de datos. Por lo general, una consulta pobremente optimizada se distingue por tener un coste estimado grande, que es muchas veces más grande que el coste estimado de la mayoría de las demás consultas.

Para empezar, ha de crear y activar un supervisor de sucesos de estadísticas e iniciar la recopilación de datos de actividad de agregado ampliados para la clase de servicio en la que se ejecutan las consultas:

```
CREATE EVENT MONITOR DB2STATISTICS
  FOR STATISTICS WRITE TO TABLE

SET EVENT MONITOR DB2STATISTICS STATE 1
```

En este ejemplo, todas las consultas se ejecutan en la subclase `SYSDEFAULTSUBCLASS` de la clase de servicio `SYSDEFAULTUSERCLASS`, que puede alterar para recopilar los datos necesarios:

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA EXTENDED
```

Un día de trabajo completo podría proporcionar una aproximación razonable del rango de consultas que el servidor de datos suele procesar. Al final del día, puede copiar las estadísticas recopiladas de la memoria en el supervisor de sucesos de estadísticas ejecutando el procedimiento almacenado `WLM_COLLECT_STATS`:

```
CALL WLM_COLLECT_STATS()
```

Con las distintas estadísticas que aparecen en las tablas del supervisor de sucesos se incluyen las estadísticas de coste estimado de las consultas. Para verlas, puede consultar la tabla de estadísticas de clase de servicio `SCSTATS_DB2STATISTICS`:

```
SELECT STATISTICS_TIMESTAMP,
       COORD_ACT_EST_COST_AVG,
       COST_ESTIMATE_TOP
FROM SCSTATS_DB2STATISTICS
WHERE SERVICE_SUPERCLASS_NAME = 'SYSDEFAULTUSERCLASS'
  AND SERVICE_SUBCLASS_NAME = 'SYSDEFAULTSUBCLASS'
```

STATISTICS_TIMESTAMP	COORD_ACT_EST_COST_AVG	COST_ESTIMATE_TOP
2008-09-03-09.49.04.455979	169440	13246445

1 registro(s) seleccionado(s).

La salida muestra que el promedio de consulta tiene un coste estimado que está dentro del rango de los cientos de miles de timerons, y que las consultas más grandes tienen costes estimados de más de diez millones de timerons. Puede confirmar que las consultas que tienen diez millones o más de timerons son valores atípicos consultando el histograma del coste estimado, que se ha generado al mismo tiempo que se han grabado en la tabla del supervisor de sucesos el promedio y las marcas de límite superior que se muestran en la salida. Puede ver el histograma consultando la tabla HISTOGRAMBIN\_DB2STATISTICS, tal como se indica a continuación:

```
SELECT STATISTICS_TIMESTAMP,
       TOP,
       NUMBER_IN_BIN
FROM HISTOGRAMBIN_DB2STATISTICS HIST,
     SYSCAT.SERVICECLASSES SC
WHERE HIST.SERVICE_CLASS_ID = SC.SERVICECLASSID
      AND SC.PARENTSERVICECLASSNAME = 'SYSDEFAULTUSERCLASS'
      AND SC.SERVICECLASSNAME = 'SYSDEFAULTSUBCLASS'
      AND HISTOGRAM_TYPE = 'COORDACTESTCOST'
```

STATISTICS_TIMESTAMP	TOP	NUMBER_IN_BIN
2008-09-03-09.49.04.455979	1	0
2008-09-03-09.49.04.455979	2	0
2008-09-03-09.49.04.455979	3	0
2008-09-03-09.49.04.455979	5	0
2008-09-03-09.49.04.455979	8	0
2008-09-03-09.49.04.455979	12	1
2008-09-03-09.49.04.455979	19	0
2008-09-03-09.49.04.455979	29	0
2008-09-03-09.49.04.455979	44	2
2008-09-03-09.49.04.455979	68	5
2008-09-03-09.49.04.455979	103	22
2008-09-03-09.49.04.455979	158	14
2008-09-03-09.49.04.455979	241	54
2008-09-03-09.49.04.455979	369	2
2008-09-03-09.49.04.455979	562	142
2008-09-03-09.49.04.455979	858	21
2008-09-03-09.49.04.455979	1309	123
2008-09-03-09.49.04.455979	1997	512
2008-09-03-09.49.04.455979	3046	643
2008-09-03-09.49.04.455979	4647	201
2008-09-03-09.49.04.455979	7089	875
2008-09-03-09.49.04.455979	10813	1445
2008-09-03-09.49.04.455979	16493	5386
2008-09-03-09.49.04.455979	25157	2409
2008-09-03-09.49.04.455979	38373	8940
2008-09-03-09.49.04.455979	58532	9820
2008-09-03-09.49.04.455979	89280	2149
2008-09-03-09.49.04.455979	136181	798
2008-09-03-09.49.04.455979	207720	2411
2008-09-03-09.49.04.455979	316840	14989
2008-09-03-09.49.04.455979	483283	9831
2008-09-03-09.49.04.455979	737162	1451
2008-09-03-09.49.04.455979	1124409	213
2008-09-03-09.49.04.455979	1715085	24
2008-09-03-09.49.04.455979	2616055	1
2008-09-03-09.49.04.455979	3990325	0
2008-09-03-09.49.04.455979	6086529	0
2008-09-03-09.49.04.455979	9283913	0

2008-09-03-09.49.04.455979	14160950	3
2008-09-03-09.49.04.455979	21600000	0
2008-09-03-09.49.04.455979	-1	0

En el histograma, el valor de la columna number\_in\_bin para las consultas cuyo valor superior es mayor que 2.616.055 es cero hasta que el valor superior alcanza 14.160.950, donde number\_in\_bin pasa a ser 3. Estas tres consultas son valores atípicos y pueden controlarse con un umbral ESTIMATEDSQLCOST establecido para activarse si el coste estimado de una consulta excede de 10 millones de timerons, que podrá utilizar para evitar la ejecución de esas actividades y para supervisarlas más de cerca.

**Ejemplo: Utilización del supervisor de sucesos de violaciones de umbral:** Para controlar las actividades de un coste estimado determinado, desea definir un umbral ESTIMATEDSQLCOST en la carga de trabajo que se aplique sólo al subconjunto de la carga de trabajo total que excede un determinado coste estimado. Tras haber consultado el histograma de coste estimado, ha determinado que las actividades que tienen un coste estimado dentro del rango 0 a menos de 3 millones de timerons se producen con frecuencia, y que las actividades que tienen un coste estimado de más de 10 millones de timerons se producen raramente (quizás sólo durante algunas horas del día y quizás debido a algún error en la consulta, como la utilización de una unión cartesiana).

Para verificar que un umbral de 10 millones de timerons es efectivo al detener esas pocas actividades un día que no ha de permitirse que se ejecuten, puede crear y activar un supervisor de sucesos de umbral:

```
CREATE THRESHOLD TH1
  FOR DATABASE ACTIVITIES
  ENFORCEMENT DATABASE
  WHEN ESTIMATEDSQLCOST > 10000000
  STOP EXECUTION

CREATE EVENT MONITOR DB2THRESHOLDVIOLATIONS
  FOR THRESHOLD VIOLATIONS
  WRITE TO TABLE

SET EVENT MONITOR DB2THRESHOLDVIOLATIONS STATE 1
```

Al finalizar el día, puede ver las violaciones de umbral que se han producido consultando la tabla de violaciones de umbral:

```
SELECT THRESHOLDID,
       SUBSTR(THRESHOLD_PREDICATE, 1, 20) PREDICATE,
       TIME_OF_VIOLATION,
       THRESHOLD_MAXVALUE,
       THRESHOLD_ACTION
FROM THRESHOLDVIOLATIONS_DB2THRESHOLDVIOLATIONS
ORDER BY TIME_OF_VIOLATION, THRESHOLDID
```

THRESHOLDID	PREDICATE	TIME_OF_VIOLATION	THRESHOLD_MAXVALUE	THRESHOLD_ACTION
1	EstimatedSQLCost	2008-09-02-22.39.10.000000	10000000	Stop

1 registro(s) seleccionado(s).

### Ejemplo: Utilización del supervisor de sucesos de actividad

En el ejemplo anterior se mostraba cómo podía recopilar información de umbral en una tabla del supervisor de sucesos para confirmar que la ejecución de las actividades con un coste estimado grande se evita mediante un umbral. Tras ver estas violaciones de umbral, desea determinar qué texto de la sentencia de SQL

produce estas consultas grandes, por lo que puede utilizar el recurso Explain para determinar si un índice es necesario en las tablas que se consultan.

La recopilación de esta información adicional requiere la creación y activación de un supervisor de sucesos de actividad y la alteración del umbral para activar la recopilación de actividades con detalles:

```
CREATE EVENT MONITOR DB2ACTIVITIES
  FOR ACTIVITIES WRITE TO TABLE

SET EVENT MONITOR DB2ACTIVITIES STATE 1

ALTER THRESHOLD TH1
  WHEN EXCEEDED
  COLLECT ACTIVITY DATA WITH DETAILS
```

Cuando vuelva a consultar la tabla de violaciones de umbral tras haber transcurrido otro día de trabajo, puede realizar una unión con la tabla ACTIVITYSTMT\_DB2ACTIVITIES para ver el texto de la sentencia de SQL de cualquier actividad que ha infringido el umbral:

```
SELECT THRESHOLDID,
       SUBSTR(THRESHOLD_PREDICATE, 1, 20) PREDICATE,
       TIME_OF_VIOLATION,
       SUBSTR(STMT_TEXT,1,70) STMT_TEXT
FROM THRESHOLDVIOLATIONS_DB2THRESHOLDVIOLATIONS TV,
     ACTIVITYSTMT_DB2ACTIVITIES A
WHERE TV.APPL_ID = A.APPL_ID
      AND TV.UOW_ID = A.UOW_ID
      AND TV.ACTIVITY_ID = A.ACTIVITY_ID
```

THRESHOLDID	PREDICATE	TIME_OF_VIOLATION	STMT_TEXT
1	EstimatedSQLCost	2008-09-02-23.04.49.000000	select count(*) from syscat.tables,syscat.tables,syscat.tables

1 registro(s) seleccionado(s).

## Datos de supervisión disponibles

Los datos de supervisión están disponibles en cargas de trabajo, subclases de servicio y superclases de servicio, clases de trabajo y colas de umbrales. Puede utilizar estos datos para diagnosticar y corregir problemas y para ajustar el rendimiento.

### Datos de supervisión de cargas de trabajo

En la figura siguiente se muestra la información de supervisión disponibles para cargas de trabajo. Puede recopilar estadísticas de cargas de trabajo e información sobre actividades que se ejecuten en las cargas de trabajo utilizando los supervisores de sucesos. El caso de las cargas de trabajo, también puede obtener estadísticas agregadas de actividades. Puede acceder a estadísticas de cargas de trabajo e información sobre apariciones de la carga de trabajo en tiempo real utilizando funciones de tabla.

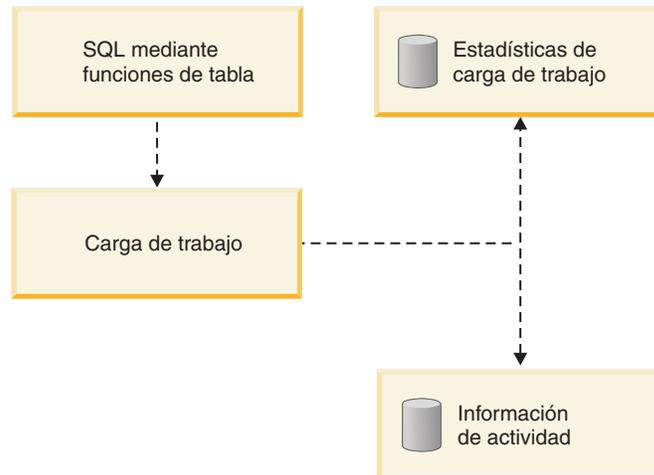


Figura 47. Datos de supervisión disponibles para cargas de trabajo

### Datos de supervisión de clases de servicio

En la figura siguiente se muestra la información de supervisión disponibles para clases de servicio. Puede recopilar estadísticas para subclases de servicio y superclases de servicio. En el caso de las subclases servicio, también puede obtener estadísticas agregadas de actividades y peticiones, además de información sobre las actividades que se ejecutan en la subclase de servicio. Puede acceder a las estadísticas y la información de subclases de servicio y superclases de servicio sobre los agentes que se ejecutan en una determinada clase de servicio en tiempo real utilizando las funciones de tabla.

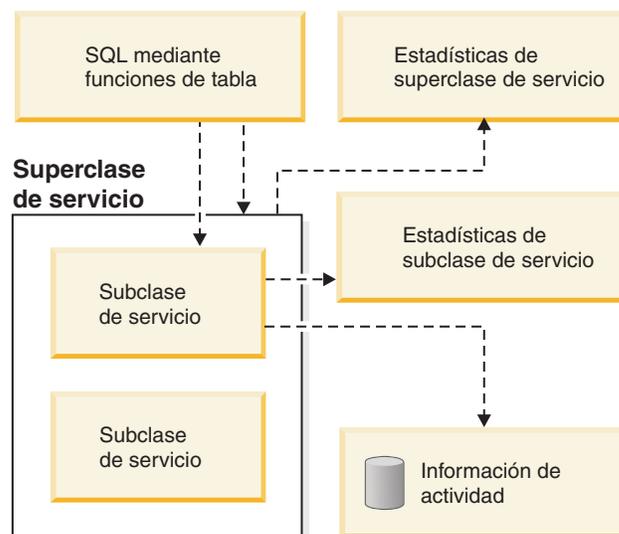


Figura 48. Datos de supervisión disponibles para clases de servicio

### Datos de supervisión de clases de trabajo

En la figura siguiente se muestra la información de supervisión disponibles para clases de trabajo. Puede recopilar estadísticas e información de clases de trabajo acerca de las actividades asociadas a una determinada clase de trabajo. Puede acceder a las estadísticas de las clases de trabajo en tiempo real utilizando las

funciones de tabla.

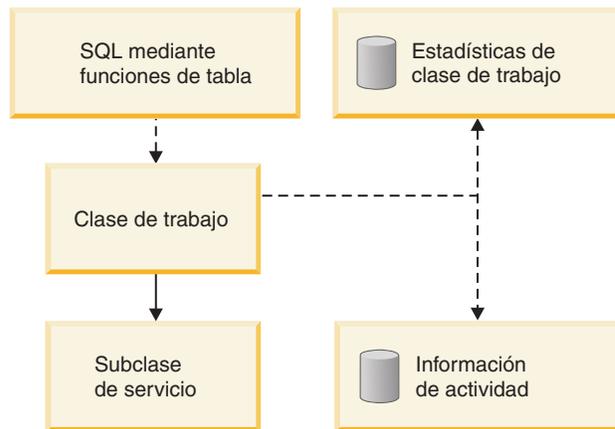


Figura 49. Datos de supervisión disponibles para clases de trabajo

## Datos de supervisión de umbrales

En la figura siguiente se muestra la información de supervisión disponibles para umbrales. Puede obtener información sobre violaciones de umbral, las actividades que han provocado las violaciones de umbral y estadísticas de colocación en cola (para umbrales de colocación en cola). Puede acceder a las estadísticas de umbrales de colocación en cola en tiempo real utilizando las funciones de tabla.

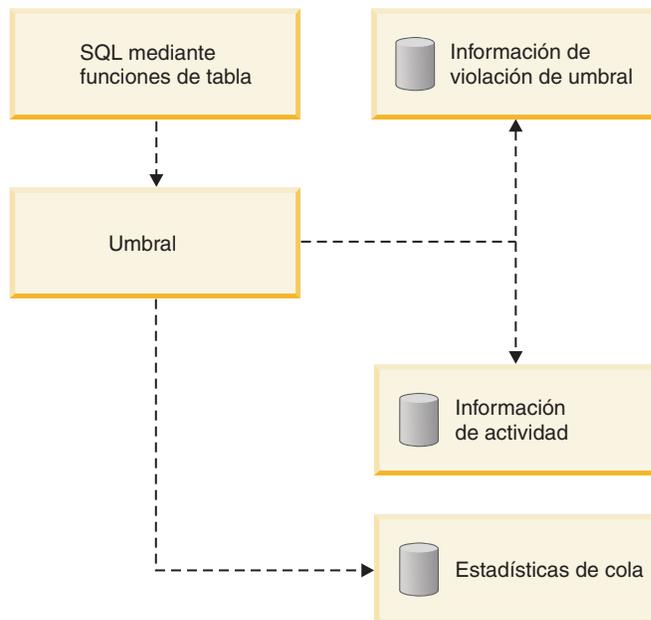


Figura 50. Datos de supervisión disponibles para umbrales

---

## Procedimientos almacenados de gestión de carga de trabajo de DB2

Puede utilizar procedimientos almacenados para cancelar una actividad, capturar detalles sobre una actividad, restablecer las estadísticas sobre objetos de gestión de carga de trabajo de DB2 y definir información del cliente en el servidor de datos.

Los procedimientos almacenados siguientes están disponibles para utilizarlos con la gestión de carga de trabajo de DB2:

**WLM\_CANCEL\_ACTIVITY**(*descriptor\_contexto\_aplicación, ID\_uow, ID\_actividad*)

Utilice este procedimiento almacenado para cancelar una actividad en ejecución o en cola. Puede identificar la actividad por su descriptor de contexto de aplicación, su identificador de unidad de trabajo y su identificador de actividad. Puede cancelar cualquier tipo de actividad. La aplicación de la actividad cancelada recibe el error SQL4725N.

**WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS**(*descriptor\_contexto\_aplicación, ID\_uow, ID\_actividad*)

Utilice este procedimiento almacenado para enviar al supervisor de sucesos de actividades información sobre una determinada actividad que se está ejecutando actualmente. Este procedimiento almacenado envía la información inmediatamente, en lugar de esperar a que se complete la actividad.

**WLM\_COLLECT\_STATS()**

Utilice este procedimiento almacenado para recopilar y restablecer estadísticas para objetos de la gestión de carga de trabajo de DB2. Todas las estadísticas que son objeto de seguimiento para clases de servicio, cargas de trabajo, colas de umbrales y conjuntos de acciones de trabajo se envían al supervisor de sucesos de estadísticas activo (si existe alguno) y se restablecen. Si no hay ningún supervisor de sucesos de estadísticas activo, las estadísticas sólo se restablecen, pero no se recopilan.

**WLM\_SET\_CLIENT\_INFO**(*ID\_usuario\_cliente, nombre\_estación\_trabajo\_cliente, nombre\_aplicación\_cliente, serie\_contabilidad\_cliente, carga\_trabajo\_cliente*)

Utilice este procedimiento para establecer los atributos de información de cliente que se utilizan en el servidor de datos con el fin de registrar la identidad de la aplicación o el usuario final que actualmente utiliza la conexión. En los casos en los que exista middleware entre las aplicaciones o los usuarios y su servidor de datos, utilice el procedimiento **WLM\_SET\_CLIENT\_INFO** para establecer explícitamente atributos de conexión diferenciadores.

---

## Estadísticas para objetos de la gestión de carga de trabajo de DB2

Se mantienen estadísticas para los objetos de la gestión de carga de trabajo de DB2, incluidas clases de servicio, clases de trabajo, cargas de trabajo y colas de umbrales. Estas estadísticas residen en la memoria y pueden visualizarse en tiempo real utilizando funciones de tabla de estadísticas de la gestión de carga de trabajo de DB2, o bien se pueden recopilar estadísticas y enviarse a un supervisor de sucesos de estadísticas, donde pueden visualizarse posteriormente para su análisis histórico.

Tenga en cuenta que también puede obtener métricas de supervisión a través del supervisor de sucesos de estadísticas. Éstas no se tratan este tema, que cubre únicamente aquellas estadísticas específicas de la gestión de carga de trabajo de DB2.

Cuando se envían estadísticas al supervisor de sucesos, los valores de la memoria se restablecen para impedir que se recopilen datos duplicados en posteriores intervalos de recopilación. Puesto que las funciones de tabla de estadísticas de la gestión de carga de trabajo de DB2 informan de los valores actuales en la memoria, después de una recopilación informan de los valores restablecidos. Las funciones

de tabla de la gestión de carga de trabajo de DB2 sólo informan de un subconjunto de las estadísticas. Para visualizar el conjunto completo de estadísticas, debe recopilar las estadísticas y enviarlas al supervisor de sucesos de estadísticas.

### **Recopilación de estadísticas de datos de actividad agregados**

Se mantienen las estadísticas siguientes sobre los objetos especificados en cada miembro, independientemente del valor de la opción COLLECT AGGREGATE ACTIVITY DATA especificada para esos objetos cuando se crean o se modifican.

Tabla 52. Estadísticas recopiladas para objetos de la base de datos independientemente del valor de COLLECT AGGREGATE ACTIVITY DATA

Objeto de base de datos	Estadística	Descripción
Subclases de servicio	<ol style="list-style-type: none"> <li>1. Máximo de actividades simultáneas (concurrent_act_top)</li> <li>2. Total de actividades de coordinador completadas (coord_act_completed_total)</li> <li>3. Total de actividades de coordinador finalizadas anormalmente (coord_act_aborted_total)</li> <li>4. Total de actividades de coordinador rechazadas (coord_act_rejected_total)</li> <li>5. Número de peticiones activas (num_requests_active)</li> <li>6. Número de actividades incluidas y excluidas mediante correlación (act_remapped_in y act_remapped_out)</li> </ol>	<ol style="list-style-type: none"> <li>1. Utilice esta marca de límite superior de simultaneidad de actividades para determinar el número más alto de actividades simultáneas (incluidas las actividades anidadas) alcanzado en un miembro para una clase de servicio en el intervalo de tiempo durante el que se recopila la estadística.</li> <li>2. Utilice esta estadística para determinar cuánto trabajo se realiza en una clase de servicio.</li> <li>3. Utilice esta estadística, que mide la finalización anómala de actividades, para determinar el grado de salud del sistema. Las actividades pueden finalizar anormalmente debido a una cancelación, a errores o a umbrales reactivos.</li> <li>4. Utilice este recuento de actividades de coordinador no anidadas rechazadas, que mide el rechazo de actividades, para obtener una indicación de la utilidad de la política de rechazo. Las actividades se cuentan como rechazadas cuando infringen un umbral predictivo que tiene una acción STOP EXECUTION o cuando una acción de trabajo impide su ejecución.</li> <li>5. Utilice esta estadística para determinar el número de peticiones que se ejecutan actualmente en una clase de servicio.</li> <li>6. Utilice estas estadísticas para determinar el número de actividades incluidas o excluidas mediante una nueva correlación en una subclase de servicio como parte de la disminución de la prioridad de las actividades en curso.</li> </ol>
Superclases de servicio	Máximo de conexiones simultáneas (concurrent_connection_top)	Utilice esta marca de límite superior de simultaneidad de conexiones de coordinador para ajustar un umbral de simultaneidad de conexiones.

Tabla 52. Estadísticas recopiladas para objetos de la base de datos independientemente del valor de COLLECT AGGREGATE ACTIVITY DATA (continuación)

Objeto de base de datos	Estadística	Descripción
Cargas de trabajo	<ol style="list-style-type: none"> <li>1. Máximo de apariciones de carga de trabajo simultáneas (concurrent_wlo_top)</li> <li>2. Máximo de actividades simultáneas en apariciones de la carga de trabajo (concurrent_wlo_act_top)</li> <li>3. Total de actividades de coordinador completadas (coord_act_completed_total)</li> <li>4. Total de actividades de coordinador finalizadas anormalmente (coord_act_aborted_total)</li> <li>5. Total de actividades de coordinador rechazadas (coord_act_rejected_total)</li> <li>6. Total de apariciones completadas de la carga de trabajo (wlo_completed_total )</li> <li>7. Total de actividades (act_total)</li> </ol>	<ol style="list-style-type: none"> <li>1. Utilice esta marca de límite superior de apariciones de la carga de trabajo para identificar el número máximo de apariciones simultáneas de la carga de trabajo con el fin de ayudar a ajustar o establecer un umbral de simultaneidad de apariciones de la carga de trabajo si el número de apariciones de la carga de trabajo que hay simultáneamente en ejecución es demasiado alto (es decir, demasiadas aplicaciones asociadas a la misma definición de carga de trabajo se ejecutan en el sistema al mismo tiempo).</li> <li>2. Utilice este elemento para conocer el número más alto de actividades simultáneas que se ha alcanzado en un miembro para cualquier instancia de esta carga de trabajo en el intervalo de tiempo de la recopilación.</li> <li>3. Utilice esta estadística, que mide la tasa de finalización correcta de las actividades, para obtener una indicación de la salud del sistema.</li> <li>4. Utilice esta estadística, que mide la finalización anómala de actividades, para determinar el grado de salud del sistema. Las actividades pueden finalizar anormalmente a causa de una cancelación, errores o umbrales reactivos.</li> <li>5. Utilice esta estadística, que mide la tasa de rechazo de actividades, para determinar la utilidad de una política de rechazo. Las actividades se cuentan como rechazadas cuando infringen un umbral predictivo que tiene una acción STOP EXECUTION o cuando una acción de trabajo impide su ejecución.</li> <li>6. Utilice esta estadística para determinar cuántas apariciones de una carga de trabajo se completan en un período de tiempo específico.</li> <li>7. Utilice esta estadística para determinar la efectividad del conjunto de acciones de trabajo y determinar los porcentajes relativos de los tipos de actividades en el sistema.</li> </ol>

Tabla 52. Estadísticas recopiladas para objetos de la base de datos independientemente del valor de COLLECT AGGREGATE ACTIVITY DATA (continuación)

Objeto de base de datos	Estadística	Descripción
Clase de trabajo (a través de una acción de trabajo)	Total de asignaciones de cola (queue_assignments_total)	Utilice esta estadística para determinar si se está produciendo una colocación en cola excesiva, o bien si se está colocando en cola el número correcto de actividades (es decir, si el umbral de simultaneidad es demasiado restrictivo o no lo es lo suficiente).
Colas de umbrales	<ol style="list-style-type: none"> <li>Máximo tamaño de cola (queue_size_top)</li> <li>Tiempo total en cola (queue_time_total)</li> </ol>	<ol style="list-style-type: none"> <li>Utilice esta estadística para ayudar a determinar el tamaño de cola máximo e identificar si el tamaño de cola es suficiente.</li> <li>Utilice esta estadística para determinar cuánto tiempo pasan las actividades en la cola y si ese tiempo es excesivo.</li> </ol>

Cuando establece el valor de la opción COLLECT AGGREGATE ACTIVITY DATA en BASE para una subclase de servicio, una carga de trabajo o una clase de trabajo (a través de una acción de trabajo), también se recopilan algunas de las estadísticas siguientes, o bien se generan los histogramas correspondientes por cada miembro. Utilice los promedios para saber rápidamente dónde pasan la mayor parte del tiempo las actividades (por ejemplo, en cola o en ejecución) y el tiempo de respuesta (vida útil). También puede utilizar los promedios para ajustar las plantillas del histograma. Es decir, puede comparar un promedio real con el promedio calculado a partir de un histograma y, si el promedio del histograma difiere del promedio real, puede plantearse modificar la plantilla del histograma correspondiente utilizando un conjunto de intervalos que sean más apropiados para sus datos.

Tabla 53. Estadísticas o histogramas recopilados cuando se establece COLLECT AGGREGATE ACTIVITY DATA en BASE

Estadística o histograma	Descripción
Promedio de tiempo de ejecución de peticiones (request_exec_time_avg)	Utilice esta estadística para determinar la media aritmética de los tiempos de ejecución de las peticiones asociadas a una clase de servicio.
Promedio de vida útil de actividades de coordinador (coord_act_lifetime_avg)	Utilice esta estadística para determinar la media aritmética de vida útil de las actividades de coordinador no anidadas asociadas a una clase de servicio, una carga de trabajo o una clase de trabajo.
Promedio de tiempo de ejecución de actividades de coordinador (coord_act_exec_time_avg)	Utilice esta estadística para determinar la media aritmética del tiempo de ejecución correspondiente a las actividades de coordinador no anidadas asociadas a una clase de servicio, una carga de trabajo o una clase de trabajo.

Tabla 53. Estadísticas o histogramas recopilados cuando se establece COLLECT AGGREGATE ACTIVITY DATA en BASE (continuación)

Estadística o histograma	Descripción
Promedio de tiempo en cola de actividades de coordinador (coord_act_queue_time_avg)	Utilice esta estadística para determinar la media aritmética de tiempo en cola de las actividades de coordinador no anidadas asociadas a una clase de servicio, una carga de trabajo o una clase de trabajo.
Coste estimado máximo (cost_estimate_top)	Utilice esta estadística para ajustar los umbrales de costes estimados.
Máximo de filas reales devueltas (rows_returned_top)	Utilice esta información para ajustar los umbrales de filas reales devueltas.
Máximo de espacio de tablas temporal agregado (agg_temp_tablespace_top)	<p>Utilice esta estadística para ajustar el uso de espacio de tablas temporal agregado del sistema.</p> <p>Esta estadística sólo se supervisa si se define un umbral para el uso de espacio de tablas temporal agregado. Para cualquier subclase de servicio, esta estadística se supervisa siempre que se defina un umbral AGGSQLTEMPSPACE en la misma subclase de servicio, o cuando se defina un umbral similar en cualquier subclase de servicio dentro de la misma superclase.</p>
Máximo de espacio de tablas temporal (temp_tablespace_top)	<p>Utilice esta estadística para ajustar los umbrales de uso de espacio de tablas temporal.</p> <p>Esta estadística sólo se supervisa si se define un umbral para el uso de espacio de tablas temporal. Para cualquier subclase de servicio, este umbral también se supervisa siempre que se defina un umbral AGGSQLTEMPSPACE en la misma subclase de servicio, o cuando se defina un umbral similar en cualquier subclase de servicio dentro de la misma superclase.</p>
Histograma de vida útil de las actividades de coordinador (CoordActLifetime)	<p>Utilice este histograma para obtener una visión del rendimiento general del sistema.</p> <p>Este histograma recopila el intervalo de tiempo transcurrido entre la hora de llegada y la hora de finalización de las actividades de coordinador no anidadas.</p> <p>Si la actividad es una rutina que deja un cursor abierto después de finalizar, el histograma de vida útil no cuenta la vida útil del cursor en relación con la vida útil de la rutina que es el padre del cursor.</p>

Tabla 53. Estadísticas o histogramas recopilados cuando se establece COLLECT AGGREGATE ACTIVITY DATA en BASE (continuación)

Estadística o histograma	Descripción
Histograma de ejecución de actividades de coordinador (CoordActExecTime)	<p>Utilice este histograma para medir el impacto de los cambios en el sistema que afectan al tiempo de ejecución.</p> <p>Este histograma recopila el tiempo de ejecución para actividades de coordinador no anidadas.</p> <p>El tiempo de ejecución se calcula del modo siguiente:</p> <ul style="list-style-type: none"> <li>• Para los cursores, el tiempo de ejecución es el tiempo combinado para la petición de apertura de cursor, las captaciones y la petición de cierre de cursor. No se cuenta como tiempo de ejecución el tiempo que el cursor pasa desocupado.</li> <li>• Para las rutinas, el tiempo de ejecución va del principio al final de la invocación de la rutina. Si la rutina deja algún cursor abierto después de finalizar, la vida útil de estos cursores no se cuenta como tiempo de ejecución de la rutina.</li> <li>• Para todas las demás actividades, el tiempo de ejecución es la diferencia entre la vida útil de la actividad y el tiempo que la actividad pasa en cola.</li> </ul>
Histograma de tiempo en cola de actividades de coordinador (CoordActQueueTime)	<p>Utilice este histograma para medir el impacto de los umbrales de cola sobre las actividades.</p> <p>Este histograma recopila la cantidad que tiempo que pasan en cola las actividades de coordinador no anidadas.</p>

Cuando se establece el valor de la opción COLLECT AGGREGATE ACTIVITY DATA en EXTENDED para una subclase de servicio, una carga de trabajo o una clase de trabajo, se recopilan las siguientes estadísticas del sistema o se generan histogramas por cada miembro para la correspondiente clase de servicio o clase de trabajo (a través de una acción de trabajo). Utilice los promedios para conocer rápidamente la velocidad media de llegada de actividades (la velocidad de llegada es el valor opuesto al tiempo entre llegadas) y el gasto de actividades (coste estimado). También puede utilizar los promedios para ajustar las plantillas del histograma. Es decir, puede comparar un promedio real con el promedio calculado a partir de un histograma y, si el promedio del histograma difiere del promedio real, puede plantearse modificar la plantilla del histograma correspondiente utilizando un conjunto de intervalos que sean más apropiados para sus datos. Las estadísticas EXTENDED son útiles para modelar el rendimiento de forma más detallada. Consulte también “Modelado del rendimiento de gestión de carga de trabajo” en la página 303.

Tabla 54. Estadísticas o histogramas recopilados cuando se establece COLLECT AGGREGATE ACTIVITY DATA en EXTENDED

Estadística o histograma	Descripción
Promedio de coste estimado de actividades de coordinador (coord_act_est_cost_avg)	Utilice esta estadística para determinar la media aritmética de los costes estimados de las actividades de DML de coordinador en el nivel de anidamiento 0 que están asociados a esta subclase de servicio, carga de trabajo o clase de trabajo desde el último restablecimiento de las estadísticas.
Promedio de tiempo entre llegadas de actividades de coordinador no anidadas (coord_act_interarrival_time_avg)	Utilice esta estadística para determinar la media aritmética del tiempo entre la llegada de una actividad de coordinador en el nivel de anidamiento 0 asociada a esta clase de servicio, carga de trabajo o clase de trabajo, y la siguiente actividad de coordinación que llega. El promedio se calcula a partir del último restablecimiento de las estadísticas.
Histograma de coste estimado de actividades de coordinador (CoordActEstCost)	Utilice este histograma para obtener una distribución aproximada del tiempo de servicio.  Este histograma recopila el coste estimado para actividades de coordinador no anidadas. Estos datos son útiles para modelar su sistema o para suministrarlos a aplicaciones de modelado del rendimiento.
Histograma de tiempo entre llegadas de actividades de coordinador(CoordActInterArrivalTime)	Utilice este histograma para obtener la distribución de tiempo entre llegadas para actividades de coordinador no anidadas.  Este histograma recopila el tiempo entre llegadas para actividades de coordinador no anidadas. Estos datos son útiles para modelar su sistema o para suministrarlos a aplicaciones de modelado del rendimiento.

En la tabla siguiente se proporciona una referencia de las estadísticas de actividades que se recopilan para cada objeto de la gestión de carga de trabajo de DB2, y se incluyen todas las estadísticas agregadas disponibles tanto de funciones de tabla como de supervisores de sucesos. Algunas estadísticas se recopilan siempre para algunos objetos. Otras estadísticas sólo se recopilan cuando se especifica una determinada opción COLLECT AGGREGATE. Para estadísticas de actividades agregadas, si se especifica COLLECT AGGREGATE ACTIVITY DATA EXTENDED, se recopilarán también todas las estadísticas de actividades agregadas BASE.

Tabla 55. Recopilación de estadísticas de actividades agregadas para objetos de la gestión de carga de trabajo de DB2

Tipo de objeto	Estadísticas de actividades que por omisión se recopilan siempre	Estadísticas de actividades que se recopilan cuando especifica COLLECT AGGREGATE ACTIVITY DATA BASE	Estadísticas de actividades que se recopilan cuando especifica COLLECT AGGREGATE ACTIVITY DATA EXTENDED
Subclase de servicio	act_remapped_in act_remapped_out concurrent_act_top coord_act_completed_total coord_act_rejected_total coord_act_aborted_total	agg_temp_tablespace_top coord_act_exec_time_avg coord_act_lifetime_avg coord_act_lifetime_top coord_act_queue_time_avg coord_act_lifetime_stddev coord_act_exec_time_stddev coord_act_queue_time_stddev CoordActLifetime histogram CoordActExecTime histogram CoordActQueueTime histogram cost_estimate_top rows_returned_top temp_tablespace_top	coord_act_est_cost_avg coord_act_interarrival_time_avg Histograma CoordActEstCost Histograma CoordActInterArrivalTime
Superclase de servicio	concurrent_connection_top	N/A	N/A
Carga de trabajo	concurrent_wlo_act_top concurrent_wlo_top coord_act_aborted_total coord_act_completed_total coord_act_rejected_total wlo_completed_total	coord_act_exec_time_avg coord_act_lifetime_top coord_act_lifetime_avg coord_act_queue_time_avg coord_act_lifetime_stddev coord_act_exec_time_stddev coord_act_queue_time_stddev CoordActLifetime histogram CoordActExecTime histogram CoordActQueueTime histogram cost_estimate_top rows_returned_top temp_tablespace_top	coord_act_est_cost_avg coord_act_interarrival_time_avg Histograma CoordActEstCost Histograma CoordActInterArrivalTime
Clase de trabajo (a través de una acción de trabajo)	act_total	agg_temp_tablespace_top coord_act_lifetime_top coord_act_lifetime_avg coord_act_exec_time_avg coord_act_queue_time_avg CoordActLifetime histogram CoordActExecTime histogram CoordActQueueTime histogram cost_estimate_top rows_returned_top temp_tablespace_top	coord_act_est_cost_avg coord_act_interarrival_time_avg Histograma CoordActEstCost Histograma CoordActInterArrivalTime
Umbral	N/A	N/A	N/A
Cola de umbral	queue_assignments_total queue_size_top queue_time_total	N/A	N/A

## Recopilación de estadísticas de datos de peticiones agregados

Cuando establece el valor de la opción COLLECT AGGREGATE REQUEST DATA en BASE para una subclase de servicio, se mantienen las estadísticas siguientes para la subclase de servicio.

*Tabla 56. Estadísticas o histogramas recopilados cuando se establece COLLECT AGGREGATE REQUEST DATA en BASE*

Estadística o histograma	Descripción
Promedio de tiempo de ejecución de peticiones (request_exec_time_avg)	Utilice esta estadística para comprender rápidamente el promedio de tiempo que se dedica a procesar cada petición en un miembro y para ayudar a ajustar la plantilla del histograma de tiempo de ejecución de peticiones correspondiente.

Tabla 56. Estadísticas o histogramas recopilados cuando se establece COLLECT AGGREGATE REQUEST DATA en BASE (continuación)

Estadística o histograma	Descripción
<p>Histograma de tiempo de ejecución de peticiones (ReqExecTime)</p>	<p>Utilice este histograma para comprender dónde se realiza el trabajo y si la distribución del trabajo entre miembros es uniforme.</p> <p>Este histograma indica el volumen de trabajo que se ejecuta en una clase de servicio y la distribución de este trabajo entre miembros de base de datos. El tiempo de ejecución para peticiones se recopila en un histograma por cada miembro y para todas las peticiones.</p> <p>Este histograma incluye las peticiones en el miembro coordinador y las subpeticiones tanto en los miembros que son coordinadores como los que no lo son (como peticiones RPC o peticiones de subagente SMP). Las peticiones incluidas pueden estar asociadas o no a una actividad: en este histograma se incluyen tanto las peticiones PREPARE como OPEN, por ejemplo, pero mientras que las peticiones OPEN se asocian siempre a una actividad de cursor, las peticiones PREPARE no forman parte de una actividad.</p> <p>El tiempo de ejecución de las peticiones se aproxima al esfuerzo que dedican los agentes a trabajar en una clase de servicio. Por ejemplo, los recuentos de actividades de coordinador pueden mostrar que la mayoría de las actividades de usuario se originan en un miembro, pero como parte del proceso de las actividades, el agente coordinador puede enviar subpeticiones a otra partición de la base de datos que realiza la mayor parte del trabajo.</p> <p>El histograma de tiempo de ejecución de peticiones puede ser útil a la hora de determinar el tamaño de las peticiones enviadas a un miembro, es decir, si el trabajo que se envía al miembro se compone principalmente de peticiones pequeñas o de peticiones grandes, o si no existe una distribución específica.</p> <p>Los histogramas de tiempo de ejecución de peticiones no deben utilizarse para el análisis del tiempo de respuesta de las actividades, puesto que las actividades pueden componerse de una serie de peticiones y subpeticiones, porque no existe una correlación unívoca entre tiempo de ejecución de actividades y peticiones, y porque no todas las peticiones se asocian a actividades.</p>

En la tabla siguiente se proporciona una referencia de las estadísticas de peticiones que se recopilan para cada objeto de la gestión de carga de trabajo de DB2, y se incluyen todas las estadísticas agregadas disponibles tanto de funciones de tabla como de supervisores de sucesos. Algunas estadísticas se recopilan siempre para algunos objetos. Otras estadísticas sólo se recopilan cuando se especifica la opción COLLECT AGGREGATE REQUEST DATA.

*Tabla 57. Recopilación de estadísticas de peticiones agregadas para objetos de la gestión de carga de trabajo de DB2*

Tipo de objeto	Estadísticas de peticiones que por omisión se recopilan siempre	Estadísticas de peticiones que se recopilan cuando especifica COLLECT AGGREGATE REQUEST DATA BASE
Subclase de servicio	num_requests_active	request_exec_time_avg request_exec_time_stddev request_exec_time_total Histograma ReqExecTime
Superclase de servicio	N/A	N/A
Carga de trabajo	N/A	N/A
Clase de trabajo (a través de una acción de trabajo)	N/A	N/A
Umbral	N/A	N/A
Cola de umbral	N/A	N/A

## Recopilación de estadísticas y supervisión con nueva correlación de actividades

El modo en que recopila estadísticas y realiza la supervisión se ve afectado por la nueva correlación dinámica de las actividades entre subclases de servicio.

La nueva correlación tiene lugar cuando una actividad que se ejecuta en una clase de servicio se traslada a otra clase de servicio para continuar su ejecución. Este cambio de correlación puede realizarse mediante umbrales, como CPUTIMEINSC, y puede constituir parte íntegra de la configuración de la gestión de carga de trabajo, como es el caso de la disminución de la prioridad.

### Estadísticas afectadas por una nueva correlación de actividades

Como excepción a la norma, el tiempo entre llegadas de actividades, el coste estimado y el tiempo en cola se asocian a la subclase en la que empieza a ejecutarse una actividad, en lugar de la subclase en la que finaliza su ejecución. Puesto que una nueva correlación de una actividad afecta a la recopilación de estadísticas de ambas subclases, puede contarse un número diferente de actividades en un histograma de tiempo entre llegadas, coste estimado o tiempo en cola que en un histograma de vida útil o de tiempo de ejecución.

Por ejemplo, supongamos que una actividad empieza a ejecutarse en la subclase de servicio A y posteriormente se vuelve a correlacionar con la subclase de servicio B, en la que finaliza su ejecución. El coste estimado de esta actividad se asocia a la subclase de servicio A, pero su vida útil se asocia a la subclase de servicio B. Como resultado, para la subclase A, se cuenta un elemento más en el histograma de

costes estimados que en el histograma de vida útil, y para la subclase de servicio B, se cuenta un elemento más en el histograma de vida útil que en el histograma de costes estimados.

Como segunda excepción a la norma, el elemento de supervisor **concurrent\_act\_top** puede actualizarse y atribuirse a cualquier subclase por la que pase una actividad. Además de que el elemento de supervisor se incrementa cuando se inicia una actividad y se reduce cuando finaliza una actividad, también se incrementa cuando se correlaciona una actividad a la subclase y se reduce cuando se anula la correlación de una actividad con la subclase (se correlaciona con una subclase diferente).

## Estadísticas sobre una nueva correlación de actividad

Puede utilizar dos elementos de supervisor para contar el número de actividades que entran o abandonan una subclase de servicio debido a una acción de nueva correlación: **act\_remapped\_in** y **act\_remapped\_out**. Los elementos de supervisor **act\_remapped\_in** y **act\_remapped\_out** cuentan el número de actividades para una determinada subclase en cualquier partición incluidas o excluidas de la correlación con la subclase desde el último restablecimiento. Puede utilizar estos elementos de supervisor para validar que la nueva correlación de actividades entre subclases de servicio se produzca según lo previsto.

Para determinar las subclases de servicio de origen y de destino mediante una acción de nueva correlación, puede hacer referencia al registro del supervisor de sucesos de violación de umbral, que incluye un ID de clase de servicio de destino (*destination\_service\_class\_id*). También puede determinar la clase de servicio de origen utilizando el registro de violación de umbral.

## Supervisión con nueva correlación de actividades

Volver a correlacionar actividades con subclases diferentes afecta el modo en que supervisa dichas actividades. Para asegurarse de que se recopilen todas las estadísticas para una actividad que empieza en una clase de servicio y que acaba en otra debido a una nueva correlación, active la recopilación de datos de actividades agregadas para la subclase de servicio en la que se empieza a ejecutar la actividad y la subclase de servicio en la que la actividad finaliza la ejecución cuando cree o modifique las clases de servicio. Si activa la recopilación de datos de actividades agregadas sólo para la subclase de servicio en la que se ha iniciado la actividad, la actividad contribuye sólo en las estadísticas de tiempo en cola y, en el caso de estadísticas ampliadas, en las estadísticas de coste estimado y tiempo entre llegadas. Si activa la recopilación de datos de actividades agregadas sólo para la subclase de servicio en la que la actividad finaliza la ejecución, la actividad sólo contribuye en las estadísticas de vida útil y tiempo de ejecución, independientemente de si la opción es **COLLECT AGGREGATE DATA BASE** o **COLLECT AGGREGATE DATA EXTENDED** cuando emite la sentencia **CREATE SERVICE CLASS** o **ALTER SERVICE CLASS**.

En las tablas siguientes se resume cómo se ven afectadas la recopilación de estadísticas por los valores de nueva correlación y recopilación.

Tabla 58. Efecto de la opción COLLECT AGGREGATE DATA BASE en una recopilación de estadísticas agregadas para subclases implicadas en una nueva correlación

Estadísticas	Valor de recopilación de la subclase de inicio y valor de recopilación de la subclase de finalización			
	NONE y NONE	BASE y NONE	NONE y BASE	BASE y BASE
Vida útil	No se recopilan	No se recopilan	Se recopilan	Se recopilan
Tiempo en cola	No se recopilan	Se recopilan	No se recopilan	Se recopilan
Tiempo de ejecución	No se recopilan	No se recopilan	Se recopilan	Se recopilan

Tabla 59. Efecto de la opción COLLECT AGGREGATE DATA EXTENDED en una recopilación de estadísticas agregadas para subclases implicadas en una nueva correlación

Estadísticas	Valor de recopilación de la subclase de inicio y valor de recopilación de la subclase de finalización			
	NONE y NONE	EXTENDED y NONE	NONE y EXTENDED	EXTENDED y EXTENDED
Vida útil	No se recopilan	No se recopilan	Se recopilan	Se recopilan
Tiempo en cola	No se recopilan	Se recopilan	No se recopilan	Se recopilan
Tiempo de ejecución	No se recopilan	No se recopilan	Se recopilan	Se recopilan
Tiempo entre llegadas	No se recopilan	Se recopilan	No se recopilan	Se recopilan
Coste estimado	No se recopilan	Se recopilan	No se recopilan	Se recopilan

Tabla 60. Efecto de combinar las opciones COLLECT AGGREGATE DATA BASE y COLLECT AGGREGATE DATA EXTENDED en una recopilación de estadísticas agregadas para subclases implicadas en una nueva correlación

Estadísticas	Valor de recopilación de la subclase de inicio y valor de recopilación de la subclase de finalización	
	BASE y EXTENDED	EXTENDED y BASE
Vida útil	Se recopilan	Se recopilan
Tiempo en cola	Se recopilan	Se recopilan
Tiempo de ejecución	Se recopilan	Se recopilan
Tiempo entre llegadas	No se recopilan	Se recopilan
Coste estimado	No se recopilan	Se recopilan

## Histogramas en la gestión de carga de trabajo

Un histograma es una colección de intervalos, que son contenedores para recopilar rangos de datos discretos. Los histogramas son útiles para gran diversidad de tareas de análisis de la carga de trabajo y de ajuste del rendimiento.

Los histogramas de la gestión de carga de trabajo de DB2 tienen un número fijo de 41 intervalos. El cuadragésimo intervalo contiene el valor definido más alto para el histograma, y el cuadragésimo primer intervalo se destina a los valores que superan el valor definido más alto. Cada intervalo representa un rango de valores específico, y los rangos de intervalos siguen una escala logarítmica donde cada intervalo representa un rango progresivamente mayor a medida que se va progresando desde el primero hasta el cuadragésimo intervalo. En la siguiente

figura se muestra un histograma de vida útil de actividades expresado mediante un diagrama de barras.

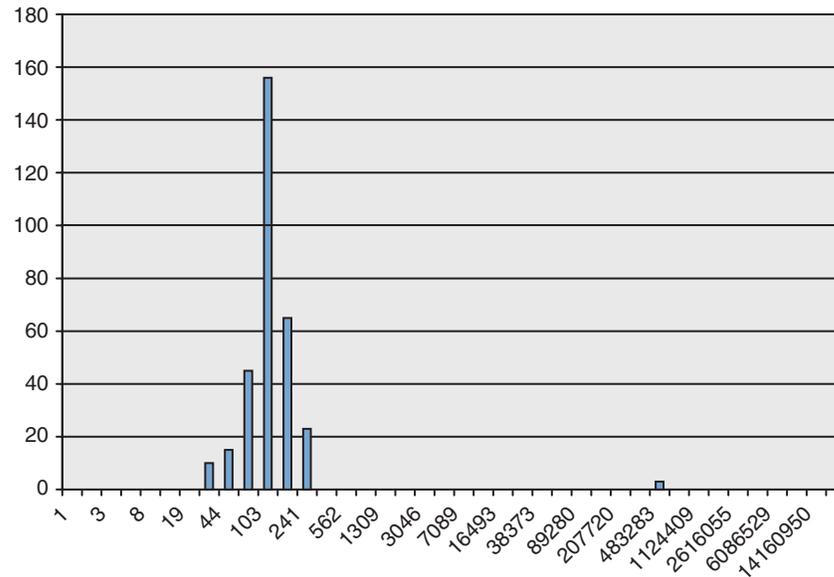


Figura 51. Histograma de vida útil de actividades expresado mediante un diagrama de barras

El histograma de vida útil de actividad corresponde a los datos siguientes. Cada recuento representa el número de actividades cuya vida útil (en milisegundos) se encuentra dentro del rango que comprende desde el valor de intervalo bajo hasta el valor de intervalo alto. Por ejemplo, 156 actividades tenían una vida útil comprendida dentro del rango de 68 milisegundos a 103 milisegundos.

Int. bajo	Int. alto	Recuento
0	1	0
1	2	0
2	3	0
3	5	0
5	8	0
8	12	0
12	19	0
19	29	10
29	44	15
44	68	45
68	103	156
103	158	65
158	241	23
241	369	0
369	562	0
562	858	0
858	1309	0
1309	1997	0
1997	3046	0
3046	4647	0
4647	7089	0
7089	10813	0
10813	16493	0
16493	25157	0
25157	38373	0
38373	58532	0
58532	89280	0
89280	136181	0
136181	207720	0
207720	316840	0

316840	483283	3
483283	737162	0
737162	1124409	0
1124409	1715085	0
1715085	2616055	0
2616055	3990325	0
3990325	6086529	0
6086529	9283913	0
9283913	14160950	0
14160950	21600000	0
21600000	Infinito	0

Puede utilizar histogramas para distintas finalidades. Por ejemplo, puede utilizarlos para ver la distribución de valores, utilizarlos para identificar valores aislados o utilizarlos para calcular promedios y desvíos estándar. Consulte “Caso de ejemplo: Ajuste de una configuración de la gestión de carga de trabajo de DB2 cuando no hay información de planificación de capacidad disponible” en la página 391 y “Ejemplo: Cálculo de promedios y desviación estándar de histogramas en una configuración de gestión de carga de trabajo de DB2” en la página 286 para ver ejemplos de cómo utilizar los histogramas para entender mejor y caracterizar la carga de trabajo.

En un entorno de base de datos de varios miembros, los histogramas se recopilan en cada miembro de base de datos. Los intervalos de histograma tiene el mismo rango de valores en todos los miembros de base de datos, con recuentos específicos por intervalo y por miembro. Puede utilizar los intervalos para analizar información relacionada con cada miembro individual. También puede combinar los histogramas de todos los miembros de base de datos sumando los recuentos de los intervalos correspondientes y utilizar este único histograma para obtener una vista global de los datos, que posteriormente podrá utilizar para tareas tales como el cálculo del promedio global y el desvío estándar.

Los histogramas están disponibles para las subclases de servicio, las cargas de trabajo y las clases de trabajo, por medio de acciones de trabajo. Los histogramas se recopilan para estos objetos cuando especifica una de las cláusulas COLLECT AGGREGATE ACTIVITY DATA, COLLECT AGGREGATE REQUEST DATA o COLLECT AGGREGATE UNIT OF WORK DATA al crear o alterar los objetos. Para las clases de trabajo, también se recopilan histogramas si aplica una acción de trabajo COLLECT AGGREGATE ACTIVITY DATA a la clase de trabajo. Los histogramas disponibles son los siguientes:

- Vida útil de actividad de coordinador no anidada, cuando especifica AGGREGATE ACTIVITY DATA BASE o AGGREGATE ACTIVITY DATA EXTENDED para una subclase de servicio, para una carga de trabajo o para una acción de trabajo aplicada a una clase de trabajo.
- Tiempo de ejecución de actividad de coordinador no anidada, cuando especifica AGGREGATE ACTIVITY DATA BASE o AGGREGATE ACTIVITY DATA EXTENDED para una subclase de servicio, para una carga de trabajo o para una acción de trabajo aplicada a una clase de trabajo.
- Tiempo en cola de actividad de coordinador no anidada, cuando especifica AGGREGATE ACTIVITY DATA BASE o AGGREGATE ACTIVITY DATA EXTENDED para una subclase de servicio, para una carga de trabajo o para una acción de trabajo aplicada a una clase de trabajo.
- Tiempo de ejecución de petición, cuando especifica AGGREGATE REQUEST DATA BASE para una subclase de servicio; este histograma no se aplica a las cargas de trabajo o las clases de trabajo.

- Histograma de tiempo entre llegadas de actividades no anidadas, cuando especifica AGGREGATE ACTIVITY DATA EXTENDED para una subclase de servicio, para una carga de trabajo o para una acción de trabajo aplicada a una clase de trabajo.
- Coste estimado de actividad de DML no anidada, cuando especifica AGGREGATE ACTIVITY DATA EXTENDED para una subclase de servicio, para una carga de trabajo o para una acción de trabajo aplicada a una clase de trabajo.
- Vida útil de unidad de trabajo, cuando especifica AGGREGATE UNIT OF WORK DATA BASE para una clase de servicio

Todos los histogramas relacionados con actividades contienen información acerca de las actividades que se han completado, cancelado o rechazado.

## Plantillas de histograma

Opcionalmente, puede especificar una plantilla de histograma que se utilice para determinar qué aspecto ha de tener un histograma en particular, incluido el valor de intervalo alto. Una plantilla de histograma es un objeto *sin unidad*, es decir, que no tiene asignada ninguna unidad de medición predefinida. Una unidad de medición, que depende del contexto en el que se utiliza la plantilla de histograma, se asigna al histograma cuando se crea o altera una clase de servicio, una carga de trabajo o una acción de trabajo. Los histogramas basados en el tiempo, como los que se especifican como ejemplo con la cláusula ACTIVITY LIFETIME HISTOGRAM, utilizan unidades de milisegundo, mientras que los histogramas basados en el coste, como los especificados como ejemplo con la cláusula ACTIVITY ESTIMATED COST HISTOGRAM, utilizan unidades de timeron.

Puede crear una plantilla de histograma mediante la sentencia CREATE HISTOGRAM TEMPLATE, especificando el valor de intervalo alto máximo. Todos los demás valores de intervalo se definirán automáticamente como valores que incrementarán potencialmente a medida que se aproximen al valor de intervalo alto. Por ejemplo, para crear un histograma con el valor de intervalo alto 3.000.000, emita una sentencia como la que se indica a continuación:

```
CREATE HISTOGRAM TEMPLATE TEMPLATE1 HIGH BIN VALUE 3000000
```

Esta sentencia crea una plantilla de histograma con los siguientes valores de intervalo:

Int.	Int.
bajo	alto
0	1
1	2
2	3
3	4
4	6
6	9
9	13
13	19
19	28
28	41
41	60
60	87
87	127
127	184
184	268
268	389
389	565
565	821

821	1192
1192	1732
1732	2514
2514	3651
3651	5300
5300	7696
7696	11173
11173	16222
16222	23553
23553	34196
34196	49649
49649	72084
72084	104657
104657	151948
151948	220609
220609	320297
320297	465030
465030	675163
675163	980250
980250	1423197
1423197	2066299
2066299	3000000
3000000	Infinito

Puede aplicar una plantilla utilizando la palabra clave HISTOGRAM TEMPLATE adecuada al crear o alterar subclases de servicio, cargas de trabajo o acciones de trabajo. Si no especifica una plantilla de histograma, se utilizará la plantilla por omisión, SYSDEFAULTHISTOGRAM. Si no habilita la recopilación AGGREGATE ACTIVITY DATA para un objeto, la plantilla de histograma se pasará por alto.

Por ejemplo, para utilizar la plantilla de histograma TEMPLATE1 para el histograma de vida útil de actividad existente de la subclase de servicio MYSUBCLASS de la superclase de servicio MYSUPERCLASS, emita la sentencia siguiente:

```
ALTER SERVICE CLASS MYSUBCLASS UNDER MYSUPERCLASS
ACTIVITY LIFETIME HISTOGRAM TEMPLATE TEMPLATE1
```

Después de haber confirmado la sentencia ALTER SERVICE CLASS, el histograma de vida útil de actividad que se ha recopilado para la subclase de servicio MYSUBCLASS tiene valores de intervalo alto que se han determinado mediante la plantilla de histograma TEMPLATE1 en lugar de mediante la plantilla de histograma SYSDEFAULTHISTOGRAM.

Si cambia una clase de servicio o una carga de trabajo para utilizar una plantilla de histograma distinta o cambia una plantilla de histograma, el cambio no entrará en vigor hasta que se produzca un restablecimiento de estadísticas.

Puede descartar una plantilla de histograma mediante la utilización de la sentencia DROP HISTOGRAM TEMPLATE.

Puede ver las plantillas de histograma consultando la vista SYSCAT.HISTOGRAMTEMPLATES y ver los valores de intervalo alto de la plantilla de histograma correspondiente consultando la vista SYSCAT.HISTOGRAMTEMPLATEBINS. El valor de intervalo bajo siempre es 0 para el primer intervalo; para los demás intervalos, el valor de intervalo bajo es el valor de intervalo alto del intervalo anterior.

## Ejemplo

El ejemplo siguiente crea una función de tabla para calcular el histograma CoordActLifetime, CoordActExecTime, CoordActQueueTime o CoordActEstCost para una superclase de servicio de forma conjunta sumando las subclases. La suma de subclases es útil cuando las actividades vuelven a correlacionarse con subclases de servicio distintas bajo la misma superclase de servicio durante la ejecución, y puede producirse en un escenario de disminución de la prioridad en el que se utilizan niveles de clase de servicio y umbrales especializados para controlar dinámicamente los recursos para las actividades. Este ejemplo no se aplica al histograma CoordActInterArrivalTime porque los promedios ponderados calculados no contabilizan por el hecho de que el histograma CoordActInterArrivalTime de una subclase mide el tiempo transcurrido entre la llegada de una consulta en esa subclase y la siguiente consulta, pero el histograma CoordActInterArrivalTime de una superclase mide el tiempo transcurrido entre la llegada de una consulta en cualquiera de sus subclases y la siguiente consulta.

```
CONNECT TO SAMPLE
```

```
DROP FUNCTION histsuper
```

```
CREATE FUNCTION histsuper(superclass varchar(128),
                        histogram_type varchar(24))
RETURNS TABLE (statistics_timestamp timestamp,
                bin_top integer,
                number_in_bin integer,
                graph varchar(60))

LANGUAGE SQL
READS SQL DATA
NO EXTERNAL ACTION
DETERMINISTIC
RETURN WITH HISTOGRAMS AS
  (SELECT HISTOGRAM_TYPE,
         substr(PARENTSERVICECLASSNAME,1,26) as SUPERCLASS,
         STATISTICS_TIMESTAMP,
         TOP as BIN_TOP,
         sum(NUMBER_IN_BIN) as NUMBER_IN_BIN
   FROM HISTOGRAMBIN_DB2STATISTICS H,
        SYSCAT.SERVICECLASSES S
   WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID
        AND PARENTSERVICECLASSNAME = histsuper.superclass
        AND HISTOGRAM_TYPE = histsuper.histogram_type
        AND HISTOGRAM_TYPE IN ('CoordActLifetime', 'CoordActExecTime',
'CoordActQueueTime', 'CoordActEstCost')
   GROUP BY HISTOGRAM_TYPE, PARENTSERVICECLASSNAME, STATISTICS_TIMESTAMP, TOP)
  SELECT STATISTICS_TIMESTAMP,
         BIN_TOP,
         NUMBER_IN_BIN,
         substr(repeat('#', cast(NUMBER_IN_BIN * 60 /
         (SELECT CASE WHEN MAX(NUMBER_IN_BIN) = 0 THEN 1
         ELSE MAX(NUMBER_IN_BIN) END FROM HISTOGRAMS) AS INTEGER)),1,60)
  AS GRAPH FROM HISTOGRAMS
```

```
CONNECT RESET
```

La salida tiene el aspecto siguiente:

STATISTICS_TIMESTAMP	BIN_TOP	NUMBER_IN_BIN	GRAPH
2008-11-06-14.47.08.833188	-1	0	
2008-11-06-14.47.08.833188	1	1	
2008-11-06-14.47.08.833188	2	1	
2008-11-06-14.47.08.833188	3	2	
2008-11-06-14.47.08.833188	5	4	
2008-11-06-14.47.08.833188	8	7	
2008-11-06-14.47.08.833188	12	15	
2008-11-06-14.47.08.833188	19	29	#
2008-11-06-14.47.08.833188	29	41	#
2008-11-06-14.47.08.833188	44	67	##

2008-11-06-14.47.08.833188	68	112	###
2008-11-06-14.47.08.833188	103	228	####
2008-11-06-14.47.08.833188	158	335	#####
2008-11-06-14.47.08.833188	241	723	#####
2008-11-06-14.47.08.833188	369	1289	#####
2008-11-06-14.47.08.833188	562	1890	#####
2008-11-06-14.47.08.833188	858	2484	#####
2008-11-06-14.47.08.833188	1309	1943	#####
2008-11-06-14.47.08.833188	1997	478	#####
2008-11-06-14.47.08.833188	3046	221	####
2008-11-06-14.47.08.833188	4647	29	#
2008-11-06-14.47.08.833188	7089	7	
2008-11-06-14.47.08.833188	10813	0	
2008-11-06-14.47.08.833188	16493	2	
2008-11-06-14.47.08.833188	25157	0	
2008-11-06-14.47.08.833188	38373	1	
2008-11-06-14.47.08.833188	58532	0	
2008-11-06-14.47.08.833188	89280	0	
2008-11-06-14.47.08.833188	136181	0	
2008-11-06-14.47.08.833188	207720	0	
2008-11-06-14.47.08.833188	316840	0	
2008-11-06-14.47.08.833188	483283	0	
2008-11-06-14.47.08.833188	737162	0	
2008-11-06-14.47.08.833188	1124409	0	
2008-11-06-14.47.08.833188	1715085	0	
2008-11-06-14.47.08.833188	2616055	0	
2008-11-06-14.47.08.833188	3990325	0	
2008-11-06-14.47.08.833188	6086529	0	
2008-11-06-14.47.08.833188	9283913	0	
2008-11-06-14.47.08.833188	14160950	0	
2008-11-06-14.47.08.833188	21600000	0	

41 registro(s) seleccionado(s).

## Creación de una plantilla de histograma

Utilice la sentencia CREATE HISTOGRAM TEMPLATE para crear una plantilla de histograma. Las subclases de servicio y las acciones de trabajo utilizan plantillas de histograma para definir los valores de intervalo para las estadísticas que se mantienen mediante histogramas.

### Antes de empezar

Para crear una plantilla de histograma, necesita la autorización WLMADM o DBADM.

Consulte los temas siguientes para obtener más información sobre los requisitos:

- “Sentencias DDL de gestión de carga de trabajo” en la página 18
- Apéndice A, “Normas de denominación generales”, en la página 545

### Acerca de esta tarea

Algunas subclases de servicio, actividades de clases de trabajo y estadísticas de peticiones de DB2 se recopilan mediante histogramas. Todos los histogramas tienen un número específico de intervalos, y cada uno de ellos representa un rango en el que se cuentan las actividades o las peticiones. El tipo de unidades que se utiliza para los intervalos depende del tipo de histograma que haya creado. La plantilla de histograma describe el valor superior del penúltimo intervalo en el histograma, que afecta a los valores de todos los intervalos del histograma. Para obtener más información acerca de los histogramas, consulte el apartado “Histogramas en la gestión de carga de trabajo” en la página 278.

### Procedimiento

Para crear una plantilla de histograma:

1. Emita la sentencia `CREATE HISTOGRAM TEMPLATE`, especificando el nombre de la plantilla de histograma que desea crear y un valor para la palabra clave `HIGH BIN VALUE` para definir el valor superior del penúltimo intervalo.
2. Confirme los cambios. Cuando confirma los cambios, el histograma se añade a la vista `SYSCAT.HISTOGRAMTEMPLATES` y los intervalos se añaden a la vista `SYSCAT.HISTOGRAMTEMPLATEBINS`.

### **Alteración de una plantilla de histograma**

Utilice la sentencia `ALTER HISTOGRAM TEMPLATE` para alterar una plantilla de histograma existente. Las subclases de servicio y las acciones de trabajo utilizan plantillas de histograma para definir los valores de intervalo para las estadísticas que se mantienen mediante histogramas.

#### **Antes de empezar**

Necesita la autorización `WLMADM` o `DBADM` para alterar una plantilla de histograma.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

#### **Acerca de esta tarea**

Algunas subclases de servicio, actividades de clases de trabajo y estadísticas de peticiones de DB2 se recopilan mediante histogramas. Todos los histogramas tienen un número específico de intervalos, y cada uno de ellos representa un rango en el que se cuentan las actividades o las peticiones. El tipo de unidades que se utiliza para los intervalos depende del tipo de histograma que haya creado. La plantilla de histograma describe el valor superior del penúltimo intervalo en el histograma, que afecta a los valores de todos los intervalos del histograma. Para obtener más información acerca de los histogramas, consulte “Histogramas en la gestión de carga de trabajo” en la página 278.

#### **Procedimiento**

Para alterar una plantilla de histograma:

1. Emita la sentencia `ALTER HISTOGRAM TEMPLATE`, especificando el nombre de la plantilla de histograma que desea alterar y un valor para el parámetro `HIGH BIN VALUE` para modificar el valor superior del penúltimo intervalo.
2. Confirme los cambios. Cuando confirma los cambios, el valor de intervalo alto del histograma se actualiza en la vista `SYSCAT.HISTOGRAMTEMPLATEBINS`. El cambio no surte efecto hasta la próxima vez que restablezca las estadísticas de gestión de carga de trabajo. Consulte “Restablecimiento de estadísticas en objetos de gestión de carga de trabajo de DB2” en la página 291 para obtener más información.
3. Opcional: Ejecute el procedimiento almacenado `WLM_COLLECT_STATS` para recopilar y restablecer las estadísticas de manera que la plantilla de histograma nueva pueda utilizarse inmediatamente.

### **Descarte de una plantilla de histograma**

Puede descartar una plantilla de histograma si ya no la necesita.

## Antes de empezar

Para descartar una plantilla de histograma, necesita la autorización WLMADM o DBADM.

Consulte “Sentencias DDL de gestión de carga de trabajo” en la página 18 para obtener más información sobre los requisitos.

No puede descartar la plantilla de histograma SYSDEFAULTHISTOGRAM.

No puede descartar una plantilla de histograma si hay una subclase de servicio, una acción de trabajo o una carga de trabajo que haga referencia a ella. Puede ver las subclases de servicio y las acciones de trabajo que hacen referencia a una plantilla de histograma consultando la vista SYSCAT.HISTOGRAMTEMPLATESUSE.

## Procedimiento

Para descartar una plantilla de histograma:

1. Utilice la sentencia DROP HISTOGRAM TEMPLATE.
2. Confirme los cambios. Cuando confirma los cambios, el histograma se elimina de la vista SYSCAT.HISTOGRAMTEMPLATES y sus intervalos se eliminan de la vista SYSCAT.HISTOGRAMTEMPLATEBINS.

## Ejemplo: Cálculo de promedios y desviación estándar de histogramas en una configuración de gestión de carga de trabajo de DB2

Un uso que tienen los histogramas es el de proporcionar la desviación estándar en la vida útil de las actividades. En el ejemplo de este tema se muestra cómo se utilizan intervalos para el cálculo de esta estadística.

El cálculo del promedio de vida útil de cada actividad es una información de gran utilidad. Sin embargo, el promedio por sí solo no describe con precisión la experiencia del usuario. Si la variabilidad de la vida útil de las actividades es grande, los usuarios a los que ofrece soporte pueden ver que las consultas se ejecutan rápidamente unas veces (lo que está bien) y lentamente otras veces (lo que puede que no sea aceptable). Cuando define un objetivo de vida útil de las actividades, no sólo es importante el promedio de vida útil de las actividades, sino también la desviación estándar de la vida útil de las actividades. Necesita comprender y controlar la variabilidad con el fin de garantizar que sus usuarios experimenten realmente el promedio observado.

En una configuración de gestión de carga de trabajo de DB2, se recopilan estadísticas de cada miembro. En el ejemplo siguiente se muestra cómo obtener el promedio de vida útil de las actividades para un solo miembro.

Supongamos que tiene un entorno de un solo miembro y un histograma con los siguientes intervalos. Hay más intervalos en los histogramas reales, pero aquí se limita a ocho intervalos para simplificar el ejemplo.

Intervalo 1 - de 0 a 2 segundos  
Intervalo 2 - de 2 a 4 segundos  
Intervalo 3 - de 4 a 8 segundos  
Intervalo 4 - de 8 a 16 segundos  
Intervalo 5 - de 16 a 32 segundos  
Intervalo 6 - de 32 a 64 segundos  
Intervalo 7 - de 64 a 128 segundos  
Intervalo 8 - de 128 segundos a infinito

Puede calcular una aproximación del promedio suponiendo que el tiempo medio de respuesta de una consulta correspondiente a un intervalo con el rango  $x$  a  $y$  sea  $(x + y)/2$ . A continuación, puede multiplicar este número por el número de consultas que corresponden al intervalo, sumar todos los intervalos y dividir la suma por el número total. Para el ejemplo anterior, supongamos que el tiempo medio de respuesta de cada intervalo es:

Promedio de vida útil del intervalo 1 =  $(0+2)/2 = 1$   
 Promedio de vida útil del intervalo 2 =  $(2+4)/2 = 3$   
 Promedio de vida útil del intervalo 3 =  $(4+8)/2 = 6$   
 Promedio de vida útil del intervalo 4 =  $(8+16)/2 = 12$   
 Promedio de vida útil del intervalo 5 =  $(16+32)/2 = 24$   
 Promedio de vida útil del intervalo 6 =  $(32+64)/2 = 48$   
 Promedio de vida útil del intervalo 7 =  $(64+128)/2 = 96$

Supongamos que se ha recopilado el siguiente histograma durante el periodo de medición:

| Recuento |
|----------|----------|----------|----------|----------|----------|----------|----------|
| int. 1   | int. 2   | int. 3   | int. 4   | int. 5   | int. 6   | int. 7   | int. 8   |
| 20       | 30       | 80       | 10       | 5        | 3        | 2        | 0        |

Para calcular el promedio de vida útil, el intervalo 8 debe estar vacío. El intervalo 8 sólo existe para que sepa cuándo necesita cambiar el límite superior de su rango. Por este motivo, debe especificar el límite superior del rango.

Puede aproximarse al promedio de vida útil del miembro 1 del modo siguiente:

$$\begin{aligned} \text{Promedio de vida útil} &= (20 \times 1 + 30 \times 3 + 80 \times 6 + 10 \times 12 + 5 \times 24 + 3 \times 48 + 2 \times 96) / 150 \\ &= (20 + 90 + 480 + 120 + 120 + 144 + 192) / 150 \\ &= 1166 / 150 \\ &= 7,77 \text{ segundos} \end{aligned}$$

Puede aproximarse a la desviación estándar de la vida útil del modo siguiente:

$$\text{Desviación estándar} = [(20 \times (1 - 7,77)^2 + 30 \times (3 - 7,77)^2 + \dots) / 150]^{1/2}$$

Para entornos de bases de datos de varios miembros, los promedios y las desviaciones estándar pueden calcularse calculando primero un histograma combinado de todas las particiones de la base de datos, que se realiza sumando los totales de cada intervalo en todas las particiones de la base de datos.

Por ejemplo, supongamos que la base de datos tiene dos miembros, que los tamaños de intervalo de histograma son los que se han descrito anteriormente y que el histograma contiene los datos siguientes:

Miembro	Recuento							
base datos	int. 1	int. 2	int. 3	int. 4	int. 5	int. 6	int. 7	int. 8
1	20	30	80	10	5	3	2	0
2	1	5	20	20	4	0	0	0

Puesto que los tamaños de los intervalos son los mismos en todos los miembros de base de datos, resulta fácil calcular el histograma global:

| Recuento |
|----------|----------|----------|----------|----------|----------|----------|----------|
| int. 1   | int. 2   | int. 3   | int. 4   | int. 5   | int. 6   | int. 7   | int. 8   |
| 21       | 35       | 100      | 30       | 9        | 3        | 2        | 0        |

A partir del histograma combinado, puede calcular el promedio de vida útil global y la desviación estándar de forma similar a como se calculan en un entorno de un solo miembro:

$$\begin{aligned} \text{Promedio de vida útil} &= (21 \times 1 + 35 \times 3 + 100 \times 6 + 30 \times 12 + 9 \times 24 + 3 \times 48 + 2 \times 96) / 200 \\ &= (21 + 105 + 600 + 360 + 216 + 144 + 192) / 200 \\ &= 1638 / 200 \end{aligned}$$

= 8,19 segundos

Desviación estándar =  $[(21 \times (1 - 8,19)^2 + 35 \times (3 - 7,77)^2 + \dots) / 200]^{1/2}$

## Herramienta de análisis histórico

La instalación del servidor de datos de DB2 incluye un par de scripts Perl de ejemplo que generan información sobre cuáles son las tablas, los índices y las columnas a los que se ha accedido o no mediante un análisis histórico.

Estos scripts proporcionan funciones de análisis histórico mediante la utilización de la información que captura el supervisor de sucesos de actividad de gestión de carga de trabajo. La herramienta de análisis histórico de gestión de carga de trabajo está escrita en Perl; puede utilizar estos scripts tal cual o modificarlos para generar informes de análisis histórico adicionales para que se adapte a sus necesidades.

La herramienta de análisis histórico de gestión de carga de trabajo consta de dos scripts, que pueden encontrarse en la vía de acceso `samples/perl` del directorio de instalación:

- `wlmhist.pl`: genera datos históricos.
- `wlmhistrep.pl`: genera informes a partir de los datos históricos.

También se incluye un archivo llamado `DB2WlmHist.pm`, que contiene las rutinas Perl comunes que utilizan los dos scripts.

Consulte el archivo `README_WLMHIST` que se encuentra en el mismo directorio de archivos para obtener más información sobre la configuración y la ejecución de los scripts.

## Recopilación de las estadísticas de gestión de carga de trabajo utilizando un supervisor de sucesos de estadísticas

Las estadísticas para los objetos de la gestión de carga de trabajo de DB2 pueden enviarse a un supervisor de sucesos de estadísticas para realizar un análisis histórico.

### Acerca de esta tarea

Puede utilizar estadísticas para comprender el comportamiento de su sistema en el transcurso del tiempo (por ejemplo, cuál es el promedio de vida útil de las actividades, cuánto tiempo pasan en cola las actividades, cuál es la distribución de las actividades grandes comparadas con las pequeñas, etc.) establecer umbrales (por ejemplo, buscar la limitación superior de las actividades simultáneas) y detectar problemas (por ejemplo, detectar si el promedio de vida útil que los usuarios observan es más alto de lo normal). Consulte el apartado “Estadísticas para objetos de la gestión de carga de trabajo de DB2” en la página 265 para obtener una descripción de las estadísticas que se recopilan para cada objeto de la gestión de carga de trabajo de DB2.

Puede enviar automáticamente las estadísticas de gestión de carga de trabajo a un supervisor de sucesos a intervalos fijos de tiempo, o puede hacerlo manualmente en un momento concreto.

### Procedimiento

Para recopilar automáticamente las estadísticas de gestión de carga de trabajo a intervalos fijos de tiempo:

1. Utilice la sentencia `CREATE EVENT MONITOR` para crear un supervisor de sucesos `STATISTICS`. Por ejemplo, puede emitir la sentencia siguiente:  

```
CREATE EVENT MONITOR STATS1 FOR STATISTICS WRITE TO TABLE
```
2. Utilice la sentencia `COMMIT` para confirmar los cambios.
3. Utilice la sentencia `SET EVENT MONITOR STATE` para activar el supervisor de sucesos. En lugar de utilizar la sentencia `SET EVENT MONITOR STATE`, puede utilizar el valor por omisión `AUTOSTART` para que el supervisor de sucesos `STATISTICS` se active la próxima vez que lo haga la base de datos. Si desea definir varios supervisores de sucesos `STATISTICS`, no debe utilizar la opción `AUTOSTART`.
4. Utilice la sentencia `COMMIT` para confirmar los cambios.
5. Opcional: Habilite la recopilación de estadísticas adicionales. Por omisión, solo se recopila un conjunto mínimo de estadísticas para cada objeto de la gestión de carga de trabajo de DB2. Consulte “Estadísticas para objetos de la gestión de carga de trabajo de DB2” en la página 265 para conocer los detalles sobre qué estadísticas se recopilan por omisión para cada objeto. Especifique la recopilación de datos de actividades agregadas de subclases de servicio, cargas de trabajo y clases de trabajo utilizando la palabra clave `COLLECT AGGREGATE ACTIVITY DATA` en las sentencias `ALTER SERVICE CLASS` y `ALTER WORK ACTION SET`. Especifique la recopilación de datos de petición agregados de subclases de servicio utilizando la palabra clave `COLLECT AGGREGATE REQUEST DATA` en la sentencia `ALTER SERVICE CLASS`. Confirme los cambios mediante `COMMIT`.
6. Especifique un intervalo de recopilación actualizando el parámetro de configuración `wlm_collect_int` de la base de datos. El parámetro `wlm_collect_int` especifica un intervalo de tiempo en minutos. En cada intervalo, se graba la copia de las estadísticas de gestión de carga de trabajo correspondientes a todos los objetos de gestión de carga de trabajo de DB2 en el supervisor activo de sucesos de estadísticas, y luego se restablecen las estadísticas. En un entorno de base de datos de varios miembros, el parámetro `wlm_collect_int` debe actualizarse en el miembro del catálogo. Este parámetro puede actualizarse dinámicamente. Por ejemplo:  

```
CONNECT TO alias_base_datos
UPDATE DATABASE CONFIGURATION USING WLM_COLLECT_INT 5 IMMEDIATE
```

## Resultados

Después de realizar los pasos anteriores, las estadísticas de gestión de carga de trabajo se graban en el supervisor de sucesos de estadísticas cada `wlm_collect_int` minutos. Cada registro que se graba en el supervisor de sucesos de estadísticas tiene un valor `STATISTICS_TIMESTAMP` y un valor `LAST_WLM_RESET`. El intervalo de tiempo de `LAST_WLM_RESET` a `STATISTICS_TIMESTAMP` define el intervalo de recopilación (es decir, el intervalo de tiempo durante el que se recopilaron las estadísticas en ese registro).

Las recopilaciones tienen lugar en los intervalos de tiempo especificados partiendo como referencia desde un Domingo a las 00:00:00. Cuando el miembro del catálogo se activa, la siguiente recopilación tiene lugar al principio del siguiente intervalo programado con respecto a esa hora fija. El intervalo programado no es relativo a la hora de activación del miembro del catálogo. Si un miembro no está activo en el momento de la recopilación, no se recopila ninguna estadística para ese miembro. Por ejemplo, si el valor del intervalo se ha establecido en 60 y el miembro del catálogo se ha activado el domingo a las 9:24 de la mañana, la ejecución de las recopilaciones quedará programada para cada hora a partir de la siguiente hora en

punto. Por lo tanto, la siguiente recopilación se producirá a las 10:00 de la mañana. Si el miembro no está activo a las 10:00 de la mañana, no se recopilará ninguna estadística para ese miembro.

Si el parámetro **wlm\_collect\_int** está establecido en un valor que no es cero y no hay ningún supervisor de sucesos de estadísticas activo, las estadísticas de gestión de carga de trabajo se restablecerán igualmente cada **wlm\_collect\_int** minutos, pero las estadísticas no se recopilarán. Los datos se perderán. Por esta razón, no se recomienda que especifique un valor de **wlm\_collect\_int** que no sea cero sin activar antes un supervisor de sucesos de estadísticas.

Si el parámetro **wlm\_collect\_int** se establece en 0 (el valor por omisión), las estadísticas no se enviarán automáticamente al supervisor de sucesos de estadísticas. Puede enviar las estadísticas manualmente al supervisor de sucesos de estadísticas para realizar posteriormente un análisis histórico utilizando el procedimiento **WLM\_COLLECT\_STATS**. Cuando se invoca este proceso, realiza las mismas acciones que las que se llevan a cabo con un intervalo de recopilación de estadísticas automático. Es decir, las estadísticas se envían al supervisor de sucesos de estadísticas, y las estadísticas se restablecen. Si no hay ningún supervisor de sucesos de estadísticas activo, se restablecen los valores pero no se recopilan los datos. Si solo desea restablecer las estadísticas, puede invocar el procedimiento **WLM\_COLLECT\_STATS** mientras no haya ningún supervisor de sucesos de estadísticas activo.

La recopilación manual de estadísticas no interfiere en la recopilación automática de estadísticas. Por ejemplo, supongamos que ha establecido **wlm\_collect\_int** en 60. Las estadísticas se envían al supervisor de sucesos de estadísticas cada hora. Ahora supongamos que la última vez que se recopilaron estadísticas fue a las 5:00 de la mañana. Puede invocar el procedimiento **WLM\_COLLECT\_STATS** a las 5:55 de la mañana, por el que se envían los valores de las estadísticas al supervisor de sucesos y se restablecen las estadísticas. La siguiente recopilación automática de estadísticas se realiza igualmente a las 6:00 de la mañana, una hora después de la última recopilación automática. El intervalo de la recopilación no se ve afectado por la recopilación manual ni por el restablecimiento de las estadísticas que se produce durante el intervalo.

**Notas:**

- Las funciones de la tabla de estadísticas de gestión de carga de trabajo de DB2 indican los valores actuales de las estadísticas. Si ha habilitado la recopilación automática de estadísticas de gestión de carga de trabajo, estos valores se restablecerán periódicamente a los intervalos definidos por el parámetro de configuración **wlm\_collect\_int** de la base de datos. Cuando observe las estadísticas indicadas por las funciones de la tabla, fíjese siempre en la columna **LAST\_RESET**. Esta columna indica la hora en que se restablecieron las estadísticas por última vez. Si el intervalo de tiempo entre la hora del último restablecimiento y la hora actual no es lo suficientemente amplio, es posible que no se hayan recopilado suficientes datos para establecer una conclusión significativa.
- Si utiliza una recopilación automática de estadísticas de gestión de carga de trabajo, deberá podar los archivos o las tablas del supervisor de sucesos de manera periódica. El supervisor de sucesos no poda automáticamente los datos que se van recopilando, y la recopilación automática llenará los archivos o las tablas con el paso del tiempo.
- Cuando se desactiva una base de datos, se restablecen las estadísticas. Al desactivar la base de datos no se envían estadísticas al supervisor de sucesos de estadísticas. Si no desea perder las estadísticas acumuladas

desde la última recopilación debido a una desactivación, invoque manualmente el procedimiento `WLM_COLLECT_STATS` antes de desactivar la base de datos.

- El procedimiento `WLM_COLLECT_STATS` restablece las estadísticas de una forma distinta a como lo hace el mandato **RESET MONITOR**. El mandato **RESET MONITOR** restablece los valores de los elementos de supervisor de instantáneas almacenando sus valores presentes. Una vez emitido el mandato **RESET MONITOR**, el proceso de instantáneas indica el delta entre estos valores y los valores actuales. En cambio, el restablecimiento que se efectúa mediante el procedimiento `WLM_COLLECT_STATS` no almacena ningún valor, pero restablece a cero todos los contadores de estadísticas para cada objeto de gestión de carga de trabajo de DB2 aplicable.

Asimismo, con el mandato **RESET MONITOR**, cada proceso (adjunto) tiene su propia vista privada de los datos del supervisor. El hecho de que un usuario lleve a cabo un restablecimiento no afecta a otros usuarios. En cambio, el restablecimiento de las estadísticas de la gestión de carga de trabajo afecta a todos los usuarios.

## Restablecimiento de estadísticas en objetos de gestión de carga de trabajo de DB2

En este procedimiento se describe cómo restablecer estadísticas para objetos de gestión de carga de trabajo de DB2.

Tenga en cuenta que el restablecimiento de estadísticas sólo se aplica a las estadísticas de gestión de carga de trabajo de DB2; se recopilarán las métricas de las que informan las interfaces de supervisión, pero no se restablecerán.

Cuatro sucesos restablecerán las estadísticas almacenadas para cada objeto de gestión de carga de trabajo de DB2. (Para obtener una descripción de las estadísticas que se mantienen para cada objeto, consulte “Estadísticas para objetos de la gestión de carga de trabajo de DB2” en la página 265.)

- Se invoca el procedimiento almacenado `WLM_COLLECT_STATS`. Consulte “Recopilación de las estadísticas de gestión de carga de trabajo utilizando un supervisor de sucesos de estadísticas” en la página 288 para obtener detalles.
- La recopilación de estadísticas automática de gestión de carga de trabajo de DB2 y el proceso de restablecimiento que el parámetro de configuración de base de datos `wlm_collect_int` controla dan lugar a una recopilación y restauración. Consulte “Recopilación de las estadísticas de gestión de carga de trabajo utilizando un supervisor de sucesos de estadísticas” en la página 288 para obtener detalles.
- La base de datos vuelve a activarse. Cada vez que la base de datos se activa en un miembro, se restablecen las estadísticas de todos los objetos de gestión de carga de trabajo de DB2 de ese miembro.
- El objeto para el que se mantienen las estadísticas se modifica y el cambio se confirma. Por ejemplo, si se altera una subclase de servicio, cuando se confirma la sentencia `ALTER`, se restablecen las estadísticas de esa subclase de servicio.

Puede determinar la última vez que se restablecieron las estadísticas de un objeto de gestión de carga de trabajo de DB2 específico utilizando las funciones de tabla de estadísticas y observando la indicación de fecha y hora de la columna `LAST_RESET`. Por ejemplo, para ver la última vez que se restablecieron las

estadísticas de la subclase de servicio SYSDEFAULTSUBCLASS bajo la superclase de servicio SYSDEFAULTUSERCLASS, puede emitir una consulta como la siguiente:

```
SELECT LAST_RESET
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS( 'SYSDEFAULTUSERCLASS',
'SYSDEFAULTSUBCLASS', -2)) AS T
```

Todas las funciones de tabla de estadísticas devuelven las estadísticas que se han acumulado desde la última vez que se restablecieron las estadísticas. Un restablecimiento de estadísticas se produce cuando una base de datos se activa o se desactiva, cuando altera un objeto de gestión de carga de trabajo de DB2 (sólo se restablecen las estadísticas de ese objeto) y cuando llama al procedimiento almacenado WLM\_COLLECT\_STATS. Las estadísticas también se restablecen automáticamente de acuerdo con el período de tiempo que define el parámetro de configuración de base de datos `wlm_collect_int`, si ha establecido este parámetro en un valor distinto de cero.

El período de tiempo que `wlm_collect_int` especifica no se ve afectado por un restablecimiento de estadísticas que se produzca durante el intervalo que el parámetro de configuración especifica. Por ejemplo, si ejecuta la función de tabla WLM\_COLLECT\_STATS cinco minutos después de haberse iniciado el intervalo de 20 minutos que `wlm_collect_int` especifica, el intervalo seguirá caducando 15 minutos después. La recopilación de estadísticas y el restablecimiento que se producen no dependen de la aparición de la siguiente recopilación de estadísticas y restablecimiento a los cinco minutos.

Si cambia una clase de servicio o una carga de trabajo para utilizar una plantilla de histograma distinta o cambia una plantilla de histograma, el cambio no entrará en vigor hasta que se produzca un restablecimiento de estadísticas.

Si invoca la función de tabla WLM\_COLLECT\_STATS para recopilar y restablecer estadísticas al mismo tiempo que otra recopilación y restablecimiento están en proceso (por ejemplo, si la invocación de la función de tabla se solapa con el intervalo de recopilación y restablecimiento periódicos que `wlm_collect_int` produce o si otro usuario invoca WLM\_COLLECT\_STATS al mismo tiempo), la petición de recopilación y restablecimiento de WLM\_COLLECT\_STATS se pasará por alto y se devolverá el aviso SQL1632W.

---

## Métricas de supervisión para la gestión de carga de trabajo de DB2

Las métricas de supervisión proporcionan datos sobre el estado y el rendimiento de las consultas del servidor de datos de DB2, que pueden utilizarse como entrada para una herramienta de otro fabricante o en conjunción con los scripts adicionales que proporcione para analizar las métricas devueltas.

Se mantienen métricas para un número de objetos de base de datos de DB2. Estas métricas residen en memoria y pueden verse en tiempo real mediante las funciones de tabla de métricas de supervisión de DB2; también pueden recopilarse y enviarse a un supervisor de sucesos, donde pueden verse para realizar un análisis histórico.

### Métricas de supervisión para actividades

Puede obtener métricas de supervisión para actividades mediante el empleo de lo siguiente:

- El supervisor de sucesos de actividades (la tabla ACTIVITYMETRICS o la columna DETAILS\_XML de la tabla ACTIVITY)

- Función de tabla MON\_GET\_ACTIVITY\_DETAILS

Las métricas de supervisión para actividades están controladas por el parámetro de configuración de base de datos **mon\_act\_metrics** y la cláusula COLLECT ACTIVITY METRICS en una carga de trabajo. Se recopilarán métricas para una actividad si el parámetro de configuración de la base de datos está establecido con un valor que no sea NONE o si la actividad se emite mediante una conexión que esté asociada con una carga de trabajo cuya cláusula COLLECT ACTIVITY METRICS tenga un valor distinto de NONE.

Puede utilizar controles de nivel de carga de trabajo para conseguir una mejor granularidad de supervisión si no desea recopilar métricas para todas las actividades. Si se habilita la recopilación de métricas de la actividad en el nivel de base de datos (por omisión está habilitada), se recopilan las métricas de todas las actividades, independientemente del valor establecido a nivel de carga de trabajo.

Consulte la documentación de supervisión para obtener más detalles.

### **Métricas de supervisión de nivel de sistema**

Puede obtener métricas de supervisión de nivel del sistema agregadas por clases de servicio y cargas de trabajo, mediante:

- El supervisor de sucesos de estadísticas (columna DETAILS\_XML en los grupos lógicos wlstats y scstats)
- Las funciones de tabla MON\_GET\_SERVICE\_SUBCLASS, MON\_GET\_SERVICE\_SUBCLASS\_DETAILS, MON\_GET\_WORKLOAD y MON\_GET\_WORKLOAD\_DETAILS

Las métricas de supervisión para peticiones al servidor de datos, incluidas las peticiones que forman parte de una actividad, se controlan mediante el parámetro de configuración de la base de datos **mon\_req\_metrics** y la cláusula COLLECT REQUEST METRICS en una superclase de servicio. Se recopilarán métricas para una petición si el parámetro de configuración de la base de datos está establecido con un valor que no sea NONE o si la petición se emite mediante una conexión que esté correlacionada con una subclase bajo una superclase cuya cláusula COLLECT REQUEST METRICS tenga un valor distinto de NONE.

Puede utilizar controles de nivel de superclase de servicio para conseguir una mejor granularidad de supervisión si no desea recopilar métricas para todas las peticiones. Si se habilita la recopilación de métricas de la petición en el nivel de base de datos (por omisión está habilitada), se recopilan las métricas de todas las peticiones, independientemente del valor establecido a nivel de superclase de servicio.

Consulte la documentación de supervisión para obtener más detalles.

---

## **Funciones de tabla de gestión de carga de trabajo e integración del supervisor de instantáneas**

Puede utilizar las funciones de tabla de la gestión de carga de trabajo de DB2 con las funciones de tabla del supervisor de instantáneas al realizar la determinación de problemas o el ajuste del rendimiento.

Las funciones de tabla de la gestión de carga de trabajo de DB2 y las funciones de tabla del supervisor de instantáneas comparten los campos siguientes. Puede

realizar uniones en estos campos para obtener los datos que necesita para llevar a cabo las actividades de diagnóstico y de ajuste del rendimiento. Observe que, a diferencia de las funciones de tabla de instantáneas, las funciones de tabla de WLM no obtienen su información del supervisor de instantáneas, por lo que la información disponible en las funciones de tabla de WLM no está disponible desde el supervisor de instantáneas.

Tabla 61. Campos que se comparten entre las funciones de tabla de la gestión de carga de trabajo de DB2 y del supervisor de instantáneas

Campo de función de tabla de la gestión de carga de trabajo	Campo de función de tabla del supervisor de instantáneas
agent_tid	agent_pid
application_handle	agent_id agent_id_holding_lock
session_auth_id	session_auth_id
member	node_number
utility_id	utility_id
workload_id	workload_id

Como ejemplo de la razón por la que ha de utilizarse una unión entre funciones de tabla diferentes, supongamos que desea obtener información básica acerca de todos los programas de utilidad que se ejecutan en la superclase de servicio BATCH. Podría emitir la consulta siguiente:

```
SELECT SUBSTR(UTILITY_TYPE,1,4) TYPE,
       UTILITY_PRIORITY PRIORITY,
       SUBSTR(UTILITY_DESCRIPTION,1,12) AS UTILITY_DESCRIPTION,
       SUBSTR(UTILITY_DBNAME,1,8) AS DBNAME,
       UTILITY_STATE,
       SUBSTR(UTILITY_INVOKER_TYPE,1,7) INVOKER,
       SUBSTR(CHAR(WLM.MEMBER),1,4) MEMB,
       SUBSTR(CLASSES.PARENTSERVICECLASSNAME,1,19) SUPERCLASS_NAME,
       SUBSTR(CLASSES.SERVICECLASSNAME,1,18) SUBCLASS_NAME
FROM SYSIBMADM.SNAPUTIL SNAP,
     TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(CAST(NULL AS BIGINT), -2)) WLM,
     SYSCAT.SERVICECLASSES CLASSES
WHERE SNAP.UTILITY_ID = WLM.UTILITY_ID
      AND WLM.SERVICE_CLASS_ID = CLASSES.SERVICECLASSID
      AND CLASSES.SERVICECLASSNAME = 'SYSDEFAULTSUBCLASS'
      AND CLASSES.PARENTSERVICECLASSNAME = 'BATCH'
ORDER BY WLM.MEMBER
```

La salida podría ser similar a la siguiente:

TYPE	PRIORITY	UTILITY_DESCRIPTION	DBNAME	UTILITY_STATE	INVOKER	MEMB	SUPERCLASS_NAME	SUBCLASS_NAME
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	1	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	1	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	1	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	2	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	2	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	2	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	3	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	3	BATCH	SYSDEFAULTSUBCLASS
LOAD	-	OFFLINE LOAD	SAMPLE	EXECUTE	USER	3	BATCH	SYSDEFAULTSUBCLASS

## Supervisión de violaciones de umbral

Cuando se infringe un umbral del gestor de carga de trabajo de DB2, se graba un registro de violación de umbral en el supervisor de sucesos THRESHOLD VIOLATIONS, si existe.

## Acerca de esta tarea

El registro de violación de umbral incluye la información siguiente:

- Una descripción del umbral que se infringió (identificador, valor máximo, etc.).
- Una identificación de la actividad que cometió la violación de umbral, incluido el identificador de la aplicación que envió la actividad, el identificador de actividad exclusivo y el identificador de la unidad de trabajo.
- La hora en que se cometió la violación de umbral.
- La acción que se llevó a cabo. La acción indica si la actividad que cometió la violación de umbral tenía permiso para continuar o se detuvo. Si la actividad se ha detenido, la aplicación que ha enviado la actividad habrá recibido un error SQL4712N.

Cuando se produce una violación de un umbral que tiene definida la acción REMAP ACTIVITY, el registro de violación de umbral es opcional. Las cláusulas NO EVENT MONITOR RECORD o LOG EVENT MONITOR RECORD de la sentencia CREATE THRESHOLD son las que determinan si se anota un registro de violación de umbral.

Si lo desea puede hacer que la información de actividad detallada (incluido el texto de la sentencia) se grabe en un supervisor activo de sucesos de actividades en caso de que la violación de umbral haya sido originada por una actividad. La información de actividad se graba cuando ésta finaliza, no cuando se comete la violación de umbral. Si desea especificar que la información de actividad se recopile cuando se produzca la violación de umbral, utilice la palabra clave COLLECT ACTIVITY DATA en las sentencias CREATE o ALTER del umbral o del conjunto de acciones de trabajo.

## Procedimiento

Para supervisar las violaciones de umbral:

1. Utilice la sentencia CREATE EVENT MONITOR para crear un supervisor de sucesos del tipo THRESHOLD VIOLATIONS. Por ejemplo:  
`CREATE EVENT MONITOR VIOLATIONS FOR THRESHOLD VIOLATIONS WRITE TO TABLE`
2. Utilice la sentencia COMMIT para confirmar los cambios.
3. Utilice la sentencia SET EVENT MONITOR STATE para activar el supervisor de sucesos. En lugar de utilizar la sentencia SET EVENT MONITOR STATE, puede utilizar el valor por omisión AUTOSTART para que el supervisor de sucesos THRESHOLD VIOLATIONS se active la próxima vez que lo haga la base de datos. Si desea definir varios supervisores de sucesos THRESHOLD VIOLATIONS, no debe utilizar la opción AUTOSTART.
4. Utilice la sentencia COMMIT para confirmar los cambios.

**Nota:** Si crea umbrales, también debe crear y activar un supervisor de sucesos de violaciones de umbral para poder supervisar las violaciones de umbral que puedan producirse. El supervisor de sucesos de violaciones de umbral no tienen ningún impacto a menos que se produzca una violación de umbral.

## Ejemplo

En este ejemplo se muestra cómo determinar las correlaciones de una actividad determinada que se producen como resultado de una violación de umbral que incluía una acción REMAP ACTIVITY. Para buscar las actividades correlacionadas, utilice una sentencia como ésta:

```

SELECT VARCHAR(APPL_ID, 30) AS APPLID,
       UOW_ID,
       ACTIVITY_ID,
       VARCHAR(T.PARENTSERVICECLASSNAME,20) AS SERVICE_SUPERCLASS,
       VARCHAR(T.SERVICECLASSNAME,20) AS FROM_SERVICE_SUBCLASS,
       VARCHAR(S.SERVICECLASSNAME,20) AS TO_SERVICE_SUBCLASS
FROM THRESHOLDVIOLATIONS_TH1,
     SYSCAT.SERVICECLASSES AS T,
     SYSCAT.SERVICECLASSES AS S
WHERE SOURCE_SERVICE_CLASS_ID = T.SERVICECLASSID AND
       DESTINATION_SERVICE_CLASS_ID = S.SERVICECLASSID AND
       THRESHOLD_ACTION = 'REMAP'
ORDER BY APPLID, ACTIVITY_ID, UOW_ID, TIME_OF_VIOLATION ASC;

```

En este ejemplo, se han producido dos correlaciones para la actividad enviada por la aplicación con el ID \*N0.swalkty.080613140844, que queda identificada por el ID de actividad 1 y la unidad de trabajo (UOW) ID 1:

APPLID	UOW_ID	ACTIVITY_ID	SERVICE_SUPERCLASS	FROM_SERVICE_SUBCLASS	TO_SERVICE_SUBCLASS
*N0.swalkty.080613140844	1	1	1 WORK	HIGH	MED
*N0.swalkty.080613140844	1	1	1 WORK	MED	LOW

2 registro(s) seleccionado(s).

La salida se ordena según la hora en que se produce la violación de umbral, y muestra que la actividad se correlacionó dos veces después de empezar a ejecutarse. Aunque no aparece en la salida, la subclase de servicio inicial con la que se correlacionó la actividad es probablemente una subclase de servicio de prioridad alta, típico de una configuración de tres niveles que permite que las peticiones de ejecución corta finalicen más rápidamente. Puesto que la actividad no finalizó lo suficientemente rápido en la subclase de servicio de alta prioridad, infringió un umbral y se correlacionó con una subclase de servicio de prioridad media, y luego volvió a correlacionarse con una subclase de servicio de prioridad baja después de una segunda violación de umbral que se cometió más tarde.

## Cómo generar notificaciones por correo electrónico para violaciones de umbral

Con el método que aquí se describe, puede generar notificaciones por correo electrónico cuando se viola un umbral del gestor de carga de trabajo (WLM) de DB2.

### Antes de empezar

Con el fin de implementar este enfoque de notificación por correo electrónico, debe tener instalado DB2 Versión 9.7 o posterior. El soporte de SMTP utilizado aquí se proporciona a partir de DB2 V9.7.

### Acerca de esta tarea

Al finalizar esta tarea, las notificaciones por correo electrónico se envían si se producen violaciones de umbral de WLM durante los diez minutos posteriores a la última ejecución del procedimiento de notificación de umbrales. En este ejemplo, se utiliza el planificador de tareas administrativas de DB2 para planificar la ejecución del procedimiento de notificación de umbrales cada diez minutos.

### Procedimiento

1. Actualice el parámetro de base de datos `smtp_server` emitiendo el mandato siguiente:

```
UPDATE DB CONFIG USING SMTP_SERVER nombre_servidor_smtp
```

2. Cree un supervisor de sucesos de grabación en tabla para violaciones de umbral y violaciones de grabación en la tabla TEST.THRESHOLDVIOLATIONS\_T emitiendo la sentencia siguiente:

```
CREATE EVENT MONITOR T FOR THRESHOLD VIOLATIONS WRITE TO TABLE
THRESHOLDVIOLATIONS( TABLE TEST.THRESHOLDVIOLATIONS_T )
```

3. Active el supervisor de sucesos de grabación en tabla T para violaciones de umbral emitiendo la sentencia siguiente:

```
SET EVENT MONITOR T STATE 1
```

4. Cree una tabla de control para realizar el seguimiento del último umbral con el que se ha generado la alerta emitiendo la sentencia siguiente:

```
CREATE TABLE TEST.THRESHOLD_NOTIFY_CONTROL( LAST_NOTIFICATION TIMESTAMP )
```

5. Cree un procedimiento almacenado de notificación de umbrales para generar mensajes de violación de umbral. El procedimiento del ejemplo siguiente repasa la tabla de violaciones de umbral y crea un informe donde se enumeran todas las violaciones de umbral que se han producido desde la última vez que se invocó el procedimiento. El informe se envía por correo electrónico utilizando los procedimientos SMTP de DB2.

```
CREATE PROCEDURE TEST.NOTIFY_ON_THRESHOLD_VIOLATION()
LANGUAGE SQL
BEGIN
    DECLARE NEWEST_VIOLATION    TIMESTAMP;
    DECLARE LAST_VIOLATION_SEEN  TIMESTAMP;
    DECLARE NOT_FOUND          INTEGER DEFAULT 0;
    DECLARE SENDER              VARCHAR(128);
    DECLARE RECIPIENTS          VARCHAR(128);
    DECLARE MESSAGE              VARCHAR(8192);
    DECLARE SUBJECT              VARCHAR(128);
    DECLARE THRESHOLDID         BIGINT;
    DECLARE APPL_ID              VARCHAR(64);
    DECLARE THRESHOLD_PREDICATE  VARCHAR(64);
    DECLARE TIME_OF_VIOLATION    TIMESTAMP;

    DECLARE C1 CURSOR FOR SELECT MAX(TIME_OF_VIOLATION)
                                FROM TEST.THRESHOLDVIOLATIONS_T;
    DECLARE C2 CURSOR FOR SELECT LAST_NOTIFICATION
                                FROM TEST.THRESHOLD_NOTIFY_CONTROL;
    DECLARE C3 CURSOR FOR SELECT THRESHOLD_PREDICATE, THRESHOLDID,
                                TIME_OF_VIOLATION, APPL_ID
                                FROM TEST.THRESHOLDVIOLATIONS_T
                                WHERE LAST_VIOLATION_SEEN IS NULL OR TIME_OF_VIOLATION
                                    > LAST_VIOLATION_SEEN;

    DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET NOT_FOUND = 1;

    OPEN C1;
    FETCH C1 INTO NEWEST_VIOLATION;
    CLOSE C1;

    IF ( NOT_FOUND = 0 ) THEN

        OPEN C2;
        FETCH C2 INTO LAST_VIOLATION_SEEN;
        CLOSE C2;

        IF ( NOT_FOUND = 1 ) THEN
            SET LAST_VIOLATION_SEEN = NULL;
        END IF;

        IF ( NOT_FOUND = 1 OR NEWEST_VIOLATION > LAST_VIOLATION_SEEN ) THEN

            DELETE FROM TEST.THRESHOLD_NOTIFY_CONTROL;
```

```

INSERT INTO TEST.THRESHOLD_NOTIFY_CONTROL VALUES
        ( NEWEST_VIOLATION );

SET SENDER = '<dirección de correo electrónico del remitente>';
SET RECIPIENTS = '<dirección de correo electrónico del destinatario>';
SET SUBJECT = 'Nuevas violaciones de umbral de WLM' ;

SET NOT_FOUND = 0;
SET MESSAGE = '';

OPEN C3;

FETCH C3 INTO THRESHOLD_PREDICATE, THRESHOLDID, TIME_OF_VIOLATION,
        APPL_ID;

WHILE ( NOT_FOUND = 0 ) DO

        SET MESSAGE = MESSAGE || 'Violation Timestamp = ' ||
                TIME_OF_VIOLATION || CHAR(X'0A');
        SET MESSAGE = MESSAGE || 'Threshold Predicate = ' ||
                THRESHOLD_PREDICATE || CHAR(X'0A');
        SET MESSAGE = MESSAGE || 'Threshold Id = ' ||
                THRESHOLDID || CHAR(X'0A');
        SET MESSAGE = MESSAGE || 'Appl Id = ' ||
                APPL_ID || CHAR(X'0A') || CHAR(X'0A');

        FETCH C3 INTO THRESHOLD_PREDICATE, THRESHOLDID,
                TIME_OF_VIOLATION, APPL_ID;

END WHILE;

CLOSE C3;

CALL UTL_MAIL.SEND( SENDER, RECIPIENTS, NULL, NULL, SUBJECT,
        MESSAGE );

COMMIT;

END IF;

END IF;

END@

```

6. Habilite el planificador de tareas administrativas de DB2 ejecutando el mandato siguiente:

```
db2set DB2_ATS_ENABLE=YES
```

7. Planifique el procedimiento de notificación de umbrales para que se ejecute cada diez minutos. Para planificar el procedimiento, debe tener privilegios de ejecución sobre el procedimiento. En el ejemplo siguiente se muestra cómo llevar esto al cabo:

```

CALL SYSPROC.ADMIN_TASK_ADD(
        'CHECK THRESHOLD VIOLATIONS EVERY 10 MINUTES',
        NULL,
        NULL,
        NULL,
        '0-59/10 * * * * ',
        'TEST',
        'NOTIFY_ON_THRESHOLD_VIOLATION',
        NULL,
        NULL,
        NULL )@

```

## Resultados

Siempre que se produzca una violación de umbral de WLM se enviará un correo electrónico (con una latencia de diez minutos como máximo). En el correo electrónico se describen todas las violaciones de umbral de WLM desde la última ejecución del procedimiento de violaciones de umbral, cuya ejecución se ha planificado cada diez minutos.

## Ejemplo

La salida siguiente es un ejemplo del contenido de un mensaje de correo electrónico de notificación, en el que se muestran las nuevas violaciones de umbral acumuladas desde la última ejecución del procedimiento de violaciones de umbral:

Asunto: Nuevas violaciones de umbral de WLM

```
Violation Timestamp = 2010-01-11-10.57.21.000000
Threshold Predicate = CPUTime
Threshold Id       = 1
Appl Id           = *LOCAL.horton.100111154912
```

```
Violation Timestamp = 2010-01-11-10.57.28.000000
Threshold Predicate = CPUTime
Threshold Id       = 1
Appl Id           = *LOCAL.horton.100111154912
```

```
Violation Timestamp = 2010-01-11-10.57.35.000000
Threshold Predicate = CPUTime
Threshold Id       = 1
Appl Id           = *LOCAL.horton.100111154912
```

---

## Recopilación de datos para actividades individuales

Puede utilizar el supervisor de sucesos ACTIVITIES para recopilar datos para actividades individuales que se ejecutan en su sistema. Los datos recopilados incluyen elementos como el texto de la sentencia y el entorno de compilación, y pueden utilizarse para investigar y diagnosticar problemas, y como entrada para otras herramientas (por ejemplo, el Asesor de diseño).

### Acerca de esta tarea

Puede recopilar información sobre actividades individuales para subclases de servicio, cargas de trabajo, clases de trabajo (a través de acciones de trabajo) e violaciones de umbral. La recopilación de actividades se habilita mediante la palabra clave COLLECT ACTIVITY DATA de las sentencias CREATE y ALTER de estos objetos de la gestión de carga de trabajo de DB2. Cuando finaliza una actividad, la información correspondiente se envía al supervisor activo de sucesos ACTIVITIES si:

- La actividad la envió una aplicación correlacionada con una carga de trabajo para la que se especificó COLLECT ACTIVITY DATA, o
- La actividad se ejecuta en una subclase de servicio para la que se especificó COLLECT ACTIVITY DATA, o
- La actividad tiene una acción de trabajo COLLECT ACTIVITY DATA aplicada, o
- La actividad infringe un umbral que se definió con la acción COLLECT ACTIVITY DATA

También puede utilizar el procedimiento WLM\_SET\_CONN\_ENV para activar la recopilación de actividades para su propia conexión de la aplicación antes de

ejecutar la petición del usuario, y luego ejecutar la petición del usuario, y utilizar WLM\_SET\_CONN\_ENV para desactivar la recopilación de actividades para su propia conexión de la aplicación. Suponiendo que ha creado y activado un supervisor de sucesos de actividades, la aplicación se parecería a esta:

```
call WLM_SET_CONN_ENV(cast (NULL as bigint),  
  '<collectactdata>WITHOUT DETAILS</collectactdata>')
```

... ejecutar petición del usuario ...

```
call WLM_SET_CONN_ENV(cast(NULL as bigint), '<collectactdata>NONE</collectactdata>')
```

La palabra clave COLLECT ACTIVITY DATA también controla la cantidad de información que se envía al supervisor de sucesos ACTIVITIES. Si la palabra clave específica WITH DETAILS, se recopilará la información de la sentencia (como el texto de la misma). Si la palabra clave específica WITH DETAILS AND VALUES, también se recopilarán los valores de datos.

Una actividad puede tener varias palabras clave COLLECT ACTIVITY DATA aplicadas. Por ejemplo, la actividad puede ejecutarse en una subclase de servicio para la que se ha especificado COLLECT ACTIVITY DATA, y mientras se ejecuta es posible que infrinja un umbral que tenga la acción COLLECT ACTIVITY DATA. En esta situación, la actividad solo se recopila una vez. La palabra clave COLLECT que especifica la cantidad más grande información que se recopila se aplica a la actividad. Por ejemplo, si se aplican COLLECT ACTIVITY DATA WITHOUT DETAILS y COLLECT ACTIVITY DATA WITH DETAILS a una actividad, ésta se recopila con la información detallada.

Si se utilizan las palabras clave ON ALL DATABASE MEMBERS con la cláusula COLLECT ACTIVITY DATA, se capturará un registro de actividad en cada miembro en el que la actividad se ejecuta en un entorno de base de datos de varios miembros. Los registros del supervisor de sucesos de actividad se graban cuando el último agente que trabaja en la actividad en ese miembro ha completado su ejecución. En función de la secuencia de sucesos de una sección, es posible que los agentes inicien y detengan su trabajo en una actividad en un miembro varias veces, dando lugar a que se capturen varios registros de actividad en ese miembro para la misma consulta. El trabajo total que la actividad realiza en ese miembro es la agregación de las métricas para cada registro que se captura para la actividad en el miembro.

## Procedimiento

Para habilitar la recopilación de actividades de un objeto de gestión de carga de trabajo de DB2 determinado:

1. Utilice la sentencia CREATE EVENT MONITOR para crear un supervisor de sucesos ACTIVITIES.
2. Utilice la sentencia COMMIT para confirmar los cambios.
3. Utilice la sentencia SET EVENT MONITOR STATE para activar el supervisor de sucesos. En lugar de utilizar la sentencia SET EVENT MONITOR STATE, puede utilizar el valor por omisión AUTOSTART para que el supervisor de sucesos ACTIVITIES se active la próxima vez que lo haga la base de datos. Si desea definir varios supervisores de sucesos ACTIVITIES, no debe utilizar la opción AUTOSTART.
4. Utilice la sentencia COMMIT para confirmar los cambios.

5. Identifique los objetos para los que desea recopilar actividades utilizando la sentencia ALTER SERVICE CLASS, ALTER WORK ACTION SET, ALTER THRESHOLD o ALTER WORKLOAD y especifique las palabras clave COLLECT ACTIVITY DATA.
6. Utilice la sentencia COMMIT para confirmar los cambios.

## Resultados

**Nota:** La recopilación de actividades individuales es más costosa que la recopilación de estadísticas de gestión de carga de trabajo. Debe intentar configurar la recopilación de actividades para recopilar el menor número de actividades posible. Por ejemplo, si necesita investigar las actividades enviadas por una aplicación específica, puede aislar dicha aplicación creando una carga de trabajo o una clase de servicio específicamente para esa aplicación, y habilitar solamente la recopilación de actividades para esa clase de servicio o carga de trabajo.

Es posible que no siempre sepa con antelación si desea capturar una actividad. Por ejemplo, suponga que tiene una petición que tarda mucho en ejecutarse y desea recopilar información sobre ella para su análisis posterior. En esta situación, es demasiado tarde para especificar la palabra clave COLLECT ACTIVITY DATA en los objetos de gestión de carga de trabajo de DB2 porque la actividad ya ha entrado en el sistema. En esta situación, puede utilizar el procedimiento almacenado WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS. El procedimiento almacenado WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS envía información sobre una actividad en ejecución al supervisor activo de sucesos ACTIVITIES. Para identificar la actividad que se va a recopilar utilice el descriptor de contexto de aplicación, el identificador de unidad de trabajo y el identificador de actividad. La información sobre la actividad se envía inmediatamente al supervisor de sucesos ACTIVITIES cuando se invoca al procedimiento: no es necesario que espere a que termine la actividad.

---

## Importación de información de actividad en el Asesor de diseño

Puede importar las actividades recopiladas por un supervisor de sucesos de actividades en el Asesor de diseño para ayudarle a tomar decisiones sobre los objetos de la base de datos a los que estas actividades acceden.

### Acerca de esta tarea

Las actividades que se importan en el asesor de diseño se han recopilado mediante las opciones COLLECT ACTIVITY DATA WITH DETAILS o COLLECT ACTIVITY DATA WITH DETAILS AND VALUES. La opción COLLECT ACTIVITY DATA WITHOUT DETAILS no es suficiente, no captará el texto de la sentencia que requiere el Asesor de diseño.

Para importar información de actividad desde las tablas del supervisor de sucesos de actividad en el Asesor de diseño, ejecute el mandato **db2adv** con el parámetro **-wlm**, seguido por los parámetros adicionales:

1. El nombre del supervisor de sucesos de actividades
2. Opcional: el nombre de la carga de trabajo o de la clase de servicio
3. Opcional: la hora de inicio y de finalización

Por ejemplo, para importar información sobre todas las actividades recopiladas por el supervisor de sucesos DB2ACTIVITIES en la base de datos SAMPLE, utilice este mandato:

```
db2adviz -d SAMPLE -wlm DB2ACTIVITIES
```

**Nota:** Solo puede importar información desde las tablas del supervisor de sucesos de actividades a través de la interfaz de línea de mandatos del Asesor de diseño.

---

## Cancelación de actividades

Si una actividad consume demasiados recursos, o tarda mucho en ejecutarse, puede cancelarla. Cancelar una actividad es menos radical que forzar la aplicación que envió la actividad. Una actividad cancelada devuelve el error SQL4725N al usuario, pero no finaliza la conexión ni afecta a otras actividades del usuario. Forzar la aplicación finaliza la conexión y las actividades del usuario.

### Acerca de esta tarea

Solo puede cancelar explícitamente una actividad si una actividad coordinadora está actualmente trabajando en una petición para la actividad. Si cancela una actividad en estado IDLE (es decir, no se está procesando ninguna petición), la actividad adquiere el estado de CANCEL\_PENDING y se cancela cuando se recibe la petición siguiente. Por ejemplo, si intenta cancelar una actividad CURSOR entre búsquedas, no se devuelve el error SQL4725N al usuario hasta la siguiente búsqueda después de la cancelación.

Todas las actividades del usuario son susceptibles de cancelación, incluso el programa de utilidad de carga y los procedimientos almacenados.

### Procedimiento

1. Identifique la actividad que desea cancelar. Puede utilizar la función de tabla WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para identificar las actividades que se ejecutan en una aplicación. También puede utilizar la función de tabla MON\_GET\_ACTIVITY\_DETAILS\_COMPLETE para ver detalles adicionales sobre una actividad en particular si la información de WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES no es suficiente para identificar el trabajo que las actividades están realizando.
2. Cancele la actividad utilizando el procedimiento almacenado WLM\_CANCEL\_ACTIVITY. El procedimiento almacenado toma los argumentos siguientes: *application\_handle*, *uow\_id* y *activity\_id*. Para conocer un ejemplo de cómo utilizar este procedimiento almacenado, consulte “Caso de ejemplo: Identificación de actividades que tardan demasiado tiempo en completarse” en la página 380.

---

## Directrices para capturar información sobre una actividad malintencionada e investigarla

En este tema se proporcionan directrices para capturar información sobre una actividad malintencionada e investigarla.

Primero, establezca un conjunto de criterios para lo que consideraría una actividad malintencionada. Por ejemplo:

- Una actividad que se ejecuta en una clase de servicio para actividades con un bajo coste estimado y que se ejecuta durante más de una hora.
- Una actividad que devuelve un número de filas inusualmente grande.
- Una actividad que consume una cantidad de espacio de tablas temporal inusualmente alto.

A continuación, cree umbrales que describan estos criterios y que contengan una acción COLLECT ACTIVITY DATA WITH DETAILS. Cuando se supere el umbral, la información sobre la actividad que ha superado el umbral se enviará al supervisor de sucesos ACTIVITIES activo cuando se complete la actividad.

Por ejemplo, para recopilar información sobre cualquier actividad de base de datos que se ejecute durante más de 3 horas, cree un umbral como el siguiente:

```
CREATE THRESHOLD LONGRUNNINGACTIVITIES
  FOR DATABASE ACTIVITIES ENFORCEMENT DATABASE
  WHEN ACTIVITYTOTALTIME > 3 HOURS COLLECT ACTIVITY DATA WITH DETAILS
  CONTINUE
```

La supervisión con gestor de carga de trabajo de DB2 es ligera. Procure aplicarla a un pequeño subconjunto de sus consultas, como se muestra en el ejemplo, donde sólo se supervisan las consultas en ejecución durante al menos tres horas. Puede acotar este ejemplo si crea un umbral que no esté en el nivel de base de datos global, sino en el nivel de una superclase definida por el usuario. Si esta supervisión, con un ámbito más restringido, se ajusta a su objetivo, puede reducir todavía más el coste de la supervisión, y se proporcionará información sólo en el nivel necesario:

```
CREATE SERVICE CLASS LONGQUERIES
  AGENT PRIORITY 20
  PREFETCH PRIORITY LOW

CREATE THRESHOLD LONGRUNNINGACTIVITIES2
  FOR SERVICE CLASS LONGQUERIES ACTIVITIES ENFORCEMENT DATABASE
  WHEN ACTIVITYTOTALTIME > 3 HOURS COLLECT ACTIVITY DATA WITH DETAILS
  CONTINUE
```

. Se asigna a la clase de servicio creada para el umbral una prioridad baja de captación previa y de agente, porque su uso está destinado en consultas con una ejecución larga (esta sentencia de SQL funciona en sistemas operativos UNIX y Linux; en sistemas operativos Windows, sustituya una prioridad de agente de -6).

Después de que su servidor de datos realice el trabajo, puede analizar la información grabada en los supervisores de sucesos de actividades e violaciones de umbral. El texto de las sentencias y la información del entorno de compilación de las actividades de DML también se graba en el supervisor de sucesos de actividades, de modo que pueda ejecutar **DB2 explain** en ellas para investigar con más detalle el rendimiento de la actividad.

---

## Modelado del rendimiento de gestión de carga de trabajo

La carga de trabajo de su sistema puede modelarse como un conjunto de actividades que llegan al sistema a un ritmo que rige la distribución de la velocidad de llegada de actividades (a menudo se define de forma inversa, distribución de tiempo *tiempo entre llegadas*) y la cantidad de tiempo que las actividades dedican a ejecutarse en el sistema después de una distribución de tiempo de servicio.

El tiempo entre llegadas es el tiempo entre la llegada de una actividad y la llegada de la actividad siguiente. El tiempo de servicio es el tiempo que tarda una actividad en ejecutarse en el sistema. Por ejemplo, si envía una consulta a los 0 segundos, pasa 2 segundos en la cola y finaliza a los 5 segundos, el tiempo de servicio será de  $5 - 2 = 3$  segundos. El tiempo de servicio no tiene en cuenta ningún otro trabajo en ejecución en el sistema (es decir, no es el tiempo de ejecución observado, sino el tiempo que la actividad tardaría en ejecutarse de

forma aislada). La distribución del tiempo de servicio puede ser aproximada para actividades de DML si se utiliza el coste estimado en timerons, que tiene cuenta tanto el tiempo de procesador como de E/S para una actividad.

Puede crear un modelo de carga de trabajo para su sistema midiendo la distribución del tiempo entre llegadas y la distribución del tiempo de servicio de las actividades del sistema. Las distribuciones de tiempo entre llegadas y las distribuciones de tiempo de servicio aproximado (utilizando el coste estimado) pueden obtenerse por medio de estadísticas de actividad agregadas para subclases de servicio o clases de trabajo (utilizando acciones de trabajo) y un supervisor de sucesos de estadísticas. Estas estadísticas no se recopilan por omisión. Para obtener más información, consulte:

- “Estadísticas para objetos de la gestión de carga de trabajo de DB2” en la página 265
- A gentle introduction to histograms
- Understanding the six histograms of DB2 workload management
- Visualizing and deriving statistics from DB2 histograms using SQL

---

## Ejemplo: Captura de información sobre una actividad para su posterior análisis

Puede utilizar las funciones de gestión de carga de trabajo para capturar información sobre una actividad para su posterior análisis.

Supongamos que tiene un procedimiento almacenado llamado `MYSHEMA.MYSLOWSTP` y que se está ejecutando más lentamente que de costumbre. Empieza a recibir quejas sobre esta situación y decide investigar la causa de la ralentización. Si es imposible investigar mientras se ejecuta el procedimiento, puede capturar información sobre la actividad del procedimiento almacenado y las actividades que tenga anidadas.

Si suponemos que tiene un supervisor de sucesos de actividades llamado `DB2ACTIVITIES`, puede crear una clase de trabajo para sentencias `CALL` que se apliquen al esquema del procedimiento almacenado `MYSHEMA.MYSLOWSTP`. A continuación, puede crear una acción de trabajo para correlacionar la actividad de `CALL` y todas las actividades anidadas con una clase de servicio que tenga habilitada la recopilación de actividades. La actividad `CALL` y las actividades anidadas en ella se envían al supervisor de sucesos. A continuación se muestran ejemplos del DDL necesario para crear los objetos de gestión de carga de trabajo de DB2:

```
CREATE SERVICE CLASS SC1;
CREATE WORKLOAD WL1 APPLNAME ('DB2BP') SERVICE CLASS SC1;
CREATE SERVICE CLASS PROBLEMQUERIESSC UNDER SC1 COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS;

CREATE WORK CLASS SET PROBLEMQUERIES
(WORK CLASS CALLSTATEMENTS WORK TYPE CALL ROUTINES IN SCHEMA MYSCHEMA);

CREATE WORK ACTION SET DATABASEACTIONS FOR SERVICE CLASS SC1 USING WORK CLASS SET PROBLEMQUERIES
(WORK ACTION CAPTURECALL ON WORK CLASS CALLSTATEMENTS MAP ACTIVITY WITH NESTED TO PROBLEMQUERIESSC);
```

Después de que el procedimiento almacenado `MYSHEMA.MYSLOWSTP` se ejecute, puede emitir la siguiente consulta para obtener el descriptor de contexto de aplicación, la unidad del identificador de trabajo y el identificador de la actividad:

```
SELECT AGENT_ID,
       UOW_ID,
       ACTIVITY_ID
FROM ACTIVITY_DB2ACTIVITIES
```

```

WHERE SC_WORK_ACTION_SET_ID = (SELECT ACTIONSETID
                               FROM SYSCAT.WORKACTIONSETS
                               WHERE ACTIONSETNAME = 'DATABASEACTIONS')
AND SC_WORK_CLASS_ID = (SELECT WORKCLASSID
                        FROM SYSCAT.WORKCLASSES
                        WHERE WORKCLASSNAME = 'CALLSTATEMENTS'
                        AND WORKCLASSSETID =
                        (SELECT WORKCLASSSETID FROM SYSCAT.WORKACTIONSETS WHERE ACTIONSETNAME
                        = 'DATABASEACTIONS'));

```

Si suponemos que la actividad capturada tiene un descriptor de contexto de aplicación 1, un identificador de unidad de trabajo 2 y un identificador de actividad 3, se generan los resultados siguientes:

AGENT_ID	UOW_ID	ACTIVITY_ID
=====	=====	=====
	1	2
		3

Con esta información, puede emitir la siguiente consulta en las tablas ACTIVITY\_DB2ACTIVITIES y ACTIVITYSTMT\_DB2ACTIVITIES para determinar a qué ha dedicado su tiempo la actividad:

```

WITH RAH (LEVEL, APPL_ID, PARENT_UOW_ID, PARENT_ACTIVITY_ID,
          UOW_ID, ACTIVITY_ID, STMT_TEXT, TIME_CREATED, TIME_COMPLETED) AS
  (SELECT 1, ROOT.APPL_ID, ROOT.PARENT_UOW_ID,
         ROOT.PARENT_ACTIVITY_ID, ROOT.UOW_ID, ROOT.ACTIVITY_ID,
         ROOTSTMT.STMT_TEXT, ROOT.TIME_CREATED, ROOT.TIME_COMPLETED
   FROM ACTIVITY_DB2ACTIVITIES ROOT, ACTIVITYSTMT_DB2ACTIVITIES ROOTSTMT
   WHERE ROOT.APPL_ID = ROOTSTMT.APPL_ID AND ROOT.AGENT_ID = 1
         AND ROOT.UOW_ID = ROOTSTMT.UOW_ID AND ROOT.UOW_ID = 2
         AND ROOT.ACTIVITY_ID = ROOTSTMT.ACTIVITY_ID AND ROOT.ACTIVITY_ID = 3
  UNION ALL
   SELECT PARENT.LEVEL +1, CHILD.APPL_ID, CHILD.PARENT_UOW_ID,
          CHILD.PARENT_ACTIVITY_ID, CHILD.UOW_ID,
          CHILD.ACTIVITY_ID, CHILDSTMT.STMT_TEXT, CHILD.TIME_CREATED,
          CHILD.TIME_COMPLETED
   FROM RAH PARENT, ACTIVITY_DB2ACTIVITIES CHILD,
        ACTIVITYSTMT_DB2ACTIVITIES CHILDSTMT
   WHERE PARENT.APPL_ID = CHILD.APPL_ID AND
         CHILD.APPL_ID = CHILDSTMT.APPL_ID AND
         PARENT.UOW_ID = CHILD.PARENT_UOW_ID AND
         CHILD.UOW_ID = CHILDSTMT.UOW_ID AND
         PARENT.ACTIVITY_ID = CHILD.PARENT_ACTIVITY_ID AND
         CHILD.ACTIVITY_ID = CHILDSTMT.ACTIVITY_ID AND
         PARENT.LEVEL < 64
  )
SELECT UOW_ID, ACTIVITY_ID, SUBSTR(STMT_TEXT,1,40),
       TIMESTAMPDIFF(2, CHAR(TIME_COMPLETED - TIME_CREATED)) AS
       LIFE_TIME
FROM RAH
ORDER BY UOW_ID, ACTIVITY_ID;

```

Los resultados serían parecidos a los siguientes:

UOW_ID	ACTIVITY_ID	STMT_TEXT	LIFE_TIME
=====	=====	=====	=====
2	3	CALL SLOWPROC	1000
2	4	SELECT COUNT(*) FROM ORG	1
2	5	SELECT * FROM MYHUGETABLE	999

Los resultados indican que el procedimiento almacenado dedica la mayor parte del tiempo en consultar la tabla MYHUGETABLE. Su siguiente paso será investigar qué cambios en la tabla MYHUGETABLE pueden provocar que las consultas que se realizan en ella se ralenticen.

Cuando se ejecutan muchos procedimientos almacenados al mismo tiempo, se produce una actividad general mayor al realizar el análisis. Para resolver este

problema, puede crear una carga de trabajo y una clase de servicio para ejecutar un procedimiento almacenado emitido con un identificador de autorización específico, una aplicación específica, o ambos. A continuación, puede utilizar el método anterior para analizar el comportamiento del procedimiento almacenado.

---

## Capítulo 5. Integración con los gestores de carga de trabajo del sistema operativo

Si está disponible, utilice la gestión de carga de trabajo de DB2 junto con un gestor de carga de trabajo del sistema operativo, ya que le proporcionará funciones adicionales.

La clave de la integración entre la gestión de carga de trabajo de DB2 y los gestores de carga de trabajo del sistema operativo radica en la clase de servicio de DB2. Puede crear una correlación entre una clase de servicio de DB2 y una clase de gestor de carga de trabajo del sistema operativo en el momento de definir una clase de servicio de DB2 utilizando la opción `OUTBOUND CORRELATOR` de la sentencia `CREATE SERVICE CLASS` o de la sentencia `ALTER SERVICE CLASS`.

Si se define el correlacionador de salida, todas las hebras de la clase de servicio de DB2 se asocian al gestor de carga de trabajo del sistema operativo empleando el correlacionador de salida cuando comienza la siguiente actividad.

---

### Integración del gestor de carga de trabajo de AIX con la gestión de carga de trabajo de DB2

En el sistema operativo AIX, la integración opcional entre clases de servicio de DB2 y clases de AIX WLM le permite controlar la cantidad de recursos de procesador asignados a cada clase de servicio.

Es posible que la implementación de controles AIX WLM no sea necesaria para satisfacer sus objetivos de rendimiento, pero aunque no necesite utilizar AIX WLM, las estadísticas del sistema operativo que proporciona AIX WLM por cada clase de AIX son útiles para fines de supervisión y ajuste.

AIX WLM asigna cantidades relativas o absolutas de recursos de procesador en forma de cuotas a las clases que se benefician de los controles, que puede cambiar dinámicamente y que entran en vigor inmediatamente. Si las cuotas relativas de CPU de AIX no proporcionan el nivel de control que requiere, también tiene la opción de asignar un porcentaje máximo fijo de recursos de CPU. Al hacer esto, sacrifica cierta flexibilidad de la asignación relativa de CPU, lo que es útil en horas que no son punta, pero también obtiene un control excelente y garantizado con un límite máximo fijado en la asignación de recursos de tiempo de CPU.

#### Correlaciones recomendadas entre clases de servicio de DB2 y clases de AIX

Utilice una correlación 1:1 de clases de servicio de DB2 con clases de servicio del gestor de carga de trabajo de AIX para sacar provecho de los controles de procesador AIX WLM. Al tener una correlación 1:1 entre clases de servicio de DB2 y clases de servicio del gestor de carga de trabajo de AIX, puede ajustar los recursos de procesador de AIX para cada clase de servicio de DB2 individualmente con el fin de cumplir sus objetivos de prioridades empresariales.

En la figura siguiente se muestra la integración de la gestión de carga de trabajo de DB2 con el gestor de carga de trabajo de AIX. Tenga en cuenta la correlación 1:1 entre cada clase de servicio de DB2 y la clase de servicio del gestor de carga de

trabajo de AIX en los niveles de superclase y subclase de servicio.

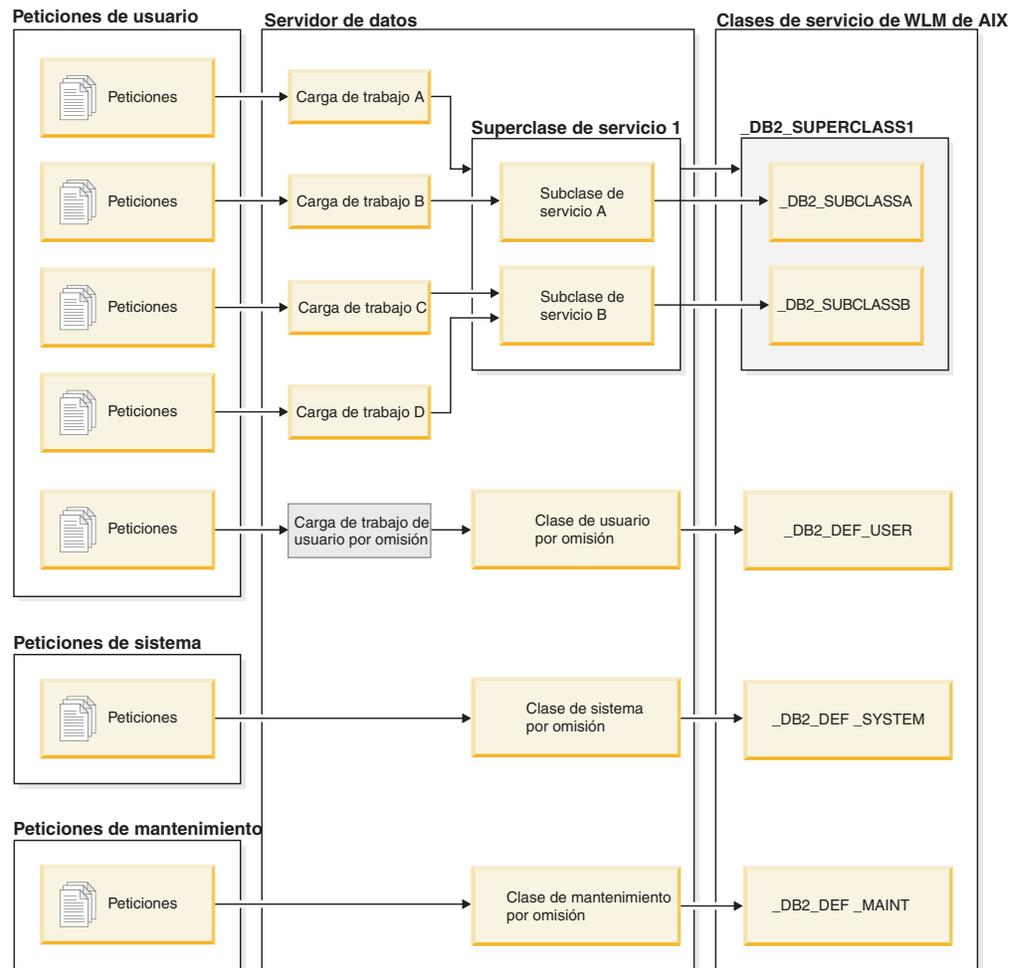


Figura 52. Integración de la gestión de carga de trabajo de DB2 con el gestor de carga de trabajo de AIX

Cuando un entorno de DB2 se compone de una única base de datos en una única instancia de DB2, como en el ejemplo de la figura anterior, es posible correlacionar directamente clases de servicio de DB2 con clases del gestor de carga de trabajo de AIX. Cada superclase de servicio de DB2 puede tener una superclase de servicio correspondiente del gestor de carga de trabajo de AIX, y cada subclase de servicio de DB2 puede correlacionarse con una subclase de servicio de AIX correspondiente.

En casos en los que el entorno de DB2 se compone de varias bases de datos e instancias de DB2, varios niveles pueden ser candidatos para el control de los recursos. Puesto que el gestor de carga de trabajo de AIX admite una jerarquía de los niveles, es decir, superclase y subclase, sólo pueden correlacionarse dos niveles de un entorno de DB2 con clases del gestor de carga de trabajo de AIX a la vez. En la figura siguiente se muestra una manera de conseguir una correlación 1:1 con varias bases de datos, cada una con varias superclases. Aquí, cada base de datos tiene su propia superclase del gestor de carga de trabajo de AIX, y cada superclase de servicio de DB2 se correlaciona con una subclase de gestor de carga de trabajo de AIX.

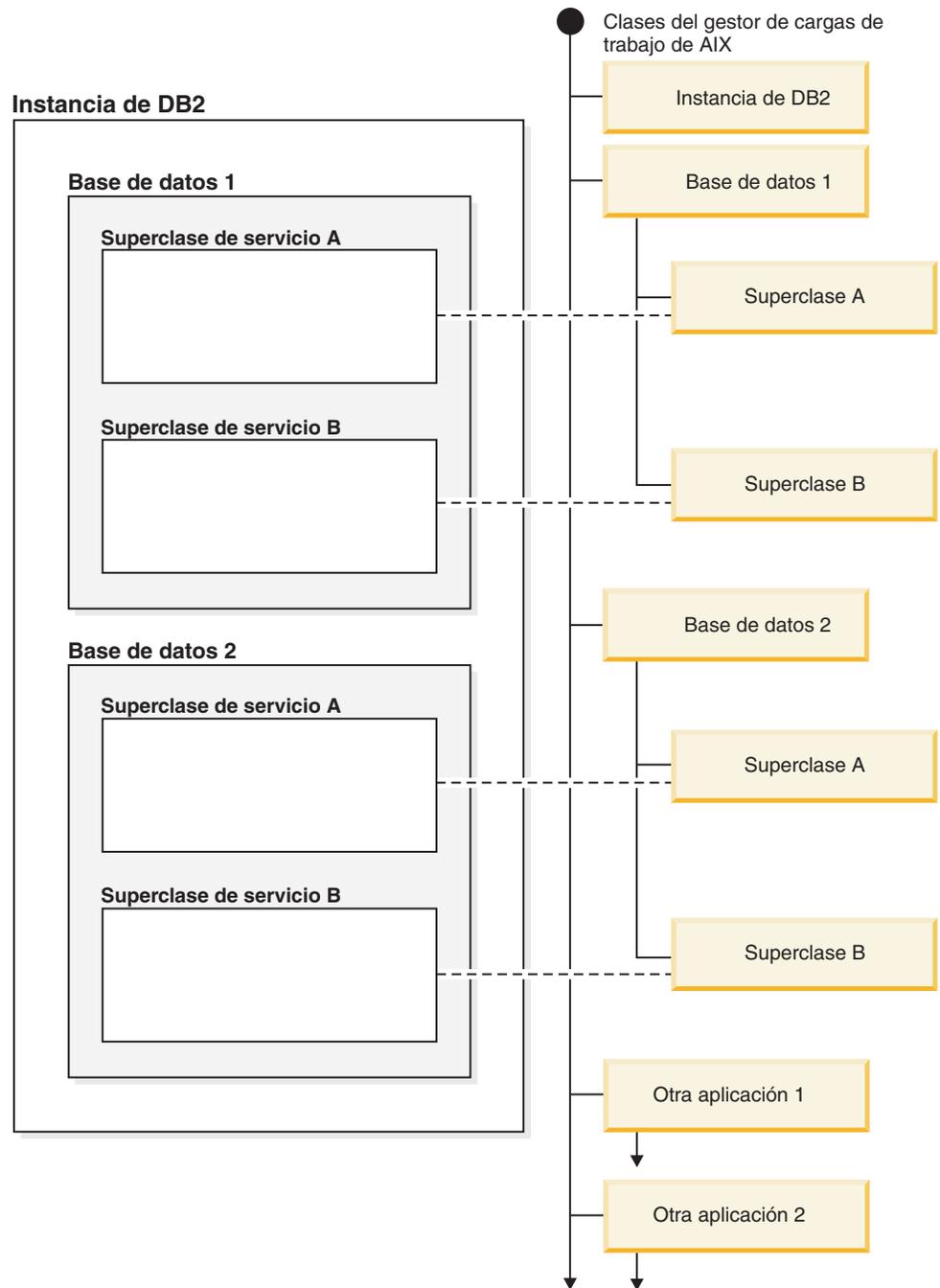


Figura 53. Clases de servicio de DB2 correlacionadas con clases de AIX (sólo con superclases de servicio de DB2)

Una configuración alternativa consiste en correlacionar cada superclase de servicio de DB2 con su propia superclase del gestor de carga de trabajo de AIX, lo que da como resultado las cuatro superclases de este ejemplo. En esta situación, en nivel de control de recursos de la base de datos se representa explícitamente en las definiciones de clase de servicio del gestor de carga de trabajo de AIX.

En la figura siguiente se muestra una correlación 1:1 en un caso en el que tiene varias bases de datos, cada una con superclases de servicio y subclases de servicio. Aquí, cada base de datos se corresponde con una superclase de AIX y cada subclase de servicio de DB2 se correlaciona con una subclase del gestor de carga

de trabajo de AIX. La superclase de servicio de DB2 no se muestra explícitamente en las definiciones de clases de servicio del gestor de carga de trabajo de AIX.

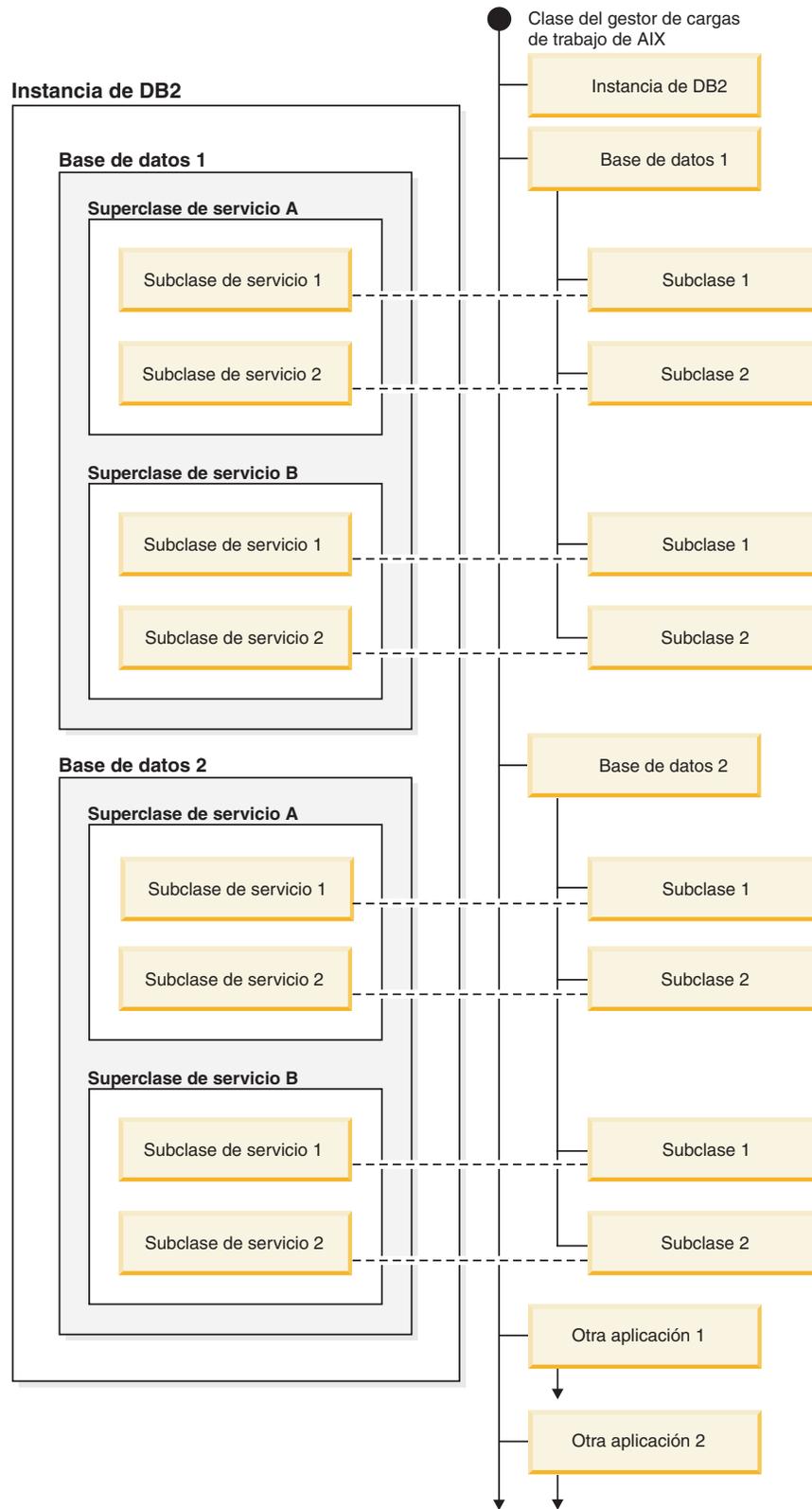


Figura 54. Clases de servicio de DB2 correlacionadas con clases de servicio del gestor de carga de trabajo de AIX (con subclases de servicio de DB2)

## Definición de correlaciones entre clases de servicio de DB2 y clases de AIX

La correlación entre clases de servicio de DB2 y clases del gestor de carga de trabajo de AIX se especifica para la clase de servicio de DB2 mediante la palabra clave `OUTBOUND CORRELATOR` de las sentencias `CREATE SERVICE CLASS` o `ALTER SERVICE CLASS`.

Los pasos para configurar las clases del gestor de carga de trabajo de AIX con el servidor de datos de DB2 son:

1. Cree las superclases y subclases de servicio de DB2 y especifique los códigos `OUTBOUND CORRELATOR`.
2. Cree las clases de AIX correspondientes.
3. Cree los archivos de normas asociados del gestor de carga de trabajo de AIX para que contengan las correlaciones de la gestión de carga de trabajo de DB2 con el gestor de carga de trabajo de AIX utilizando los códigos `OUTBOUND CORRELATOR` bajo las columnas de códigos.
4. Inicie el gestor de carga de trabajo de AIX.
5. Si es necesario, establezca esta configuración del gestor de carga de trabajo de AIX como activa.

Cuando una hebra se incorpore a una clase de servicio de DB2, el servidor de datos de DB2 llamará a la API apropiada del gestor de carga de trabajo de AIX para asociar la hebra con la clase de servicio de AIX correspondiente. El servidor de datos de DB2 envía la clase de servicio de AIX de destino de la hebra al gestor de carga de trabajo de AIX proporcionándole el conjunto de códigos de la aplicación en el parámetro `OUTBOUND CORRELATOR`.

Debe asegurarse de que el gestor de carga de trabajo de AIX esté correctamente instalado, configurado y activo. Si el servidor de datos de DB2 que no se puede comunicar con el gestor de carga de trabajo de AIX, se anotará un mensaje en los archivos de anotaciones cronológicas de `db2diag` y en las anotaciones cronológicas del administrador de DB2. La actividad de la base de datos continuará.

El servidor de datos de DB2 no puede detectar si el gestor de carga de trabajo de AIX reconoce el valor de `OUTBOUND CORRELATOR` que proporciona al gestor de carga de trabajo de AIX. Debe verificar que el valor especificado para la clase de servicio de DB2 coincide con los códigos de aplicación que correlacionan las hebras de DB2 con las clases de servicio de AIX. Si el gestor de carga de trabajo de AIX no reconoce el valor de `OUTBOUND CORRELATOR`, la actividad de la base de datos prosigue su ejecución.

Otros aspectos para tener en cuenta:

- Las clases de servicio de DB2 no funcionan con la característica de herencia del gestor de carga de trabajo de AIX. La herencia es el valor por omisión para una clase de servicio de AIX; la herencia debe inhabilitarse explícitamente estableciendo el atributo de herencia en `N0`. La herencia del gestor de carga de trabajo de AIX obliga a que todas las hebras y procesos secundarios se correlacionen con la misma clase que la hebra o proceso principal. Si se habilita la herencia, la gestión de carga de trabajo de DB2 no puede cambiar la clase del gestor de carga de trabajo de AIX de una hebra utilizando códigos. Esta restricción hace que la integración de la gestión de carga de trabajo de DB2 y el gestor de carga de trabajo de AIX quede inutilizable. El servidor de datos de

DB2 no puede detectar si la herencia del gestor de carga de trabajo de AIX está habilitada, por lo que no emite un mensaje de error si la herencia está habilitada.

- Las clases de servicio de DB2 no son compatibles con la característica de asignación manual del gestor de carga de trabajo de AIX. Con la característica de asignación manual, los usuarios pueden asignar manualmente un proceso a una clase específica del gestor de carga de trabajo de AIX. Al asignar manualmente el proceso de DB2, todas las hebras del proceso se asignan a una clase del gestor de carga de trabajo de AIX de destino, se vulnera la lógica de correlación de clases de servicio de DB2 y los resultados son impredecibles.

Para obtener más información sobre el gestor de carga de trabajo de AIX, consulte el Centro de información de AIX en <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp>

## **Definición de controles de procesador en clases de AIX**

El gestor de carga de trabajo de AIX puede utilizarse para controlar la cantidad de recursos de procesador asignados a cada clase de servicio. Las opciones incluyen la definición de una cuota mínima, máxima o relativa de recursos de procesador para cada clase de servicio.

Cuando se integra el gestor de carga de trabajo de AIX con la gestión de carga de trabajo de DB2, sólo se admite la asignación de recursos de procesador. No debe establecer valores de memoria y E/S para las clases de AIX. La memoria de nivel de base de datos de DB2 se comparte entre todos los agentes de las diferentes clases de servicio de DB2, por lo que no puede dividir la asignación de memoria entre diferentes clases de servicio. El control de E/S de nivel de AIX no admite el modelo de hebras del motor de DB2. Para controlar las E/S, puede utilizar el atributo de prioridad de captación previa de una clase de servicio de DB2 para diferenciar las prioridades de E/S entre diferentes clases de servicio de DB2.

Si utiliza AIX para controlar la cantidad de recursos de procesador asignados a una clase de servicio, no cambie el valor de prioridad del agente para esa clase de servicio de DB2. Utilice únicamente uno de los mecanismos para controlar el acceso al recurso de procesador. No puede establecer tanto el valor AGENT PRIORITY como el valor OUTBOUND CORRELATOR para una clase de servicio. Consulte "Prioridad de agente de clases de servicio" en la página 82 para obtener más información.

Los valores del gestor de carga de trabajo de AIX debe ser coherentes en todos los sistemas físicos que participan en una instancia. Por ejemplo, si el valor de recursos para una clase de servicio de AIX es alto en un sistema, debe utilizarse el mismo valor para esa clase de servicio de AIX en todos los demás sistemas. Si los valores de uso de recursos son incoherentes entre los sistemas, las peticiones que se ejecuten en la misma clase de servicio de AIX mostrarán diferentes niveles de rendimiento en los diferentes miembros de base de datos. Esta situación puede provocar un rendimiento general deficiente para las conexiones en una clase de servicio de AIX.

---

## Integración de la gestión de carga de trabajo de Linux con la gestión de carga de trabajo de DB2

En el sistema operativo Linux, la integración opcional entre las clases de servicio de DB2 y las clases de Linux (grupos de control) permiten controlar la cantidad de recursos de procesador que hay asignados a cada clase de servicio. Si están habilitadas, todas las hebras que se ejecutan en una clase de servicio de DB2 se correlacionan a una clase de Linux donde quedan sujetos a los controles de recursos de procesador que tenga definidos.

Para contar con el soporte de gestión de carga de trabajo de Linux, se necesita la versión 2.6.26 o posterior del kernel de Linux en un sistema de 64 bits y el paquete de biblioteca `libcgroup`.

La gestión de carga de trabajo de Linux admite una jerarquía de clases con superclases y subclases, con cuotas de procesador para las subclases que se dividen proporcionalmente según las cuotas de la clase padre. Estas cuotas proporcionan un método de control sobre los recursos de procesador, de tal manera que todas las hebras del sistema siempre se ejecutarán, pero la cantidad de tiempo de procesador que recibe cada hebra depende del número de cuotas asignadas a la clase de Linux.

Los recursos de procesador en el sistema operativo Linux se asignan en cuotas relativas a la clase por omisión de gestión de carga de trabajo de Linux, que por omisión tiene una cuota de procesador con un valor de 1024. Si no define ninguna otra clase de Linux, todas las hebras se ejecutan en esta clase por omisión. Si define una clase con un valor de cuota igual a 1024, esta clase recibe la misma cantidad de recursos de procesador que la clase por omisión de Linux con la cuota de procesador por omisión. De igual forma, una clase con una cuota de 2048 recibe una cuota de uso del procesador de destino que es el doble de la cuota de la clase por omisión. En sistemas más complejos, deberá plantearse aumentar la cuota de procesador de la clase por omisión de Linux, lo que mejora la granularidad de las cuotas en todo el sistema, para poder asignar recursos de procesador con más precisión.

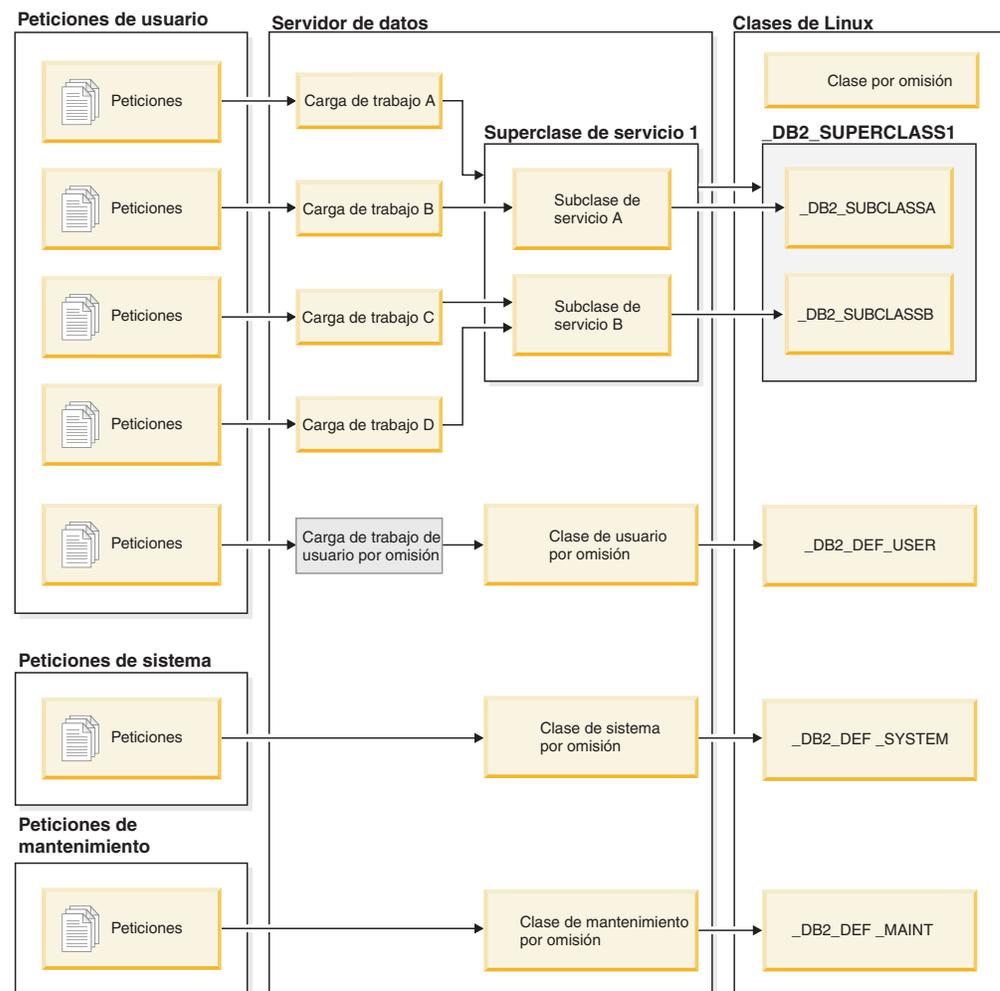
### Correlaciones recomendadas entre las clases de servicio de DB2 y las clases de Linux

Debe utilizar una correlación de 1:1 entre las clases de servicio de DB2 y las clases de Linux, lo cual le permitirá ajustar las cuotas de procesador de Linux asignadas a las actividades de cada clase de servicio de DB2 individualmente según la prioridad empresarial. Conviene que asocie cada clase de servicio de DB2 con una clase de Linux WLM, ya sea estableciendo un correlacionador de salida para cada superclase y subclase de servicio, o a través de la herencia de la clase de servicio padre para subclases. Esto incluye las clases de servicio por omisión `SYSDEFAULTSYSTEMCLASS`, `SYSDEFAULTMAINTENANCECLASS` y `SYSDEFAULTUSERCLASS`.

La figura siguiente muestra de qué manera dos subclases de servicio de DB2 bajo la misma superclase de servicio definida por el usuario pueden correlacionarse como 1:1 con las subclases de Linux bajo una superclase común. En este ejemplo, el trabajo identificado y asignado por dos cargas de trabajo para cada subclase de servicio de DB2 está sujeto a los controles de recursos de procesador que imponen las subclases de Linux correspondientes (`_DB2_SUBCLASSA`, `_DB2_SUBCLASSB`). También se muestran tres clases de Linux que corresponden a las clases de servicio

por omisión de la gestión de carga de trabajo de DB2 (`_DB2_DEF_USER`, `_DB2_DEF_SYSTEM`, `_DB2_DEF_MAINT`). Si integra la gestión de carga de trabajo de DB2 con la gestión de carga de trabajo de Linux, deberá crear estas clases adicionales de Linux para que coincidan con las clases de servicio de DB2 por omisión. Para evitar cuellos de botella, la clase de Linux que corresponde a la clase del sistema por omisión de DB2 debe recibir más cuotas de procesador que ninguna otra clase de Linux con las que se correlacionen las actividades de DB2, mientras que la clase de Linux que corresponde a la clase de mantenimiento por omisión debe recibir menos cuotas de procesador.

Figura 55. Integración de la gestión de carga de trabajo de DB2 con la gestión de carga de trabajo de Linux



### Definición de correlaciones entre clases de servicio de DB2 y clases de la gestión de carga de trabajo de Linux

Los pasos para integrar la gestión de carga de trabajo de DB2 con la gestión de carga de trabajo de Linux, que se ejecuta como un servicio del sistema operativo, son los siguientes:

1. Definir las clases de Linux, los permisos de clase y las cuotas de procesador a través de la edición del archivo de configuración de grupos de control `/etc/cgconfig.conf`. Las clases de Linux que cree dependerán de las

condiciones que dicten las prioridades de su empresa para el trabajo que realiza su servidor de datos. Si desea aplicar los recursos de procesador en función del origen de un determinado trabajo, por ejemplo, cree una clase de Linux que coincida con la clase de servicio de DB2 a la que se asignará el trabajo mediante la carga de trabajo que identifica ese trabajo. Defina una entrada para cada clase de Linux que corresponda a la clase de servicio de DB2 que vaya a crear y que desee utilizar para la correlación. Deberá incluir las secciones siguientes en el archivo de configuración `/etc/cgconfig.conf`:

- `group`: Nombre de clase de Linux. Por ejemplo, si especifica `group _class1`, se crea una superclase `_class1`. Si especifica `group _class1/_subclass1`, se crea la subclase `_subclass1` bajo la superclase `_class1`.
  - `perm`: Sección de permisos que determina quién puede controlar qué hebras se asignan a una clase de Linux y quién puede cambiar las cuotas de procesador de las clases en el archivo de configuración `/etc/cgconfig.conf`.
    - `task`: ID de usuario (**uid**) e ID de grupo (**gid**) cuyas hebras pueden ejecutarse en la clase de gestión de carga de trabajo de Linux. Para permitir que la gestión de carga de trabajo de Linux trabaje con la gestión de carga de trabajo de DB2, establezca en **uid** el ID de usuario del propietario de la instancia de DB2.
    - `admin`: ID de usuario (**uid**) e ID de grupo (**gid**) que puede cambiar las cuotas de procesador para una clase de gestión de carga de trabajo de Linux.
  - `cpu`: Sección donde se definen las cuotas de procesador.
    - `cpu.shares`: Cuota que se asigna a esta clase de Linux relativa a la clase por omisión.

El archivo de configuración `/etc/cgconfig.conf` debe contener estas secciones con el formato siguiente:

```
# Superclass name
group _name
{
    perm
    {
        task
        {
            uid = db2inst1;
            gid = db2iadml;
        }
        admin
        {
            uid = db2inst1;
            gid = db2iadml;
        }
    }

    cpu
    {
        cpu.shares = 1024;
    }
}
```

2. Inicie el daemon del servicio de gestión de carga de trabajo de Linux con el mandato **service cgconfig start**; luego inicie el servidor de datos de DB2 con el mandato **db2start**.
3. Para correlacionar una clase de servicio de DB2 con una de las clases de Linux, incluya el nombre de la clase de Linux en la cláusula `OUTBOUND CORRELATOR` cuando cree o modifique la clase de servicio, lo cual asocia las hebras de la clase de servicio de DB2 con la clase de Linux externa.

4. Si desea averiguar qué hebras están asignadas a una clase de Linux en particular, puede utilizar el mandato `cat` en el archivo `/cgroup/nombre_clase/tasks`, donde *nombre\_clase* representa el nombre de la clase de Linux en la que está interesado. Todas las hebras que no estén correlacionadas con una clase de Linux definida por el usuario se asignan a la clase de Linux por omisión, que puede encontrar en `MOUNTPOINT/sysdefault`, donde `MOUNTPOINT` está definido en el archivo de configuración `cgconfig.conf`.
5. Para añadir o eliminar clases de Linux, detenga el servicio de gestión de carga de trabajo de Linux con el mandato **`service cgconfig stop`**, realice los cambios oportunos y reinicie el servicio. Tenga en cuenta que al detener el servicio todo el sistema se verá afectado, porque todas las tareas se pasarán a la clase por omisión. Si utilizó el script `/etc/init.d/cgred` para iniciar el daemon de servicio, emita **`/etc/init.d/cgred stop`** para detenerlo.

Para lograr la integración con la gestión de carga de trabajo de DB2, asegúrese de que el servicio de gestión de carga de trabajo de Linux está bien instalado, configurado y activo. Si el servidor de datos de DB2 no puede comunicarse con el servicio de gestión de carga de trabajo de Linux, se anotará un mensaje en los archivos de anotaciones cronológicas `db2diag` y en el archivo de anotaciones cronológicas del administrador de DB2. Las actividades de la base de datos seguirán ejecutándose.

El servidor de datos de DB2 no puede detectar si la gestión de carga de trabajo de Linux puede reconocer al correlacionador de salida que pasa a los gestores de carga de trabajo externos. Verifique si el valor de `OUTBOUND CORRELATOR` especificado para una clase de servicio de DB2 coincide con el nombre de clase de Linux de manera que las hebras de DB2 se correlacionen con la clase de Linux. Si no se reconoce el correlacionador de salida, las actividades de la base de datos seguirán ejecutándose.

## Ejemplo

El ejemplo siguiente ilustra de qué manera puede utilizar los controles del procesador de gestión de carga de trabajo de Linux integrándolos con la gestión de carga de trabajo de DB2. En este ejemplo, hemos creado dos clases de servicio de DB2 definidas por el usuario, una para aplicaciones por lotes (`BATCHAPPS`) y otra para aplicaciones en línea (`ONLINEAPPS`). Para simplificar las cosas, este ejemplo no muestra las clases de servicio por omisión, que deberían incluirse en una implementación que crea la correlación 1:1 recomendada entre las clases de servicio de DB2 y las clases de Linux. Puesto que el tiempo de respuesta es crítico en las aplicaciones en línea, deseamos que la clase de servicio `ONLINEAPPS` reciba el triple de cuotas de procesador relativas al trabajo que se ejecuta en la clase de Linux por omisión ( $3 \times 1024 = 3072$  cuotas). Las aplicaciones por lotes tienen una prioridad empresarial menor, y a la clase `BATCHAPPS` debe asignársele la mitad de los recursos de trabajo de procesador que se ejecutan en la clase de Linux por omisión ( $1024 / 2 = 512$  cuotas). El resto del trabajo del sistema se ejecuta en la clase de Linux por omisión. Tenga en cuenta que en este ejemplo no se crean clases de Linux que se correspondan con las tres clases por omisión de servicio de gestión de carga de trabajo de DB2.

Para crear esta configuración, primero debe crear las dos clases de Linux correspondientes, `_BATCHAPPS` y `_ONLINEAPPS`, y luego establecer sus cuotas de procesador relativas editando el archivo de tareas `/etc/cgconfig.conf`. Después de la edición, el archivo de tareas contiene las dos entradas siguientes, una para cada clase de Linux:

```

# Superclass ONLINEAPPS
group _ONLINEAPPS
{
    perm
    {
        task
        {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin
        {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }

    cpu
    {
        # 3 x 1024 = 3072 cuotas
        cpu.shares = 3072;
    }
}

# Superclass BATCHAPPS
group _BATCHAPPS
{
    perm
    {
        task
        {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin
        {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }

    cpu
    {
        # 1024 / 2 = 512 cuotas
        cpu.shares = 512;
    }
}

```

El tiempo absoluto de procesador, en forma de porcentaje, que se asigna a cada clase de Linux como cuotas de procesador es el siguiente:

*Tabla 62. Cuotas de procesador y tiempo absoluto de procesador asignados a las clases de Linux*

Clase de Linux	Cuotas	Tiempo absoluto de procesador, en porcentaje
Clase por omisión	1024 (por omisión)	1024 / 4608 = 22%
_ONLINEAPPS	1024 x 3 = 3072	3072 / 4608 = 67%
_BATCHAPPS	1024 x $\frac{1}{2}$ = 512	512 / 4608 = 11%
	Total = 1024 + 3072 + 512 = 4608 cuotas	

Una vez creadas las clases de Linux WLM, puede iniciar el servicio de gestión de carga de trabajo de Linux:

```
service cgconfig start
```

A continuación, cree las clases de servicio de DB2 asociadas con las sentencias siguientes:

```
DB2 CREATE SERVICE CLASS BATCHAPPS OUTBOUND CORRELATOR '_BATCHAPPS'  
DB2 CREATE SERVICE CLASS ONLINEAPPS OUTBOUND CORRELATOR '_ONLINEAPPS'
```

Para averiguar qué hebras se están ejecutando en una clase de Linux, emita el mandato `cat`. Para la clase `_ONLINEAPPS` de Linux clave de la empresa, el mandato y el aspecto de la salida son los siguientes. Observará que hay seis hebras ejecutándose en esta clase de Linux:

```
cat /cgroup/_ONLINEAPPS/tasks
```

```
1056  
1087  
1107  
985  
1036  
1205
```



---

## Capítulo 6. Guía de aprendizaje de gestión de carga de trabajo de DB2

Los ejercicios de esta guía de aprendizaje se han diseñado para proporcionarle una introducción práctica a la gestión de carga de trabajo de DB2. En cada ejercicio se destaca una o varias de las funciones de gestión de carga de trabajo que están disponibles con la gestión de carga de trabajo de DB2.

Estos ejercicios proporcionan directrices para utilizar las funciones de gestión de carga de trabajo de DB2 que puede adaptar en función de sus objetivos, pero debe tener en cuenta que la configuración inicial que selecciona para su propio servidor de datos podría diferir y debería basarse en sus objetivos de gestión de carga de trabajo específicos.

### Antes de empezar

Esta guía de aprendizaje se ha diseñado para ejecutarse con la base de datos SAMPLE y, a menos que se indique algo distinto, requiere autorización DBADM o WLMADM (o autorización SQLADM si sólo se especifica la cláusula COLLECT ACTIVITY DATA). También ha de iniciar la instancia y activar la base de datos SAMPLE antes de continuar:

```
db2start
db2 activate db sample
```

Algunas de las sentencias de mandatos y de consultas que se muestran en estos ejercicios son bastantes largas. Encontrará la mayor parte de estas sentencias en el archivo de texto `wlm-tutorial-steps.txt`, que podrá copiar mientras trabaja con los ejercicios. También se incluyen los scripts que representan las cargas de trabajo necesarias para los distintos ejercicios.

El archivo `wlm-tutorial-steps.txt` y los scripts de carga de trabajo se encuentran aquí.

---

### Ejercicio 1: Iniciación a la supervisión básica utilizando objetos de la gestión de carga de trabajo de DB2 por omisión

Este ejercicio muestra los tipos básicos de información de supervisión que pueden obtenerse de los objetos de carga de trabajo y de clase de servicio por omisión.

Tiempo estimado: 20-25 minutos

Por omisión, la carga de trabajo de usuario (SYSDEFAULTUSERWORKLOAD) y una clase de servicio de usuario (SYSDEFAULTUSERCLASS) siempre se crean para cada base de datos. Estos objetos por omisión pueden utilizarse para beneficiarse de las nuevas funciones de supervisión de la gestión de carga de trabajo de DB2 sin tener que crear ninguna clase de trabajo o clase de servicio definidas por el usuario. Si no se ha creado ninguna carga de trabajo ni clase de servicio definidas por el usuario, todas las actividades de usuario se asociarán a estos objetos por omisión.

En este ejercicio se muestran dos funciones de supervisión distintas:

1. La capacidad de recopilar estadísticas agregadas para todas las actividades que se ejecutan en una clase de servicio. Las estadísticas de actividades agregadas proporcionan una forma asequible de consultar el trabajo de una clase de servicio en conjunto. Muestran información como el número de actividades que se han ejecutado en la clase de servicio y el promedio de vida útil de esas actividades.
2. La capacidad de capturar información acerca de actividades individuales. La información de actividad puede ser útil al investigar el rendimiento o el comportamiento de una actividad en particular. La información de actividad incluye datos como el texto de la sentencia, el entorno de compilación, etc. La información de actividad ofrece una recopilación más asequible que las estadísticas de actividades agregadas y suele destinarse a un subconjunto específico de actividades.

## Paso 1: Crear y habilitar supervisores de sucesos

Conéctese con la base de datos y cree y habilite supervisores de sucesos para actividades y estadísticas.

```
CONNECT TO SAMPLE
```

```
CREATE EVENT MONITOR DB2ACTIVITIES FOR ACTIVITIES WRITE TO TABLE  
CREATE EVENT MONITOR DB2STATISTICS FOR STATISTICS WRITE TO TABLE
```

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1  
SET EVENT MONITOR DB2STATISTICS STATE 1
```

## Paso 2: Recopilar actividades individuales

Habilite la recopilación de actividades individuales mediante la utilización de la cláusula `COLLECT ACTIVITY DATA` en `CREATE` o `ALTER WORKLOAD STATEMENT`. Cuando se especifica la cláusula `COLLECT ACTIVITY DATA` para una carga de trabajo, la información relacionada con cualquier actividad que envíe una aparición de esa carga de trabajo se enviará al supervisor de sucesos `ACTIVITIES` activo cuando se haya completado la actividad. La cláusula `COLLECT ACTIVITY DATA` le permite especificar cuánta información ha de recopilarse aplicando una de las opciones siguientes:

- `WITHOUT DETAILS`: Recopila información de actividad, sin la sentencia ni el entorno de compilación.
- `WITH DETAILS`: Recopila información de actividad, incluidos la sentencia y el entorno de compilación.
- `WITH DETAILS AND VALUES`: Recopila información de actividad, incluidos la sentencia y el entorno de compilación, y valores de datos de entrada.

Para este ejercicio, especificaremos la cláusula `WITH DETAILS` para que se capture la información del texto de la sentencia.

```
ALTER WORKLOAD SYSDEFAULTUSERWORKLOAD  
COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS
```

En este ejemplo se recopilan datos de actividad para la carga de trabajo de usuario por omisión. El resultado es la obtención de información acerca de todas las actividades de usuario que están recopilándose, pues ninguna otra carga de trabajo definida por el usuario está actualmente activa. Ello sería demasiado caro en un entorno de producción. Un enfoque más adecuado consistiría en aislar las actividades que interesan mediante la utilización de una carga de trabajo definida por el usuario específica o una clase de servicio y aplicar la cláusula `COLLECT ACTIVITY DATA` sólo a esa carga de trabajo o clase de servicio.

Información adicional: la cláusula COLLECT ACTIVITY DATA también puede especificarse en una clase de servicio, clase de trabajo (mediante una acción de trabajo) o un umbral. Si la cláusula se especifica para una clase de servicio, se recopilará información para cualquier actividad que se ejecute en la clase de servicio. Si se especifica para una clase de trabajo (mediante una acción de trabajo), se recopilará cualquier actividad a la que se aplica la acción de trabajo. Si la cláusula se especifica para un umbral, se recopilará información de actividad si se ha infringido el umbral.

### Paso 3: Recopilar estadísticas de actividades agregadas

Habilite la recopilación de estadísticas de actividades agregadas para la subclase por omisión de la clase de servicio de usuario por omisión mediante la utilización de la cláusula COLLECT AGGREGATE ACTIVITY DATA. Cuando se especifique esta cláusula, las estadísticas agregadas se mantendrán en memoria para la clase de servicio correspondiente (por ejemplo, estadísticas como el promedio de vida útil de actividad). Las estadísticas pueden verse utilizando la función de tabla de estadísticas de subclase de servicio o pueden recopilarse y enviarse al supervisor de sucesos de estadísticas activo para su posterior análisis.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
    COLLECT AGGREGATE ACTIVITY DATA BASE
```

Información adicional: existe un conjunto de estadísticas que se recopilan por omisión para todos los objetos de la gestión de carga de trabajo de DB2. La cláusula COLLECT AGGREGATE ACTIVITY DATA habilita la recopilación de varias estadísticas opcionales adicionales, como el histograma de vida útil de actividad.

En este ejemplo, todas las actividades de usuario se ejecutarán en la subclase de servicio SYSDEFAULTSUBCLASS de la superclase de servicio SYSDEFAULTUSERCLASS porque no se ha creado ninguna clase de servicio definida por el usuario. Por lo tanto, se recopilará información para todas las actividades de usuario.

### Paso 4: Ejecutar varias actividades

Ejecute varias actividades, lo que dará como resultado la actualización de las estadísticas y la recopilación de las actividades.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
```

Los scripts que representan aplicaciones (como work1.db2 y work2.db2) le desconectan de la base de datos, por lo tanto, tras su ejecución, deberá volver a conectarse.

### Paso 5: Ver estadísticas

Puede ver las estadísticas de clase de servicio utilizando la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS. Por ejemplo:

```
CONNECT TO SAMPLE
```

```
SELECT VARCHAR(SERVICE_SUPERCLASS_NAME, 30) AS SUPERCLASS,
       VARCHAR(SERVICE_SUBCLASS_NAME, 30) AS SUBCLASS,
       LAST_RESET,
       COORD_ACT_COMPLETED_TOTAL,
       COORD_ACT_REJECTED_TOTAL,
```

```

        COORD_ACT_ABORTED_TOTAL,
        COORD_ACT_LIFETIME_AVG
FROM TABLE(SYSPROC.WLM_GET_SERVICE_SUBCLASS_STATS('SYSDEFAULTUSERCLASS',
        'SYSDEFAULTSUBCLASS', -1)) AS T

```

La salida de esta consulta tendrá un aspecto similar al siguiente:

```

SUPERCLASS                SUBCLASS                LAST_RESET
COORD_ACT_COMPLETED_TOTAL COORD_ACT_REJECTED_TOTAL COORD_ACT_ABORTED_TOTAL
COORD_ACT_LIFETIME_AVG
-----
-----
-----
SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS      2007-07-18-16.03.51.752190
74                        0                        0      +1.402880000000000E+002

1 registro(s) seleccionado(s).

```

La columna `COORD_ACT_COMPLETED_TOTAL` cuántas actividades se han completado correctamente en esta clase de servicio. La hora de último restablecimiento indica la última hora en la que las estadísticas de esta clase de servicio se han restablecido.

Información adicional: si no habilita las estadísticas de actividades agregadas para una clase de servicio utilizando la cláusula `COLLECT AGGREGATE ACTIVITY DATA`, algunas de las estadísticas de las que informa la función de tabla `WLM_GET_SERVICE_SUBCLASS_STATS` serán nulas.

## Paso 6: Enviar las estadísticas al supervisor de sucesos

Utilice el procedimiento almacenado `WLM_COLLECT_STATS` para enviar las estadísticas de todos objetos de la gestión de carga de trabajo de DB2 al supervisor de sucesos de estadísticas activo. Cuando se recopilan y envían estadísticas al supervisor de sucesos de estadísticas, los valores se restablecen.

```
CALL SYSPROC.WLM_COLLECT_STATS()
```

Información adicional: si no existe ningún supervisor de sucesos de estadísticas activo, todavía puede utilizar el procedimiento `WLM_COLLECT_STATS` para restablecer las estadísticas, pero los valores actuales se perderán. Es posible automatizar la recopilación de estadísticas de gestión de carga de trabajo utilizando el parámetro de configuración de base de datos `WLM_COLLECT_INT`. Si establece este parámetro en un valor distinto de cero, las estadísticas de gestión de carga de trabajo se recopilarán automáticamente cada `wlm_collect_int` minutos (como si invocara manualmente el procedimiento `WLM_COLLECT_STATS` cada `wlm_collect_int` minutos).

## Paso 7: Volver a ver las estadísticas

Vuelva a invocar la función de tabla `WLM_GET_SERVICE_SUBCLASS_STATS`. Observe que la indicación de fecha y hora de `LAST_RESET` se ha actualizado y que las estadísticas se han restablecido.

```

SELECT VARCHAR(SERVICE_SUPERCLASS_NAME, 30) AS SUPERCLASS,
        VARCHAR(SERVICE_SUBCLASS_NAME, 30) AS SUBCLASS,
        LAST_RESET,
        COORD_ACT_COMPLETED_TOTAL,
        COORD_ACT_REJECTED_TOTAL,
        COORD_ACT_ABORTED_TOTAL,
        COORD_ACT_LIFETIME_AVG
FROM TABLE(SYSPROC.WLM_GET_SERVICE_SUBCLASS_STATS('SYSDEFAULTUSERCLASS',
        'SYSDEFAULTSUBCLASS', -1)) AS T

```

La salida tendrá un aspecto similar al siguiente:

```

SUPERCLASS                SUBCLASS                LAST_RESET
COORD_ACT_COMPLETED_TOTAL COORD_ACT_REJECTED_TOTAL
COORD_ACT_ABORTED_TOTAL  COORD_ACT_LIFETIME_AVG
-----
SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS      2007-07-18-
16.04.03.505818          0                        0
0 +0.000000000000000E+000

```

1 registro(s) seleccionado(s).

## Paso 8: Ver las estadísticas de clase de servicio que el supervisor de sucesos de estadísticas ha recopilado

El procedimiento WLM\_COLLECT\_STATS ha enviado las estadísticas de clase de servicio al supervisor de sucesos de estadísticas. Puede ver las estadísticas que el supervisor de sucesos ha recopilado utilizando una sentencia como la siguiente:

```

SELECT VARCHAR(SERVICE_SUPERCLASS_NAME, 30) AS SUPERCLASS,
       VARCHAR(SERVICE_SUBCLASS_NAME, 30) AS SUBCLASS,
       LAST_WLM_RESET,
       STATISTICS_TIMESTAMP,
       COORD_ACT_COMPLETED_TOTAL,
       COORD_ACT_REJECTED_TOTAL,
       COORD_ACT_ABORTED_TOTAL,
       COORD_ACT_LIFETIME_AVG
FROM SCSTATS_DB2STATISTICS

```

La salida tendrá un aspecto similar al siguiente:

```

SUPERCLASS                SUBCLASS                LAST_WLM_RESET
STATISTICS_TIMESTAMP
COORD_ACT_COMPLETED_TOTAL COORD_ACT_REJECTED_TOTAL
COORD_ACT_ABORTED_TOTAL  COORD_ACT_LIFETIME_AVG
-----
SYSDEFAULTSYSTEMCLASS    SYSDEFAULTSUBCLASS      2007-07-18-
16.03.46.333724 2007-07-18-16.04.03.505818 0
0 0 -1
SYSDEFAULTMAINTENANCECLASS SYSDEFAULTSUBCLASS      2007-07-18-
16.03.46.334301 2007-07-18-16.04.03.505818 0
0 0 -1
SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS      2007-07-18-
16.03.51.752190 2007-07-18-16.04.03.505818 75
0 0 136

```

3 registro(s) seleccionado(s).

Cada vez que se envíen estadísticas al supervisor de sucesos se creará un registro de estadísticas para cada objeto de la gestión de carga de trabajo de DB2. Observe las dos indicaciones de fecha y hora LAST\_WLM\_RESET y STATISTICS\_TIMESTAMP. El intervalo de tiempo desde LAST\_WLM\_RESET hasta STATISTICS\_TIMESTAMP indica el período de tiempo durante el que se han recopilado las estadísticas de ese registro. STATISTICS\_TIMESTAMP indica cuándo se han recopilado las estadísticas. Observe que el promedio de vida útil de las actividades del coordinador es -1 para las clases de servicio de mantenimiento y del sistema por omisión. Las estadísticas del promedio de vida útil de actividad sólo se mantienen para una clase de servicio si las estadísticas de actividades agregadas se han habilitado mediante la utilización de la cláusula COLLECT AGGREGATE ACTIVITY DATA.

## Paso 9: Ver información de actividad

El supervisor de sucesos de actividad también recopila información acerca de cada actividad individual asociada a la carga de trabajo de usuario omisión, debido a la especificación de la cláusula COLLECT ACTIVITY DATA en la carga de trabajo por omisión en el Paso 2. Puede consultar esta información de actividad utilizando una consulta como la siguiente:

```
SELECT VARCHAR(A.APPL_NAME, 15) as APPL_NAME,
        VARCHAR(A.TPMON_CLIENT_APP, 20) AS CLIENT_APP_NAME,
        VARCHAR(A.APPL_ID, 30) as APPL_ID,
        A.ACTIVITY_ID,
        A.UOW_ID,
        VARCHAR(S.STMT_TEXT, 300) AS STMT_TEXT
FROM ACTIVITY_DB2ACTIVITIES AS A,
     ACTIVITYSTMT_DB2ACTIVITIES AS S
WHERE A.APPL_ID = S.APPL_ID AND
      A.ACTIVITY_ID = S.ACTIVITY_ID AND
      A.UOW_ID = S.UOW_ID
```

La salida tendrá un aspecto similar al siguiente:

```
APPL_NAME      CLIENT_APP_NAME  APPL_ID
ACTIVITY_ID    UOW_ID          STMT_TEXT
-----
-----
-----
-----
-----
db2bp          CLP wlmmonbasic.db2 *LOCAL.db2inst1.070718200344
1              8 ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER
SYSDEFAULTUSERCLASS COLLECT AGGREGATE ACTIVITY DATA BASE
db2bp          CLP work1.db2      *LOCAL.db2inst1.070718200352
1              1 values(current client_applname)
db2bp          CLP work1.db2      *LOCAL.db2inst1.070718200352
2              1 select * from org
db2bp          CLP work1.db2      *LOCAL.db2inst1.070718200352
3              1 select * from employee
db2bp          CLP work1.db2      *LOCAL.db2inst1.070718200352
4              1 select * from sales
...

```

Tenga en cuenta que es posible que vea algunos avisos de truncamiento (SQL0445).

Cuando el CLP ejecute un script, establecerá el registro especial CURRENT CLIENT\_APPLNAME en "CLP **nombre\_script**". Por lo tanto, a partir de la consulta anterior podrá determinar qué script ha enviado cada actividad.

## Paso 10: Restablecer para el siguiente ejercicio

Actualice la carga de trabajo SYSDEFAULTUSERWORKLOAD y la subclase de servicio SYSDEFAULTSUBCLASS para que no se recopile ningún dato de actividad ni estadísticas de actividades agregadas, inhabilite los supervisores de sucesos y borre las tablas de actividades y de estadísticas y llame a WLM\_COLLECT\_STATS() para restablecer las estadísticas.

```
ALTER WORKLOAD SYSDEFAULTUSERWORKLOAD COLLECT ACTIVITY DATA NONE

ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
COLLECT AGGREGATE ACTIVITY DATA NONE

SET EVENT MONITOR DB2ACTIVITIES STATE 0
```

```
SET EVENT MONITOR DB2STATISTICS STATE 0

DELETE FROM ACTIVITY_DB2ACTIVITIES
DELETE FROM ACTIVITY$TMT_DB2ACTIVITIES
DELETE FROM SCSTATS_DB2STATISTICS
DELETE FROM WLSTATS_DB2STATISTICS

CALL WLM_COLLECT_STATS()
```

---

## Ejercicio 2: Aislamiento de actividades mediante clases de servicio y cargas de trabajo

Este ejercicio muestra cómo crear clases de servicio y cómo enviar actividades a una clase de servicio mediante la utilización de una carga de trabajo. También muestra cómo utilizar varias funciones de supervisión de WLM para determinar la carga de trabajo con la que se correlacionan esas actividades y para obtener información acerca de las actividades que se ejecutan en una clase de servicio y bajo una carga de trabajo.

Tiempo estimado: 20-25 minutos

Las clases de servicio son el punto principal de control de recursos para las actividades de base de datos. También son útiles para la supervisión. Por ejemplo, puede recopilar estadísticas para actividades en una clase de servicio en particular para determinar si se cumplen los objetivos de rendimiento para esa clase de servicio. Por omisión, para cada base de datos se crean tres clases de servicio por omisión (SYSDEFAULTSYSTEMCLASS, SYSDEFAULTMAINTENANCECLASS y SYSDEFAULTUSERCLASS). Si no se ha creado ninguna clase de servicio definida por el usuario, las actividades de usuario se ejecutan bajo la clase de servicio de usuario por omisión (SYSDEFAULTUSERCLASS).

Una carga de trabajo es una entidad que agrupa una o más unidades de trabajo basándose en criterios tales como el ID de usuario del sistema, el ID de usuario de sesión, etc. Las cargas de trabajo proporcionan un medio de asignar trabajo a una clase de servicio para que el trabajo pueda gestionarse posteriormente. Para cada base de datos se crean una carga de trabajo de usuario por omisión (SYSDEFAULTUSERWORKLOAD) y una carga de trabajo de administración por omisión (SYSDEFAULTADMWORKLOAD). Si no se ha creado ninguna carga de trabajo definida por el usuario, todas las actividades de usuario se asocian a la carga de trabajo de usuario por omisión.

En este ejercicio se muestran cuatro funciones distintas:

- Cómo crear una clase de servicio.
- Cómo crear una carga de trabajo.
- Cómo examinar las estadísticas de carga de trabajo básicas.
- Cómo recopilar información de actividad para las actividades que se ejecutan bajo una carga de trabajo individual.

### **Paso 1: Examinar dónde se ejecutan las actividades que no tienen clases de servicio ni cargas de trabajo definidas por el usuario**

En primer lugar, examine dónde se ejecutan las actividades si no existe ninguna clase de servicio o carga de trabajo definida por el usuario. Todas las actividades de DB2 se asignan a una carga de trabajo y se ejecutan en una clase de servicio. Si no se ha creado ninguna clase de servicio definida por el usuario, las actividades

se ejecutan en la subclase por omisión (SYSDEFAULTSUBCLASS), bajo la clase de servicio de usuario por omisión (SYSDEFAULTUSERCLASS), y si no se ha creado ninguna carga de trabajo definida por el usuario, las actividades se ejecutan bajo la carga de trabajo de usuario por omisión (SYSDEFAULTUSERWORKLOAD).

Ejecute los scripts work1.db2 y work2.db2 y, a continuación, examine las estadísticas para SYSDEFAULTSUBCLASS de SYSDEFAULTUSERCLASS utilizando WLM\_GET\_SERVICE\_SUBCLASS\_STATS.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
CONNECT TO SAMPLE

SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 30) SUPERCLASS,
        VARCHAR( SERVICE_SUBCLASS_NAME, 30) SUBCLASS,
        COORD_ACT_COMPLETED_TOTAL
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS('', '', -1)) AS T
```

Verá salida similar a la siguiente:

SUPERCLASS D_TOTAL	SUBCLASS	COORD_ACT_COMPLETE
-----	-----	-----
SYSDEFAULTSYSTEMCLASS 0	SYSDEFAULTSUBCLASS	
SYSDEFAULTMAINTENANCECLASS 0	SYSDEFAULTSUBCLASS	
SYSDEFAULTUSERCLASS 75	SYSDEFAULTSUBCLASS	

3 registro(s) seleccionado(s).

Observe que todas las actividades se ejecutan en la superclase de servicio SYSDEFAULTUSERCLASS.

Información adicional: también existen otras dos clases de servicio, SYSDEFAULTSYSTEMCLASS y SYSDEFAULTMAINTENANCECLASS. Estas clases de servicio se utilizan para realizar tareas de mantenimiento interno y de nivel del sistema. Las actividades de usuario no se ejecutarán en estas clases de servicio. Puede que también observe recuentos de actividad distintos de cero en estas clases de servicio si el servidor de datos de DB2 ha emitido alguna actividad interna.

Utilice la función de tabla WLM\_GET\_WORKLOAD\_STATS para ver estadísticas de carga de trabajo con el fin de determinar a qué carga de trabajo se asocian las aplicaciones.

```
SELECT SUBSTR(WORKLOAD_NAME, 1, 22) AS WL_DEF_NAME,
        WLO_COMPLETED_TOTAL,
        CONCURRENT_WLO_ACT_TOP FROM
        TABLE(WLM_GET_WORKLOAD_STATS(CAST(NULL AS VARCHAR(128)), -2))
        AS WLSTATS
```

La salida tendrá un aspecto similar al que se muestra a continuación:

WL_DEF_NAME	WLO_COMPLETED_TOTAL	CONCURRENT_WLO_ACT_TOP
-----	-----	-----
SYSDEFAULTUSERWORKLOAD	3	5
SYSDEFAULTADMWORKLOAD	0	0

2 registro(s) seleccionado(s).

Observe que existe una aparición de carga de trabajo completada para los dos scripts (work1.db2 y work2.db2), así como una aparición de carga de trabajo para la

conexión utilizada para ejecutar el mandato anterior.

## Paso 2: Crear una clase de servicio y una carga de trabajo

Cree una clase de servicio y, a continuación, cree una carga de trabajo para que todas las actividades que se ejecuten desde el script work1.db2 se correlacionen con la clase de servicio recientemente creada. Cuando el CLP ejecute un script, el valor del registro especial CURRENT CLIENT\_APPLNAME se establecerá en "CLP nombre\_script".

```
CREATE SERVICE CLASS work1_sc

CREATE WORKLOAD work1_w1 CURRENT CLIENT_APPLNAME('CLP work1.db2')
    SERVICE CLASS work1_sc
```

Información adicional: es posible especificar varios atributos al crear una carga de trabajo o una clase de servicio. Por ejemplo, al crear una carga de trabajo, puede identificar la conexión basándose en el nombre de la aplicación, el usuario de la sesión, etc. Para obtener más información, consulte CREATE WORKLOAD en la documentación de CREATE SERVICE CLASS.

## Paso 3: Otorgar privilegio de uso para la carga de trabajo

Otorgue privilegio de uso para la carga de trabajo (requiere autorización ACCESSCTRL o SECADM).

```
GRANT USAGE ON WORKLOAD work1_w1 TO PUBLIC
```

Información adicional: una conexión puede asociarse a una carga de trabajo sólo si el usuario de la sesión dispone de privilegio USAGE para la carga de trabajo. Esto es necesario para impedir que los usuarios puedan cambiar los atributos de conexión de su aplicación si intentan ejecutar el trabajo en una clase de servicio de prioridad más alta. Algunos atributos de conexión pueden cambiarse mediante programación (utilizando la API sqleset, por ejemplo). En este ejercicio, sólo otorgaremos privilegio USAGE a PUBLIC. En un sistema real, posiblemente deseará aplicar criterios más restrictivos. Puesto que el ejemplo se ejecutará con DBADM, este paso puede pasarse por alto en conjunto.

## Paso 4: Restablecer estadísticas

Restablezca las estadísticas utilizando la función WLM\_COLLECT\_STATS, para borrar las estadísticas recopiladas.

```
CALL SYSPROC.WLM_COLLECT_STATS()
```

## Paso 5: Ejecutar varias actividades

Ejecute los scripts work1.db2 y work2.db2, ambos.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
```

## Paso 6: Ver estadísticas de carga de trabajo y de clase de servicio

Utilice la función de tabla WLM\_GET\_WORKLOAD\_STATS para ver estadísticas de carga de trabajo con el fin de determinar a qué carga de trabajo se asocian las aplicaciones.

```

CONNECT TO SAMPLE

SELECT SUBSTR(WORKLOAD_NAME, 1, 22) AS WL_DEF_NAME,
       WLO_COMPLETED_TOTAL,
       CONCURRENT_WLO_ACT_TOP
FROM TABLE(WLM_GET_WORKLOAD_STATS(CAST(NULL AS VARCHAR(128)), -2))
AS WLSTATS

```

La salida tendrá un aspecto similar al que se muestra a continuación:

WL_DEF_NAME	WLO_COMPLETED_TOTAL	CONCURRENT_WLO_ACT_TOP
WORK1_WL	1	5
SYSDEFAULTUSERWORKLOAD	1	5
SYSDEFAULTADMWORKLOAD	0	0

Observe que una aparición de carga de trabajo se ha completado en WORK1\_WL, que es el script work1.db2. Una aparición de carga de trabajo se ha completado en SYSDEFAULTUSERWORKLOAD, que es el script work2.db2.

Puede que vea una segunda aparición de carga de trabajo completada para SYSDEFAULTUSER WORKLOAD, que es la conexión que se ha utilizado para llamar al procedimiento WLM\_COLLECT\_STATS. WLM\_COLLECT\_STATS es un procedimiento asíncrono que podría completarse antes de que se recopilen realmente las estadísticas y, por lo tanto, podría incluirse.

También puede utilizar la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS para que se muestre bajo qué clase de servicio se ejecutan las actividades como resultado de haber creado la nueva carga de trabajo.

```

SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 30) SUPERCLASS,
       VARCHAR( SERVICE_SUBCLASS_NAME, 23) SUBCLASS,
       COORD_ACT_COMPLETED_TOTAL COORDACTCOMP
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS('','",-1)) AS T

```

Los resultados tendrán un aspecto similar al siguiente:

SUPERCLASS	SUBCLASS	COORDACTCOMP
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	37
WORK1_SC	SYSDEFAULTSUBCLASS	37

Observe que las actividades se han completado bajo WORK1\_SC debido a la correlación de la carga de trabajo de WORK1\_WL.

## Paso 7: Crear otra clase de servicio y otra carga de trabajo

Cree una segunda clase de servicio y, a continuación, cree una carga de trabajo para que todas las actividades que se ejecuten desde la aplicación work2.db2 se correlacionen con la clase de servicio recientemente creada. Asimismo, defina la carga de trabajo para que recopile datos de actividad. Para este ejemplo, sólo recopilaremos datos de actividad sin detalles ni valores adicionales.

```

CREATE SERVICE CLASS work2_sc

CREATE WORKLOAD work2_w1
       CURRENT_CLIENT_APPLNAME('CLP work2.db2')
       SERVICE CLASS work2_sc
       COLLECT ACTIVITY DATA

```

Información adicional: cuando se especifica la cláusula COLLECT ACTIVITY DATA para una carga de trabajo, la información relacionada con cualquier actividad que envíe una aparición de esa carga de trabajo se enviará al supervisor de sucesos ACTIVITIES activo cuando se haya completado la actividad. La cláusula COLLECT ACTIVITY DATA le permite especificar cuánta información ha de recopilarse aplicando una de las opciones siguientes:

- WITHOUT DETAILS: Recopila información de actividad, sin la sentencia ni el entorno de compilación (valor por omisión).
- WITH DETAILS: Recopila información de actividad, incluidos la sentencia y el entorno de compilación.
- WITH DETAILS AND VALUES: Recopila información de actividad, incluidos la sentencia y el entorno de compilación, y valores de datos de entrada.

## Paso 8: Habilitar el supervisor de sucesos de actividad

Habilite los supervisores de sucesos para actividades.

El supervisor de sucesos de actividad se ha creado en el Ejercicio 1.

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

## Paso 9: Restablecer estadísticas y ejecutar varias actividades

Utilice el procedimiento almacenado WLM\_COLLECT\_STATS para volver a restablecer las estadísticas y vuelva a ejecutar los scripts work1.db2 y work2.db2.

```
CALL SYSPROC.WLM_COLLECT_STATS()
```

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
```

## Paso 10: Ver estadísticas de carga de trabajo y de clase de servicio

Vuelva a utilizar la función de tabla WLM\_GET\_WORKLOAD\_STATS para determinar a qué carga de trabajo se asocian las aplicaciones.

```
CONNECT TO SAMPLE
```

```
SELECT SUBSTR(WORKLOAD_NAME, 1, 22) AS WL_DEF_NAME,
       WLO_COMPLETED_TOTAL,
       CONCURRENT_WLO_ACT_TOP
FROM TABLE(WLM_GET_WORKLOAD_STATS(CAST(NULL AS VARCHAR(128)), -2))
AS WLSTATS
```

La salida tendrá un aspecto similar al que se muestra a continuación:

WL_DEF_NAME	WLO_COMPLETED_TOTAL	CONCURRENT_WLO_ACT_TOP
WORK1_WL	1	5
WORK2_WL	1	5
SYSDEFAULTUSERWORKLOAD	0	0
SYSDEFAULTADMWORKLOAD	0	0

Observe que, esta vez, ambas definiciones de carga de trabajo tienen una ejecución de aparición de carga de trabajo, una para cada script.

Puede que vea o no una aparición de carga de trabajo completada para SYSDEFAULTUSERWORKLOAD, en función de si la aparición de carga de trabajo sobre la que la llamada al procedimiento WLM\_COLLECT\_STATS se ha enviado se ha cerrado antes de que se recopilaran las estadísticas.

Vuelva a utilizar WLM\_GET\_SERVICE\_SUBCLASS\_STATS para que se muestre bajo qué clase de servicio se ejecutan las actividades como resultado de haber creado la nueva carga de trabajo.

```
SELECT VARCHAR( SERVICE_SUPERCLASS_NAME, 30) SUPERCLASS,
        VARCHAR( SERVICE_SUBCLASS_NAME, 23) SUBCLASS,
        COORD_ACT_COMPLETED_TOTAL COORDACTCOMP
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS('','",-1)) AS T
```

Y los resultados tendrán un aspecto similar al siguiente:

SUPERCLASS	SUBCLASS	COORDACTCOMP
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1
WORK1_SC	SYSDEFAULTSUBCLASS	37
WORK2_SC	SYSDEFAULTSUBCLASS	37

Observe que, esta vez, la superclase de servicio work2\_sc tiene varias ejecuciones de actividades debido a la correlación de WORK2\_WL. La actividad bajo SYSDEFAULTUSERCLASS es la consulta ejecutada anteriormente en WLM\_GET\_WORKLOAD\_STATS.

## Paso 11: Ver los datos de actividad recopilados

Consulte la tabla de actividades para obtener información acerca de las actividades que se han ejecutado. Observe que sólo se han recopilado las actividades del script work2.db2 porque sólo la definición de carga de trabajo de work2\_wl tiene especificado el atributo COLLECT ACTIVITY DATA.

```
SELECT SUBSTR(WORKLOADNAME, 1, 20) WL_DEF_NAME,
        SUBSTR(APPL_NAME, 1, 20) APPL_NAME,
        SUBSTR(ACTIVITY_TYPE, 1, 10) ACT_TYPE
FROM SYSCAT.WORKLOADS, ACTIVITY_DB2ACTIVITIES
WHERE WORKLOADID = WORKLOAD_ID
```

Los resultados tienen un aspecto similar al siguiente:

WL_DEF_NAME	APPL_NAME	ACT_TYPE
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	WRITE_DML
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	OTHER
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	OTHER
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	LOAD
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL

WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	CALL
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	CALL
WORK2_WL	db2bp	CALL
WORK2_WL	db2bp	CALL
WORK2_WL	db2bp	DDL
:		
:		

## Paso 12: Asignar recursos a clases de servicio

Ahora que ha aislado las actividades que han emitido estos dos scripts en clases de servicio separadas, puede asignar recursos a las clases de servicio o supervisar las actividades que se ejecutan en esas clases de servicio. Algunos ejemplos: Si el trabajo que ha realizado el script `work2.db2` es más importante que el trabajo que ha realizado el script `work1.db2`, puede incrementar la prioridad de los agentes ejecutando la clase de servicio `WORK2_SC` mediante la utilización de una sentencia como la siguiente:

En entornos operativos UNIX (un valor negativo especifica una prioridad más alta):

```
ALTER SERVICE CLASS WORK2_SC AGENT PRIORITY -6
```

En entornos operativos Windows (un valor positivo especifica una prioridad más alta):

```
ALTER SERVICE CLASS WORK2_SC AGENT PRIORITY 6
```

Si deseara capturar detalles acerca de cada actividad individual que se ejecuta en la clase de servicio `WORK2_SC`, podría habilitar la recopilación de actividades para esa clase de servicio mediante lo siguiente:

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER WORK2_SC
  COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS
```

## Paso 13: Restablecer para el siguiente ejercicio

Actualice la carga de trabajo `work2_w1` para que no se recopile ningún dato de actividad, inhabilite el supervisor de sucesos y borre la tabla del supervisor de sucesos y llame a `WLM_COLLECT_STATS()` para restablecer las estadísticas.

```
ALTER WORKLOAD work2_w1
  COLLECT ACTIVITY DATA NONE

SET EVENT MONITOR DB2ACTIVITIES STATE 0

DELETE from ACTIVITY_DB2ACTIVITIES

CALL WLM_COLLECT_STATS()
```

---

## Ejercicio 3: Utilización de umbrales para controlar actividades malintencionadas y utilización del supervisor de violaciones de umbral

Este ejercicio muestra cómo puede utilizar umbrales para detectar un uso incorrecto de los recursos o el inicio de una sobrecarga del sistema estableciendo límites relacionados con el consumo de un recurso específico.

Tiempo estimado: 15-20 minutos

Si se infringe un umbral, puede activarse una acción especificada. Las acciones que reciben soporte son:

- STOP EXECUTION: Detener el proceso de la actividad que ha causado la infracción del umbral.
- CONTINUE: Continuar el proceso.
- Recopilar información acerca de la actividad que ha infringido el umbral. Esta acción puede especificarse junto con la acción CONTINUE o STOP EXECUTION.

Con independencia de si una actividad que infringe un umbral ha de detenerse o ha de permitirse que continúe la ejecución, se grabará un registro de la violación en un supervisor de sucesos THRESHOLD VIOLATIONS activo (dándose por supuesto que anteriormente se ha definido uno) cada vez que se infrinja un umbral. El registro contiene información como qué umbral se ha infringido, la hora a la que ha infringido y la acción de umbral.

Este ejercicio muestra cómo pueden utilizarse los umbrales para detectar o evitar la ejecución de actividades malintencionadas en el sistema y la utilización de recursos del sistema. Una actividad malintencionada es cualquier actividad que utiliza una cantidad de recursos inesperadamente elevada. Por ejemplo, una consulta que se ejecuta durante un período de tiempo anormalmente prolongado o que devuelve un conjunto de resultados inesperadamente grande.

### Paso 1: Crear un supervisor de sucesos de violación de umbral

Cree y habilite un supervisor de sucesos de grabación en tabla, que se utilizará para capturar la información de violación de umbral, y habilite el supervisor de sucesos de actividad que se ha creado en el Ejercicio 1.

```
CREATE EVENT MONITOR threvio FOR THRESHOLD VIOLATIONS WRITE TO TABLE
    THRESHOLDVIOLATIONS(IN userspace1),
    CONTROL(IN userspace1)
```

```
SET EVENT MONITOR threvio STATE 1
```

```
SET EVENT MONITOR db2activities STATE 1
```

### Paso 2: Crear una carga de trabajo

Cree una carga de trabajo para que todas las actividades ejecutadas desde el script workth.db2 se correlacionen con la clase de servicio work1\_sc.

La clase de servicio work1\_sc ya existe, pues se ha creado en el Ejercicio 2.

```
CREATE WORKLOAD workth_wl
    CURRENT CLIENT_APPLNAME('CLP workth.db2')
    SERVICE CLASS work1_sc
```

### Paso 3: Crear umbrales

Cree dos umbrales, uno de ellos (th\_estcost) será un umbral ESTIMATEDSQLCOST y el otro (th\_sqlrows) será un umbral SQLROWSRETURNED, y aplíquelos a la clase de servicio cuyas actividades desea controlar (en este caso, la clase de servicio work1\_sc).

El umbral th\_estcost especifica un límite superior (10.000 timerons) para el coste estimado por el optimizador (en timerons) de una actividad que se ejecuta en la clase de servicio work1\_sc. Si cualquier consulta con un coste estimado de más de 10.000 timerons intenta ejecutarse en la clase de servicio work1\_sc, este umbral se infringe y no se permite que la consulta se ejecute.

El umbral th\_sqlrows especifica que cualquier actividad que se ejecute en la clase de servicio work1\_sc puede devolver, como máximo, 30 filas del servidor de datos. Si cualquier consulta intenta devolver más de 30 filas, este umbral se infringirá, sólo se devolverán 30 filas al cliente y la consulta se detendrá. Asimismo, también se recopilarán datos acerca de la actividad que ha dado lugar a la violación de umbral.

En cualquier caso, cuando una actividad infringe un umbral, se graba un registro de violación de umbral en el supervisor de sucesos THRESHOLD VIOLATIONS tal como se ha definido en el Paso 1 y la ejecución de la actividad se detiene (debido a la acción STOP EXECUTION). La aplicación que ha enviado la actividad recibirá un error SQL4712N.

```
CREATE THRESHOLD th_estcost
  FOR SERVICE CLASS work1_sc ACTIVITIES
  ENFORCEMENT DATABASE
  WHEN ESTIMATEDSQLCOST > 10000
  STOP EXECUTION

CREATE THRESHOLD th_sqlrows
  FOR SERVICE CLASS work1_sc ACTIVITIES
  ENFORCEMENT DATABASE
  WHEN SQLROWSRETURNED > 30
  COLLECT ACTIVITY DATA WITH DETAILS AND VALUES
  STOP EXECUTION
```

Información adicional: un umbral puede ser predictivo o reactivo:

- Umbral predictivo: los límites de un umbral predictivo se comprueban antes de iniciarse la ejecución de la actividad. Para comprobar si un umbral predictivo podría infringirse, el servidor de datos obtiene estimaciones de uso del compilador de consultas. En este ejemplo, el umbral th\_estcost es un umbral predictivo.
- Umbral reactivo: los límites de un umbral reactivo se comprueban mientras se ejecuta una actividad. Para evaluar los límites de los umbrales reactivos se utilizan estimaciones de uso de tiempo de ejecución aproximado del recurso controlado. Las estimaciones de uso de tiempo de ejecución no se obtienen continuamente sino en puntos de comprobación predefinidos seleccionados durante la vida útil del trabajo cuyo seguimiento se realiza. En este ejemplo, el umbral th\_sqlrows es un umbral reactivo.

### Paso 4: Ejecutar varias actividades

Ejecute varias actividades; algunas de éstas han de infringir los límites superiores de umbral definidos en el paso anterior.

```
db2 -o -tvf workth.db2
```

Observe que las sentencias que infringen los umbrales definidos anteriormente no se ejecutan correctamente y generan un error SQL4712N/SQLSTATE 5U026.

### Paso 5: Ver el supervisor de sucesos de violación de umbral

El supervisor de sucesos THRESHOLD VIOLATIONS recopila información acerca de cada violación de umbral. Puede consultar la información de violación de umbral emitiendo sentencias de SQL normales para la tabla del supervisor de violaciones de umbral, tal como se muestra en el ejemplo siguiente:

```
CONNECT TO SAMPLE
SELECT APPL_ID,
       UOW_ID,
       ACTIVITY_ID,
       COORD_PARTITION_NUM AS COORDPART,
       THRESHOLD_PREDICATE,
       THRESHOLD_ACTION,
       TIME_OF_VIOLATION
FROM THRESHOLDVIOLATIONS_THREVIEW
ORDER BY THRESHOLD_ACTION, THRESHOLD_PREDICATE, TIME_OF_VIOLATION
```

La salida tendrá un aspecto similar al que se muestra a continuación:

APPL_ID	UOW_ID	ACTIVITY_ID	COORDPART	THRESHOLD_PREDICATE	THRESHOLD_ACTION	TIME_OF_VIOLATION
*LOCAL.DB2.070821150008	11	1	0	EstimatedSQLCost	Stop	2007-08-21-
11.00.11.000000						
*LOCAL.DB2.070821150008	10	1	0	SQLRowsReturned	Stop	2007-08-21-
11.00.10.000000						

2 registro(s) seleccionado(s).

### Paso 6: Ver información para la actividad que ha infringido el umbral

Se recopila información de actividad para cualquier actividad que infringe un umbral que se ha definido con una cláusula COLLECT. Visualice información detallada acerca de las actividades que han infringido un umbral mediante la utilización de la consulta siguiente:

```
SELECT VARCHAR(A.APPL_NAME, 15) AS APPL_NAME,
       VARCHAR(A.TPMON_CLIENT_APP, 20) AS CLIENT_APP_NAME,
       A.ACTIVITY_ID,
       A.ACTIVITY_TYPE,
       A.WORKLOAD_ID,
       T.THRESHOLD_PREDICATE,
       A.QUERY_CARD_ESTIMATE,
       T.THRESHOLD_MAXVALUE,
       T.TIME_OF_VIOLATION,
       VARCHAR(AS.STMT_TEXT, 100) AS STMT_TEXT
FROM THRESHOLDVIOLATIONS_THREVIEW AS T,
     ACTIVITY_DB2ACTIVITIES AS A,
     ACTIVITYSTMT_DB2ACTIVITIES AS AS
WHERE T.APPL_ID = A.APPL_ID AND
      T.UOW_ID = A.UOW_ID AND
```

```
T.ACTIVITY_ID = A.ACTIVITY_ID AND
A.APPL_ID = AS.APPL_ID AND
A.ACTIVITY_ID = AS.ACTIVITY_ID AND
A.UOW_ID = AS.UOW_ID
```

La salida tendrá un aspecto similar al que se muestra a continuación:

```
APPL_NAME          CLIENT_APP_NAME      ACTIVITY_ID          ACTIVITY_TYPE
                   WORKLOAD_ID THRESHOLD_PREDICATE
                   QUERY_CARD_ESTIMATE THRESHOLD_MAXVALUE
TIME_OF_VIOLATION  STMT_TEXT
-----
db2bp              CLP workth.db2          3 READ_DML
                   3 SQLRowsReturned
                   41
0 2007-08-31-09.01.16.000000 SELECT * FROM SALES
```

Observe que la actividad que ha infringido el umbral `th_estcost` (EstimatedSqlCost) no se muestra. La razón es que el umbral no ha especificado la cláusula `COLLECT ACTIVITY DATA`, por lo tanto, no se ha recopilado ningún dato de actividad para esa actividad.

### Paso 7: Restablecer para el siguiente ejercicio

Inhabilite los supervisores de sucesos que se han habilitado. Inhabilite también y descarte los umbrales `th_estcost` y `th_sqlrows` que se han creado.

```
SET EVENT MONITOR threvio STATE 0
SET EVENT MONITOR db2activities STATE 0
```

```
ALTER THRESHOLD th_estcost DISABLE
DROP THRESHOLD th_estcost
```

```
ALTER THRESHOLD th_sqlrows DISABLE
DROP THRESHOLD th_sqlrows
```

Asimismo, borre las tablas del supervisor de sucesos de actividades y la tabla de violaciones de umbral.

```
DELETE from ACTIVITY_DB2ACTIVITIES
DELETE from ACTIVITYSTMT_DB2ACTIVITIES
DELETE from THRESHOLDVIOLATIONS_THREVIEW
```

```
CALL WLM_COLLECT_STATS()
```

---

## Ejercicio 4: Diferenciación de actividades por tipo de actividad

Este ejercicio muestra cómo puede utilizarse un conjunto de acciones de trabajo para recopilar información acerca de todas las actividades de un tipo determinado, aplicar un umbral a todas las actividades de un tipo determinado y aislar actividades de un tipo determinado correlacionándolas con una subclase de servicio específica.

Tiempo estimado: 25-30 minutos

Los conjuntos de acciones de trabajo se utilizan para aplicar una acción a una actividad basándose en la acción que la actividad realiza en lugar de en quién la ha enviado (como se hace con las cargas de trabajo).

Las acciones pueden aplicarse a:

- Todas las actividades de base de datos de un tipo determinado (mediante un conjunto de acciones de trabajo de base de datos).
- Sólo las actividades de un tipo de terminado de una clase de servicio en particular (mediante un conjunto de acciones de trabajo de clase de servicio).

En este ejercicio se muestran ambos métodos:

Información adicional: pueden aplicarse otras acciones, como la recopilación de estadísticas para actividades de un tipo determinado, que no se explican en este ejercicio.

## Paso 1: Crear un conjunto de clases de trabajo

En primer lugar, cree un conjunto de clases de trabajo que contenga las clases de trabajo que representarán los tipos específicos de actividades que le interesan. Este conjunto de clases de trabajo se utilizará junto con los conjuntos de acciones de trabajo para realizar acciones en los tipos seleccionados de actividades. El ejemplo siguiente crea un conjunto de clases de trabajo que contiene clases de trabajo para todos los tipos posibles, pero si está interesado en sólo un único tipo de actividad, el conjunto de clases de trabajo debería crearse para que sólo contenga esa clase de trabajo.

```
CREATE WORK CLASS SET all_class_types
(WORK CLASS read_wc WORK TYPE READ,
 WORK CLASS write_wc WORK TYPE WRITE,
 WORK CLASS ddl_wc WORK TYPE DDL,
 WORK CLASS call_wc WORK TYPE CALL,
 WORK CLASS load_wc WORK TYPE LOAD,
 WORK CLASS all_wc WORK TYPE ALL POSITION LAST)
```

## Paso 2: Habilitar el supervisor de sucesos de actividad

Habilite el supervisor de sucesos para actividades que se ha creado en el Ejercicio 1.

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

## Paso 3: Crear un conjunto de acciones de trabajo de base de datos

Si desea realizar una acción en particular para todas las actividades de un tipo específico (como aplicar un umbral o recopilar información de actividad), utilice un conjunto de acciones de trabajo de base de datos.

Cree un conjunto de acciones de trabajo en el nivel de la base de datos que contenga acciones de trabajo para la clase de trabajo específica que representa el tipo de actividades que desea aislar. Para este ejemplo, deseamos recopilar datos de actividad para todas las actividades DDL, READ y LOAD que se ejecutan en el sistema y también deseamos detener la ejecución de cualquier actividad de lectura grande. Para este ejercicio, una actividad de lectura grande es cualquier sentencia de selección que tenga un coste estimado (en timerons) superior a 10.000.

```
CREATE WORK ACTION SET db_was FOR DATABASE
USING WORK CLASS SET all_class_types
(WORK ACTION collect_load_wa ON WORK CLASS load_wc
 COLLECT ACTIVITY DATA WITH DETAILS AND VALUES,
 WORK ACTION collect_ddl_wa ON WORK CLASS ddl_wc
 COLLECT ACTIVITY DATA WITH DETAILS AND VALUES,
 WORK ACTION collect_read_wa ON WORK CLASS read_wc
```

```

COLLECT ACTIVITY DATA WITH DETAILS AND VALUES,
WORK ACTION stop_large_read_wa on WORK CLASS read_wc
WHEN ESTIMATEDSQLCOST > 10000 STOP EXECUTION )

```

## Paso 4: Ejecutar actividades y ver las estadísticas del conjunto de acciones de trabajo

Ejecute los scripts work1.db2 y work3.db2.

```

db2 -o -tvf work1.db2
db2 -o -tvf work3.db2

```

Puede utilizar la función de tabla WLM\_GET\_WORK\_ACTION\_SET\_STATS para acceder a las estadísticas del conjunto de acciones de trabajo en memoria con el fin de obtener el número de veces que se han ejecutado tipos de actividades específicas. Observe que la ejecución de la siguiente consulta sólo muestra las clases de trabajo load\_wc, read\_wc y ddl\_wc, pues son las únicas clases de trabajo que tienen una acción de trabajo aplicable. El recuento de todas las demás actividades se realiza bajo el "\*":

```

CONNECT TO SAMPLE

SELECT SUBSTR(WORK_ACTION_SET_NAME, 1, 12) AS WORK_ACTION_SET_NAME,
       SUBSTR(WORK_CLASS_NAME, 1, 12) AS WORK_CLASS_NAME,
       LAST_RESET,
       SUBSTR(CHAR(ACT_TOTAL), 1, 10) AS TOTAL_ACTS
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS('', -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME

```

La salida tendrá un aspecto similar al siguiente:

WORK_ACTION_SET_NAME	WORK_CLASS_NAME	LAST_RESET	TOTAL_ACTS
DB_WAS	*	2007-08-15-19.02.47.305556	12
DB_WAS	DDL_WC	2007-08-15-19.02.47.305556	12
DB_WAS	LOAD_WC	2007-08-15-19.02.47.305556	1
DB_WAS	READ_WC	2007-08-15-19.02.47.305556	13

4 registro(s) seleccionado(s).

## Paso 5: Ver los datos de actividad recopilados

El supervisor de sucesos de actividad ha recopilado información acerca de cada actividad DDL, READ y LOAD individual debido a la especificación de la acción de trabajo COLLECT ACTIVITY DATA que se ha aplicado a las clases de trabajo ddl\_wc, read\_wc y load\_wc en el Paso 3. A continuación se muestran dos ejemplos de cómo podría desear que se visualizara esta información de actividad.

Para obtener información básica acerca de las actividades, puede consultar simplemente la tabla del supervisor de sucesos de actividad con una sentencia como la siguiente:

```

SELECT ACTIVITY_ID,
       SUBSTR(ACTIVITY_TYPE, 1, 8) AS ACTIVITY_TYPE,
       VARCHAR(APPL_ID, 30) AS APPL_ID,
       VARCHAR(APPL_NAME, 10) AS APPL_NAME
FROM ACTIVITY_DB2ACTIVITIES

```

La salida tendrá un aspecto similar al siguiente:

ACTIVITY_ID	ACTIVITY_TYPE	APPL_ID	APPL_NAME
1	READ_DML	*LOCAL.karenam.070815192410	db2bp
1	READ_DML	*LOCAL.karenam.070815192418	db2bp

```

2 READ_DML *LOCAL.karenam.070815192418 db2bp
3 READ_DML *LOCAL.karenam.070815192418 db2bp
4 READ_DML *LOCAL.karenam.070815192418 db2bp
1 DDL *LOCAL.karenam.070815192418 db2bp
2 DDL *LOCAL.karenam.070815192418 db2bp
3 DDL *LOCAL.karenam.070815192418 db2bp
2 READ_DML *LOCAL.karenam.070815192418 db2bp
1 READ_DML *LOCAL.karenam.070815192418 db2bp
2 READ_DML *LOCAL.karenam.070815192418 db2bp
3 READ_DML *LOCAL.karenam.070815192418 db2bp
4 READ_DML *LOCAL.karenam.070815192418 db2bp
6 LOAD *LOCAL.karenam.070815192418 db2bp
1 DDL *LOCAL.karenam.070815192418 db2bp
1 DDL *LOCAL.karenam.070815192418 db2bp
2 DDL *LOCAL.karenam.070815192418 db2bp
3 DDL *LOCAL.karenam.070815192418 db2bp
4 DDL *LOCAL.karenam.070815192418 db2bp
5 READ_DML *LOCAL.karenam.070815192418 db2bp
10 READ_DML *LOCAL.karenam.070815192418 db2bp
1 DDL *LOCAL.karenam.070815192418 db2bp
2 DDL *LOCAL.karenam.070815192418 db2bp
3 DDL *LOCAL.karenam.070815192418 db2bp
4 DDL *LOCAL.karenam.070815192418 db2bp
1 READ_DML *LOCAL.karenam.070815192426 db2bp

```

26 registro(s) seleccionado(s).

Para obtener información adicional acerca de cada actividad, como el texto de la actividad y bajo qué clase de servicio se ha ejecutado, puede realizar una consulta similar a la siguiente:

```

SELECT VARCHAR(A.APPL_NAME, 15) as APPL_NAME,
       VARCHAR(A.TPMON_CLIENT_APP, 20) AS CLIENT_APP_NAME,
       VARCHAR(A.APPL_ID, 30) as APPL_ID,
       VARCHAR(A.SERVICE_SUPERCLASS_NAME, 20) as SUPER_CLASS,
       VARCHAR(A.SERVICE_SUBCLASS_NAME, 20) as SUB_CLASS,
       SQLCODE,
       VARCHAR(S.STMT_TEXT, 300) AS STMT_TEXT
FROM ACTIVITY_DB2ACTIVITIES AS A, ACTIVITYSTMT_DB2ACTIVITIES AS S
WHERE A.APPL_ID = S.APPL_ID AND
      A.ACTIVITY_ID = S.ACTIVITY_ID AND
      A.UOW_ID = S.UOW_ID

```

La salida tendrá un aspecto similar al siguiente:

```

APPL_NAME      CLIENT_APP_NAME      APPL_ID
SUPER_CLASS
      SUB_CLASS              SQLCODE      STMT_TEXT
-----
---
-----
---
-----
---
-----
---
-----
---
-----
db2bp          CLP wasdbsc.db2     *LOCAL.karenam.070815192410
SYSDEFAULTUS
ERCLASS SYSDEFAULTSUBCLASS      0 SELECT DISTINCT CURRENT SQLID FROM
SYS
IBM.SYSTABLES

db2bp          CLP work1.db2       *LOCAL.karenam.070815192418
SYSDEFAULTUS

```

```

ERCLASS SYSDEFAULTSUBCLASS          0 values(current client_applname)
:
:
db2bp          CLP work1.db2          *LOCAL.karenam.070815192418
SYSDEFAULTUS
ERCLASS SYSDEFAULTSUBCLASS          0 drop procedure stp2

db2bp          CLP work3.db2          *LOCAL.karenam.070815192426
SYSDEFAULTUS
ERCLASS SYSDEFAULTSUBCLASS          -4712 select count(*) from syscat.tables,
sy
scat.tables, syscat.tables, syscat.tables, syscat.tables, syscat.tables
:
:

```

Observe que una de las actividades tiene el SQLCODE -4712. Esto indica que la ejecución de la actividad se ha detenido debido a una violación de umbral. El umbral definido para la acción de trabajo stop\_large\_read\_wa impedirá la ejecución de cualquier sentencia SELECT que tenga un coste estimado superior a 10.000.

Información adicional: las actividades de carga (sin incluir la carga desde un cursor) no tienen una entrada en la tabla del supervisor de sucesos de sentencias de actividad (tabla activitystmt\_db2activities) que explique por qué no existe ningún registro para la única actividad de carga que el script work1.db2 ejecuta en la salida que se muestra en la última consulta anterior. La razón es que las actividades de carga no son sentencias de SQL. Para las actividades de carga desde cursor, existe una entrada para la sentencia de cursor en la tabla del supervisor de sucesos de sentencias de actividad porque el propio cursor es una actividad por separado. Existe una entrada para todas las actividades de carga en la tabla del supervisor de sucesos de actividad (activity\_db2activities).

## Paso 6: Inhabilitar las acciones de trabajo

Antes de continuar con el conjunto de acciones de trabajo de clase de servicio, descarte el conjunto de acciones de trabajo de base de datos.

```
DROP WORK ACTION SET db_was
```

Información adicional: antes de descartar cualquier umbral de simultaneidad, primero ha de inhabilitarse ese umbral. En este caso, no existe ninguna acción de trabajo que represente un umbral de simultaneidad pero, si existiera, la única forma de inhabilitarlo sería inhabilitando la acción de trabajo. Un umbral de acción de trabajo no puede manipularse mediante sentencias THRESHOLD de SQL; sólo pueden manipularse mediante sentencias WORK ACTION SET de SQL. Antes de descartar el siguiente conjunto de acciones de trabajo, sólo han de inhabilitarse las acciones de trabajo que representan umbrales de simultaneidad. Para este ejercicio, puesto que no existe ninguna acción de trabajo que represente un umbral de simultaneidad, no es necesario inhabilitar ninguna de las acciones de trabajo antes de descartar el conjunto de acciones de trabajo.

Si desea aplicar una acción en particular, como un umbral, a todas las actividades de un tipo determinado que se ejecutan en una superclase de servicio, debe considerar la utilización de un conjunto de acciones de trabajo de clase de servicio. Puede crear una acción de trabajo de correlación para correlacionar tipos específicos de actividades con una subclase de servicio específica y, a continuación, aplicar un umbral a esa subclase de servicio. En los pasos siguientes se muestra

cómo podrían utilizarse los conjuntos de acciones de trabajo de clase de servicio.

## Paso 7: Crear una clase de servicio y crear una carga de trabajo

Cree una subclase de servicio bajo la superclase de servicio work1\_sc que se ha creado en el Ejercicio 2, Paso 2.

La superclase de servicio work1\_sc es la clase de servicio con la que se correlacionarán las actividades mediante las cargas de trabajo. La subclase de servicio work1\_sc\_read es la clase de servicio con la que se correlacionarán las actividades de lectura mediante la acción de trabajo.

```
CREATE SERVICE CLASS work1_sc_read UNDER work1_sc
```

Cree una carga de trabajo para que todas las actividades que el script work3.db2 envía se correlacionen con la superclase de servicio work1\_sc. Observe que las actividades de work1.db2 ya se correlacionan con work1\_sc desde uno de los ejercicios anteriores.

```
CREATE WORKLOAD work3_w1 CURRENT CLIENT_APPLNAME('CLP work3.db2')
  SERVICE CLASS work1_sc
```

## Paso 8: Crear un conjunto de acciones de trabajo de clase de servicio

Cree un conjunto de acciones de trabajo en el nivel de la clase de servicio que contenga acciones de trabajo que se apliquen a las clases de trabajo específicas que representan los tipos de actividades que desea aislar. Para este ejemplo, deseamos recopilar datos de actividad para todas las actividades DDL, READ y LOAD que se ejecutan bajo la clase de servicio work1\_sc y también deseamos correlacionar las actividades de lectura con una subclase de servicio por separado para poder gestionarlas diferenciándolas; en este caso, se aplicará un umbral a la subclase de servicio para detener la ejecución de cualquier sentencia SELECT grande.

```
CREATE WORK ACTION SET sc_was FOR SERVICE CLASS work1_sc
  USING WORK CLASS SET all_class_types (
    WORK ACTION collect_load_wa ON WORK CLASS load_wc
      COLLECT ACTIVITY DATA ON ALL MEMBERS WITH DETAILS AND VALUES,
    WORK ACTION collect_ddl_wa ON WORK CLASS ddl_wc
      COLLECT ACTIVITY DATA ON ALL MEMBERS WITH DETAILS AND VALUES,
    WORK ACTION collect_read_wa ON WORK CLASS read_wc
      COLLECT ACTIVITY DATA ON ALL MEMBERS WITH DETAILS AND VALUES,
    WORK ACTION map_read_wa ON WORK CLASS read_wc
      MAP ACTIVITY TO work1_sc_read)
```

## Paso 9: Crear un umbral de clase de servicio

Para obtener un efecto similar al de la acción de trabajo stop\_large\_read\_wa que impedía la ejecución de cualquier sentencia SELECT grande, cree un umbral ESTIMATEDSQLCOST y aplique éste a la subclase de servicio work1\_sc\_read.

```
CREATE THRESHOLD stop_large_activities FOR SERVICE CLASS work1_sc_read
  UNDER work1_sc
  ACTIVITIES ENFORCEMENT DATABASE
  WHEN ESTIMATEDSQLCOST >10000 STOP EXECUTION
```

## Paso 10: Borrar las tablas de actividades, restablecer las estadísticas y ejecutar actividades

Borre todas las tablas de actividades para poder empezar desde cero antes de volver a ejecutar el script. A continuación, llame al procedimiento almacenado `wlm_collect_stats()` para restablecer las estadísticas.

```
DELETE FROM activity_db2activities
DELETE FROM activitystmt_db2activities
DELETE FROM activityvals_db2activities
```

```
CALL wlm_collect_stats()
```

Ahora, ejecute los scripts `work1.db2` y `work3.db2` una vez.

```
db2 -o -tvf work1.db2
db2 -o -tvf work3.db2
```

Observe el error `SQL04712` para las actividades que han dado lugar a que se excediera el umbral.

## Paso 11: Ver las estadísticas del conjunto de acciones de trabajo

Utilice la función de tabla `WLM_GET_WORK_ACTION_SET_STATS` para acceder a las estadísticas del conjunto de acciones de trabajo en memoria con el fin de obtener el número de veces que se han ejecutado tipos de actividades específicas. Observe que la ejecución de la siguiente consulta sólo muestra las clases de trabajo `load_wc`, `ddl_wc` y `read_wc`, pues son las únicas tres clases de trabajo a las que se ha aplicado una acción de trabajo. El recuento de la finalización de todas las demás actividades se realiza bajo el "\*":

```
CONNECT TO SAMPLE

SELECT SUBSTR(WORK_ACTION_SET_NAME, 1, 12) AS WORK_ACTION_SET_NAME,
       SUBSTR(CHAR(MEMBER), 1, 4) AS MEMB,
       SUBSTR(WORK_CLASS_NAME, 1, 12) AS WORK_CLASS_NAME, LAST_RESET,
       SUBSTR(CHAR(ACT_TOTAL), 1, 10) AS TOTAL_ACTS
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS(' ', -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME, MEMB
```

Esta vez, la salida tendrá un aspecto similar al que se muestra a continuación:

WORK_ACTION_SET_NAME	MEMB	WORK_CLASS_NAME	LAST_RESET	TOTAL_ACTS
SC_WAS	0	*	2007-08-15-19.02.54.597999	12
SC_WAS	0	DDL_WC	2007-08-15-19.02.54.597999	12
SC_WAS	0	LOAD_WC	2007-08-15-19.02.54.597999	1
SC_WAS	0	READ_WC	2007-08-15-19.02.54.597999	12

4 registro(s) seleccionado(s).

## Paso 12: Ver los datos de actividad recopilados

Ahora, vuelva a consultar las tablas de actividades para obtener información acerca de las actividades individuales. Observe la subclase de servicio bajo la que se han ejecutado las actividades.

```
SELECT VARCHAR(A.APPL_NAME, 15) as APPL_NAME,
       VARCHAR(A.TPMON_CLIENT_APP, 20) AS CLIENT_APP_NAME,
       VARCHAR(A.APPL_ID, 30) as APPL_ID,
       VARCHAR(A.SERVICE_SUPERCLASS_NAME, 20) as SUPER_CLASS,
       VARCHAR(A.SERVICE_SUBCLASS_NAME, 20) as SUB_CLASS,
       SQLCODE,
       VARCHAR(S.STMT_TEXT, 300) AS STMT_TEXT
```

```

FROM ACTIVITY_DB2ACTIVITIES AS A, ACTIVITYSTMT_DB2ACTIVITIES AS S
WHERE A.APPL_ID = S.APPL_ID AND
      A.ACTIVITY_ID = S.ACTIVITY_ID AND
      A.UOW_ID = S.UOW_ID

```

La salida tendrá un aspecto similar al siguiente:

```

APPL_NAME      CLIENT_APP_NAME      APPL_ID      SUPER_CLASS
      SUB_CLASS      SQLCODE      STMT_TEXT
-----
db2bp          CLP work1.db2      *LOCAL.karenam.070815195555  WORK1_SC
      WORK1_SC_READ      0 values(current client_aplname)
db2bp          CLP work1.db2      *LOCAL.karenam.070815195555  WORK1_SC
      WORK1_SC_READ      0 select * from org
:
:
db2bp          CLP work1.db2      *LOCAL.karenam.070815195555  WORK1_SC
      SYSDEFAULTSUBCLASS      0 drop procedure stp2
db2bp          CLP work3.db2      *LOCAL.karenam.070815195600  WORK1_SC
      WORK1_SC_READ      -4712 select count(*) from syscat.tables,
scat.tables, syscat.tables, syscat.tables, syscat.tables, syscat.tables

```

Observe nuevamente que una de las actividades tiene un SQLCODE -4712, esta vez debido a que se ha infringido el umbral de clase de servicio stop\_large\_activities que se ha creado en el Paso 9 porque el coste estimado para esa sentencia de selección era demasiado grande. Observe también que todas las actividades de lectura se ejecutan bajo la subclase de servicio work1\_sc\_read.

### Paso 13: Restablecer para el siguiente ejercicio

Inhabilite el supervisor de sucesos, descarte el umbral de clase de servicio y descarte el conjunto de acciones de trabajo de clase de servicio.

```

SET EVENT MONITOR DB2ACTIVITIES STATE 0

DROP THRESHOLD STOP_LARGE_ACTIVITIES
ALTER WORK ACTION SET SC_WAS
ALTER WORK ACTION COLLECT_LOAD_WA DISABLE
ALTER WORK ACTION COLLECT_DDL_WA DISABLE
ALTER WORK ACTION COLLECT_READ_WA DISABLE
ALTER WORK ACTION MAP_READ_WA DISABLE;
DROP WORK ACTION SET SC_WAS

```

Borre todas las tablas de actividades, para poder empezar desde cero antes de volver a ejecutar el script.

```

DELETE FROM activity_db2activities
DELETE FROM activitystmt_db2activities
DELETE FROM activityvals_db2activities

```

Inhabilite todas las cargas de trabajo que se han creado para que todas las actividades se ejecuten bajo la carga de trabajo de usuario por omisión y se correlacionen con la superclase de servicio por omisión.

```

ALTER WORKLOAD work1_w1 DISABLE
ALTER WORKLOAD work2_w1 DISABLE
ALTER WORKLOAD work3_w1 DISABLE
ALTER WORKLOAD workth_w1 DISABLE

```

Llame al procedimiento almacenado `wlm_collect_stats()` para restablecer las estadísticas.

```
CALL WLM_COLLECT_STATS()
```

---

## Ejercicio 5: Utilización de histogramas para clases de servicio

Este ejercicio muestra cómo utilizar la opción `COLLECT AGGREGATE ACTIVITY DATA BASE` en una clase de servicio para producir histogramas de vida útil, tiempos de ejecución y tiempos en cola de las actividades de coordinador.

Tiempo estimado: 25-30 minutos

Estos tres histogramas son útiles para conocer algo más que el promedio de vida útil, el tiempo de ejecución o el tiempo en cola de las actividades que se han ejecutado en el sistema, puesto que pueden utilizarse para calcular desviaciones estándar y pueden revelar valores atípicos. Para obtener más información acerca de los histogramas, consulte “Histogramas en la gestión de carga de trabajo” en la página 278.

El acceso a los histogramas se realiza por medio del supervisor de sucesos de estadísticas. En este ejercicio vuelve a utilizarse el supervisor de sucesos de estadísticas que se ha creado en el Ejercicio 1, Paso 1.

Información adicional: el supervisor de sucesos de estadísticas es un supervisor de sucesos de grabación en tabla, y contiene grupos de datos lógicos. El primero es el grupo de datos lógicos de control, que todo supervisor de sucesos tiene, y luego existen los grupos de datos lógicos que son específicos del tipo de supervisor de sucesos de estadísticas. Los grupos de datos lógicos específicos son:

- `histogrambin` para información de histograma
- `qstats` para estadísticas de cola de umbral
- `scstats` para estadísticas de clase de servicio
- `wcstats` para estadísticas de clase de trabajo
- `wlstats` para estadísticas de carga de trabajo

### Paso 1: Crear vistas para visualizar las estadísticas de histograma

Cree varias vistas para facilitar la realización de consultas de la tabla `HISTOGRAMBIN_DB2STATISTICS`. En la primera vista se visualizarán todos los tipos de histogramas disponibles. En este ejercicio sólo se informará de los tres tipos básicos: vida útil, tiempo de ejecución y tiempo en cola.

```
CREATE VIEW HISTOGRAMTYPES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) AS HISTOGRAM_TYPE
  FROM HISTOGRAMBIN_DB2STATISTICS
```

Una segunda vista facilitará la tarea de averiguar para qué clases de servicio se recopilan histogramas. La tabla `HISTOGRAMBIN_DB2STATISTICS` identifica las clases de servicio para las que se recopilan estadísticas mediante la utilización del ID de clase de servicio. La unión de esta tabla a la tabla de catálogo `SERVICECLASSES` permite presentar la información de clase de servicio con el nombre de la superclase de servicio y el nombre de la subclase de servicio en lugar del ID de clase de servicio.

```
CREATE VIEW HISTOGRAMSERVICECLASSES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) AS HISTOGRAM_TYPE,
    SUBSTR(PARENTSERVICECLASSNAME,1,24) AS SERVICE_SUPERCLASS,
```

```

        SUBSTR(SERVICECLASSNAME,1,24) AS SERVICE_SUBCLASS
FROM HISTOGRAMBIN_DB2STATISTICS AS H,
    SYSCAT.SERVICECLASSES AS S
WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID

```

En la tercera vista se visualizarán todas las horas a las que se ha recopilado un histograma de un tipo determinado para una clase de servicio determinada. Como la vista `histogramserviceclasses`, también une la tabla `HISTOGRAMBIN_DB2STATISTICS` a la tabla de catálogo `SERVICECLASSES`. La diferencia es que incluye la columna `STATISTICS_TIMESTAMP` como una de las columnas de la vista.

```

CREATE VIEW HISTOGRAMTIMES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) AS HISTOGRAM_TYPE,
    SUBSTR(PARENTSERVICECLASSNAME,1,24) AS SERVICE_SUPERCLASS,
    SUBSTR(SERVICECLASSNAME,1,24) AS SERVICE_SUBCLASS,
    STATISTICS_TIMESTAMP AS TIMESTAMP
FROM HISTOGRAMBIN_DB2STATISTICS AS H,
    SYSCAT.SERVICECLASSES AS S
WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID

```

La cuarta y última vista se utilizará para mostrar los propios histogramas. También se mostrará algo que suele ser necesario cuando se utilizan histogramas, que es agregar éstos a lo largo del tiempo. Esta vista muestra el valor superior de cada intervalo y el número de actividades que se ha contado para cada intervalo. Para los tres histogramas de este ejercicio, el campo `BIN_TOP` mide el número de milisegundos de vida útil de la actividad, el tiempo de ejecución o el tiempo en cola. Cuando el valor de `BIN_TOP` es de, por ejemplo, 3.000 milisegundos y el valor de `BIN_TOP` del intervalo anterior es de 2.000 milisegundos y el valor de `NUMBER_IN_BIN` es diez para un histograma de vida útil, significa que diez actividades han tenido una vida útil comprendida entre 2 y 3 segundos.

```

CREATE VIEW HISTOGRAMS(HISTOGRAM_TYPE,
    SERVICE_SUPERCLASS,
    SERVICE_SUBCLASS,
    BIN_TOP,
    NUMBER_IN_BIN) AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) AS HISTOGRAM_TYPE,
    SUBSTR(PARENTSERVICECLASSNAME,1,24) AS SERVICE_SUPERCLASS,
    SUBSTR(SERVICECLASSNAME,1,24) AS SERVICE_SUBCLASS,
    TOP AS BIN_TOP,
    SUM(NUMBER_IN_BIN) AS NUMBER_IN_BIN
FROM HISTOGRAMBIN_DB2STATISTICS AS H,
    SYSCAT.SERVICECLASSES AS S
WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID
GROUP BY HISTOGRAM_TYPE, PARENTSERVICECLASSNAME, SERVICECLASSNAME, TOP

```

## Paso 2: Activar la recopilación de histogramas

Los histogramas de vida útil, tiempo en cola y tiempo de ejecución de actividad se recopilan para una subclase de servicio cuando se ha habilitado la opción de recopilación de datos base de actividades agregadas para la subclase. Habilite la recopilación de datos base de actividades agregadas para la subclase por omisión de la superclase de usuario por omisión mediante la utilización de la cláusula `COLLECT AGGREGATE ACTIVITY DATA`.

Observe que todas las actividades se ejecutarán en la clase de servicio de usuario por omisión porque todas las cargas de trabajo definidas por el usuario se han inhabilitado al final del ejercicio anterior.

```

ALTER SERVICE CLASS SYSDEFAULTSUBCLASS
  UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA BASE

```

### Paso 3: Activar el supervisor de sucesos de estadísticas

Active el supervisor de sucesos de estadísticas que se ha creado anteriormente para que éste pueda recibir los datos agregados siempre que se recopilen.

```
SET EVENT MONITOR DB2STATISTICS STATE 1
```

### Paso 4: Ejecutar actividades y enviar estadísticas al supervisor de sucesos de estadísticas

Ahora, pueden ejecutarse varias actividades. Después de haberse completado las actividades, se llama al procedimiento almacenado WLM\_COLLECT\_STATS para enviar estadísticas (incluidos los histogramas de vida útil, de tiempo de ejecución y de tiempo en cola de actividad correspondientes a la clase de servicio de usuario por omisión) al supervisor de sucesos de estadísticas activo. Estos histogramas contienen datos acerca de todas las actividades que se han ejecutado en la clase de servicio de usuario por omisión desde que se habilitaran las estadísticas de actividades agregadas. La llamada a este procedimiento almacenado también restablece las estadísticas. Para mostrar los cambios en la actividad de base de datos a lo largo del tiempo, se crean tres intervalos de recopilación. En el primer intervalo, ejecute dos scripts, work1.db2 y work2.db2 y, a continuación, recopile y restablezca las estadísticas.

```
db2 -o -tvf work1.db2  
db2 -o -tvf work2.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS()
```

En el segundo intervalo, sólo ejecute el script work1.db2 una vez y, a continuación, recopile y restablezca las estadísticas.

```
db2 -o -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS()
```

En el tercer intervalo, ejecute work1.db2 dos veces y ejecute el script work2.db2 una vez; a continuación, recopile y restablezca las estadísticas.

```
db2 -o -tvf work1.db2  
db2 -o -tvf work2.db2  
db2 -o -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS()
```

La recopilación periódica de datos de esta forma le permite observar cómo cambia el trabajo en el sistema a lo largo del tiempo.

Información adicional: la recopilación periódica de datos no ha de ser necesariamente una operación manual. Mediante la utilización del parámetro de configuración de base de datos WLM\_COLLECT\_INT puede establecer el intervalo en minutos transcurrido el cual habrá de producirse automáticamente la recopilación y el restablecimiento de las estadísticas.

## Paso 5: Consultar vistas para ver estadísticas

Ahora que se han recopilado estadísticas, las vistas anteriormente creadas pueden utilizarse para ver las estadísticas. La vista HISTOGRAMTYPES sólo devuelve los tipos de histogramas disponibles.

```
SELECT * FROM HISTOGRAMTYPES
```

```
HISTOGRAM_TYPE
-----
CoordActExecTime
CoordActLifetime
CoordActQueueTime
```

3 registro(s) seleccionado(s).

Puesto que la opción BASE se ha utilizado al alterar la clase de servicio, existen tres histogramas: vida útil, tiempo de ejecución y tiempo en cola. La vista HISTOGRAMSERVICECLASSES le permite ver las clases de servicio para las que se ha recopilado un histograma. El ejemplo siguiente limita la salida a sólo la salida del histograma CoordActLifetime. Puesto que la recopilación de actividades agregadas sólo se ha activado para la subclase por omisión de la clase de servicio de usuario por omisión, sólo se muestra esa clase al realizar la selección desde la vista HISTOGRAMSERVICECLASSES.

```
SELECT * FROM HISTOGRAMSERVICECLASSES
  WHERE HISTOGRAM_TYPE = 'CoordActLifetime'
  ORDER BY SERVICE_SUPERCLASS, SERVICE_SUBCLASS
```

```
HISTOGRAM_TYPE      SERVICE_SUPERCLASS  SERVICE_SUBCLASS
-----
-
CoordActLifetime    SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS
```

1 registro(s) seleccionado(s).

La vista HISTOGRAMTIMES muestra las horas a las que se han recopilado los histogramas. Puesto que el procedimiento WLM\_COLLECT\_STATS se ha ejecutado tres veces, existen tres indicaciones de fecha y hora para el histograma de vida útil que se muestra.

```
SELECT * FROM HISTOGRAMTIMES
  WHERE HISTOGRAM_TYPE = 'CoordActLifetime'
  AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
  AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
ORDER BY TIMESTAMP
```

```
HISTOGRAM_TYPE  SERVICE_SUPERCLASS  SERVICE_SUBCLASS  TIMESTAMP
-----
CoordActLifetime SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  2007-08-05-
20.44.51.519380
CoordActLifetime SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  2007-08-05-
21.04.27.131281
CoordActLifetime SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS  2007-08-05-
21.08.27.474168
```

3 registro(s) seleccionado(s).

La última vista, HISTOGRAMS, se destina a la visualización de los propios histogramas. A diferencia de la vista HISTOGRAMTIMES, que muestra cada intervalo de recopilación en su propia fila, esta vista agrega datos de histograma de los diversos intervalos para producir un único histograma de un tipo determinado para una clase de servicio determinada.

```

SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
  WHERE HISTOGRAM_TYPE = 'CoordActLifetime'
    AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
    AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
  ORDER BY BIN_TOP

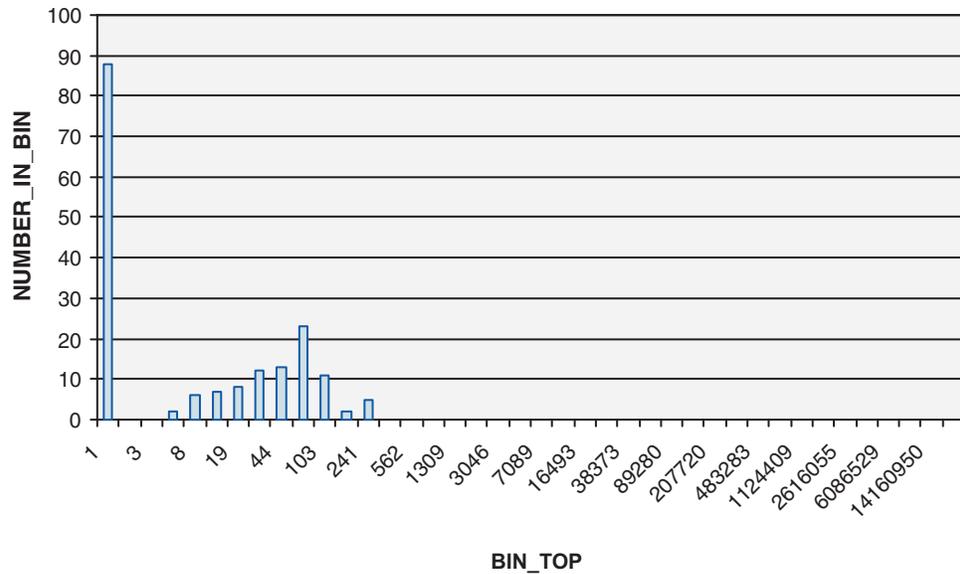
```

BIN_TOP	NUMBER_IN_BIN
-1	0
1	88
2	0
3	0
5	2
8	6
12	7
19	8
29	12
44	13
68	23
103	11
158	2
241	5
369	0
562	0
858	0
1309	0
1997	0
3046	0
4647	0
7089	0
10813	0
16493	0
25157	0
38373	0
58532	0
89280	0
136181	0
207720	0
316840	0
483283	0
737162	0
1124409	0
1715085	0
2616055	0
3990325	0
6086529	0
9283913	0
14160950	0
21600000	0

41 registro(s) seleccionado(s).

La salida de los histogramas puede utilizarse como entrada en una herramienta de gráficos para generar un gráfico. El diagrama siguiente muestra un gráfico que se ha creado mediante la utilización de una biblioteca gráfica de Ruby denominada Gruff Graphs.

Histograma de vida útil para SYSDEFAULTUSERCLASS (CoordActLifetime):



La ejecución de la consulta de esta vista produciría salida que no sería exactamente igual a la que se muestra en el gráfico de histograma de vida útil anterior porque la vida útil de las actividades depende del rendimiento del sistema. En la salida anterior, existen 41 intervalos y todos los intervalos más grandes están vacíos. En la parte superior existe un intervalo cuyo valor BIN\_TOP es -1. Este intervalo representa todas las actividades cuya vida útil era demasiado grande para poder representarse en el histograma. El hecho de ver un NUMBER\_OF\_BIN mayor que cero cuando BIN\_TOP es -1 indica que probablemente debe incrementar el valor de intervalo alto del histograma. En la salida anterior, NUMBER\_IN\_BIN es 0 y por lo tanto no es necesario realizar este cambio. Un número de actividades grande, 88 en este caso, es el recuento obtenido en el intervalo con el valor BIN\_TOP de 1. Es el intervalo más bajo y significa que 88 actividades han tenido una vida útil comprendida entre 0 y 1 milisegundo. Otra información que puede extraerse del histograma es que, puesto que el valor BIN\_TOP más grande para el que existe un valor NUMBER\_IN\_BIN correspondiente distinto de cero es 241, la vida útil más grande de cualquiera de las actividades de las cargas de trabajo recopiladas en este histograma estaba comprendida entre 158 y 241 milisegundos. La columna COORD\_ACT\_LIFETIME\_TOP de la tabla SCSTATS\_DB2STATISTICS proporciona una medición más precisa de la vida útil de la actividad que tiene la vida útil más grande.

La misma consulta puede repetirse con un tipo de histograma CoordActExecTime en lugar de CoordActLifetime. Se espera que el histograma de tiempo de ejecución sea similar, pero no idéntico, al histograma de vida útil. La razón por la que son distintos, incluso cuando no existe ninguna colocación en cola, es que el tiempo de ejecución no incluye el tiempo de inicialización ni el tiempo de inactividad de cursor, mientras que la vida útil sí los incluye.

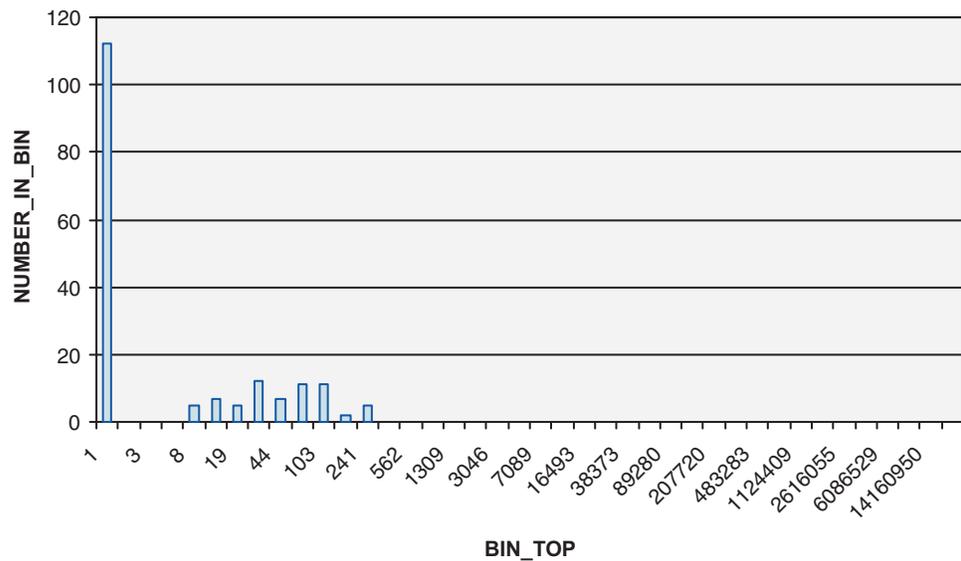
```
SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
  WHERE HISTOGRAM_TYPE = 'CoordActExecTime'
    AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
    AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
  ORDER BY BIN_TOP
```

BIN_TOP	NUMBER_IN_BIN
-1	0
1	112
2	0

3	0
5	0
8	5
12	7
19	5
29	12
44	7
68	11
103	11
158	2
241	5
369	0
562	0
858	0
1309	0
1997	0
3046	0
4647	0
7089	0
10813	0
16493	0
25157	0
38373	0
58532	0
89280	0
136181	0
207720	0
316840	0
483283	0
737162	0
1124409	0
1715085	0
2616055	0
3990325	0
6086529	0
9283913	0
14160950	0
21600000	0

41 registro(s) seleccionado(s).

Histograma de tiempo de ejecución para SYSDEFAULTUSERCLASS  
(CoordActExecTime):



Una vez más, se ha obtenido un número elevado de actividades en el primer intervalo, y el tiempo de ejecución más alto de cualquiera de las actividades es de, como máximo, 241 milisegundos.

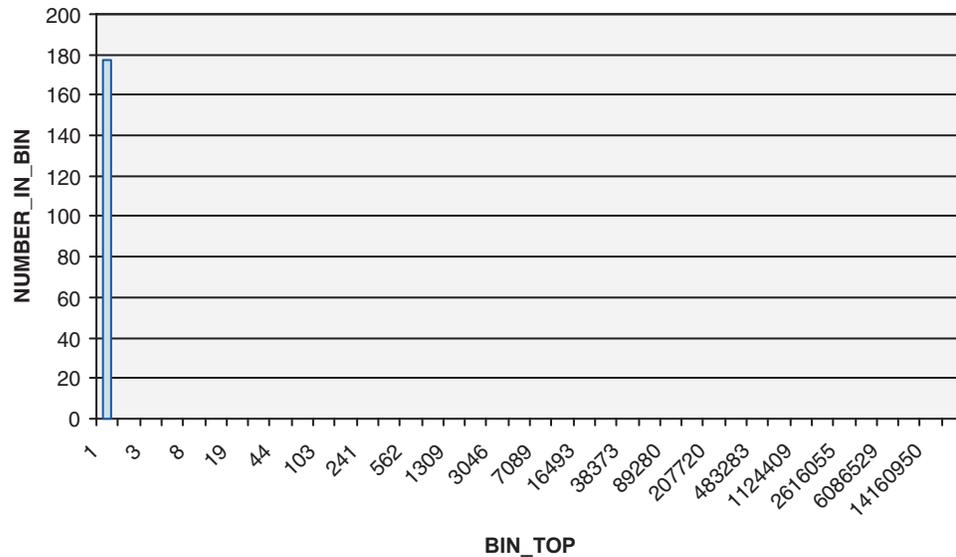
Para finalizar, la vista HISTOGRAMS se utilizará para ver el histograma CoordActQueueTime. Es el histograma más simple porque no existe ninguna colocación en cola, pues no se ha creado ningún umbral de colocación en cola en este ejercicio.

```
SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
WHERE HISTOGRAM_TYPE = 'CoordActQueueTime'
AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
ORDER BY BIN_TOP
```

BIN_TOP	NUMBER_IN_BIN
-1	0
1	177
2	0
3	0
5	0
8	0
12	0
19	0
29	0
44	0
68	0
103	0
158	0
241	0
369	0
562	0
858	0
1309	0
1997	0
3046	0
4647	0
7089	0
10813	0
16493	0
25157	0
38373	0
58532	0
89280	0
136181	0
207720	0
316840	0
483283	0
737162	0
1124409	0
1715085	0
2616055	0
3990325	0
6086529	0
9283913	0
14160950	0
21600000	0

41 registro(s) seleccionado(s).

Histograma de tiempo en cola para SYSDEFAULTUSERCLASS (CoordActQueueTime):



Todas las actividades se han incluido en el recuento del intervalo de 0 a 1 milisegundos porque cada una de ellas ha pasado 0 milisegundos en la cola.

Las últimas consultas han permitido ver la vida útil, los tiempos de ejecución y los tiempos en cola de las actividades distribuidos en intervalos ("bins"), pero agregados en varios intervalos. La consulta siguiente presenta la misma información pero desde una perspectiva distinta. Muestra promedios en lugar de histogramas y, en lugar de combinar intervalos, muestra cada intervalo individualmente. También informa del número de actividades que se han completado, que indica cuántas actividades se han completado en cada intervalo. Utiliza la tabla SCSTATS\_DB2STATISTICS en lugar de la tabla HISTOGRAMBIN\_DB2STATISTICS.

```
SELECT STATISTICS_TIMESTAMP,
       COORD_ACT_LIFETIME_AVG AS LIFETIMEAVG,
       COORD_ACT_EXEC_TIME_AVG AS EXECTIMEAVG,
       COORD_ACT_QUEUE_TIME_AVG AS QUEUETIMEAVG,
       COORD_ACT_COMPLETED_TOTAL AS COMPLETED_TOTAL
FROM SCSTATS_DB2STATISTICS
WHERE SERVICE_SUPERCLASS_NAME = 'SYSDEFAULTUSERCLASS'
      AND SERVICE_SUBCLASS_NAME = 'SYSDEFAULTSUBCLASS'
ORDER BY STATISTICS_TIMESTAMP
```

STATISTICS_TIMESTAMP COMPLETED_TOTAL	LIFETIMEAVG	EXECTIMEAVG	QUEUETIMEAVG
----- ----- 2007-08-07-14.07.44.511153 77	508	475	0
2007-08-07-14.07.46.537777 39	513	508	0
2007-08-07-14.07.51.882173 113	314	253	0

3 registro(s) seleccionado(s).

El resultado muestra que el promedio de vida útil es ligeramente superior al promedio de tiempo de ejecución para cada intervalo y que los tres sólo se han superado en medio segundo o menos. El promedio de tiempo en cola es cero, como se esperaba. Los recuentos del número de actividades completadas de cada intervalo es el esperado porque las cargas de trabajo 1 y 2 se han ejecutado en el primer intervalo, lo que ha generado la recopilación de 77 actividades, la carga de

trabajo 1 se ha ejecutado sola en el segundo intervalo, lo que ha generado 39 actividades, y la carga 1 se ha ejecutado dos veces y la carga de trabajo 2 se ha ejecutado una vez en el tercer intervalo, lo que ha generado 113 actividades.

### **Paso 6: Restablecer para el siguiente ejercicio**

El último paso consiste en desactivar la recopilación de actividades agregadas en la clase de servicio de usuario por omisión y en descartar las vistas y suprimir la información de las tablas de estadísticas.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS  
  UNDER SYSDEFAULTUSERCLASS  
  COLLECT AGGREGATE ACTIVITY DATA NONE
```

```
DROP VIEW histograms  
DROP VIEW histogramtimes  
DROP VIEW histogramserviceclasses  
DROP VIEW histogramtypes
```

```
SET EVENT MONITOR DB2STATISTICS STATE 0
```

```
DELETE FROM HISTOGRAMBIN_DB2STATISTICS  
DELETE FROM SCSTATS_DB2STATISTICS
```

---

## **Ejercicio 6: Investigación de retardos con funciones de tabla de WLM**

Este ejercicio muestra cómo puede determinar la causa de la ralentización de una aplicación con los recursos de supervisión del WLM de DB2.

Tiempo estimado: 10-15 minutos

Los recursos de supervisión del WLM de DB2 proporcionan información y estadísticas para el trabajo en una base de datos. Después de haber identificado la causa de una ralentización, puede solucionar la situación.

### **Paso 1: Ejecutar actividades**

En este ejercicio se utilizan dos aplicaciones, app1.db2 y app2.db2. Ambas aplicaciones realizan operaciones de DML en la base de datos SAMPLE. Ejecute el script app1.db2 en una ventana e inmediatamente después ejecute el script app2.db2 en una segunda ventana.

```
db2 -tvf app1.db2  
db2 -tvf app2.db2
```

### **Paso 2: Ver apariciones de carga de trabajo activa simultáneamente**

Ahora, el script app2.db2 debería colgarse. Desde una tercera ventana, emita la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES para conocer los estados de todas las aplicaciones que se ejecutan en la base de datos. Para este ejemplo, piense que una aparición de carga de trabajo es lo mismo que una aplicación. Esta función de tabla muestra información para todas las apariciones de carga de trabajo en una clase de servicio. Puesto que deseamos ver todas las apariciones de la carga de trabajo en la base de datos, utilizamos caracteres comodín, representados mediante \*, para los parámetros de entrada *service\_superclass\_name* y *service\_subclass\_name*.

```
CONNECT TO SAMPLE
```

```
SELECT INTEGER(APPLICATION_HANDLE) APPL_HANDLE,
```

```

VARCHAR(CLIENT_APPLNAME, 15) AS APPL_NAME,
VARCHAR(SYSTEM_AUTH_ID, 20) AS USER_ID
FROM TABLE
(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES('', '', -2))

```

La salida tendrá un aspecto similar al que se muestra a continuación:

```

APPL_HANDLE APPL_NAME      USER_ID
-----
          12 CLP app1.db2   DB2USR1
          17 CLP app2.db2   DB2USR1
          18 -              DB2USR1
          19 -              DB2USR1

```

4 registro(s) seleccionado(s).

A partir de la salida, podemos determinar que el descriptor de contexto de aplicación para app2.db2 es 17.

### Paso 3: Buscar el agente para la aplicación

Para averiguar qué hacen los agentes para app2.db2, utilice la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS. Esta función de tabla muestra información acerca de los agentes que trabajan en una clase de servicio. Puesto que deseamos ver los agentes que trabajan para el descriptor de contexto de aplicación 17, especificamos esto en el parámetro de entrada application\_handle. Para este ejemplo, no nos interesan los agentes de una clase de servicio en particular, por lo que especificamos caracteres comodín para los parámetros de entrada service\_superclass\_name y service\_subclass\_name.

```

SELECT INTEGER(APPLICATION_HANDLE) AS APPL_HANDLE,
       UOW_ID, ACTIVITY_ID,
       VARCHAR(AGENT_TYPE, 15) AS AGENT_TYPE,
       VARCHAR(AGENT_STATE, 10) AS AGENT_STATE,
       VARCHAR(EVENT_TYPE, 10) AS EVENT_TYPE,
       VARCHAR(EVENT_OBJECT, 10) AS EVENT_OBJ,
       VARCHAR(EVENT_STATE, 10) AS EVENT_STATE
FROM TABLE
(WLM_GET_SERVICE_CLASS_AGENTS('', '', 17, -2))

```

La salida tendrá un aspecto similar a:

```

APPL_HANDLE UOW_ID      ACTIVITY_ID AGENT_TYPE      AGENT_STATE EVENT_TYPE
EVENT_OBJ   EVENT_STATE
-----
          17          1          2 COORDINATOR    ACTIVE      ACQUIRE
LOCK        IDLE

```

1 registro(s) seleccionado(s).

A partir de la salida, vemos que el agente coordinador para la aplicación 17 está desocupado y a la espera de adquirir un bloqueo. Esta es la razón por la que app2.db2 aparece como colgada.

### Paso 4: Buscar la aplicación con el problema y solucionar el problema

Ahora que sabemos la razón por la que se ha colgado la aplicación, podemos solucionar la situación. Sabemos que la aplicación está a la espera de un bloqueo. Para averiguar qué bloque espera la aplicación y qué aplicación retiene el bloqueo, podemos utilizar la herramienta db2pd. En primer lugar, debemos averiguar el

número de transacción actual de la aplicación que se ha colgado: emita db2pd -transactions para el descriptor de contexto de aplicación 17.

```
db2pd -db sample -transactions app=17
```

La salida tendrá un aspecto similar al que se muestra a continuación:

Address	AppHandl [nod-index]	TranHdl	Locks	State	
Tflag	Tflag2	Firstlsn	Lastlsn	LogSpace	
SpaceReserved	TID	AxRegCnt	GXID		
0x07000000302A7080	17	[000-00017]	7	5	READ
0x00000000	0x00000000	0x000000000000	0x000000000000	0	
0	0x00000000AC3	1	0		

A partir de la salida, podemos determinar que la aplicación 17 tiene el descriptor de contexto de transacción 7. Ahora podemos buscar qué bloqueos espera esta transacción emitiendo el mandato db2pd -locks para el descriptor de contexto de transacción 7.

```
db2pd -db sample -locks 7 wait
```

La salida tendrá un aspecto similar al que se muestra a continuación:

Address	TranHdl	Lockname	Type	Mode	Sts
Owner	Dur	HoldCount	Att	ReleaseFlg	
0x07000000304013F0	7	0002001000000000640002D52	Row	.NS	W
2	1	0	0x00	0x00000002	

La salida muestra que la aplicación espera un bloqueo de fila. El propietario del bloqueo tiene el descriptor de contexto de transacción 2. Esta transacción está reteniendo el bloqueo y da lugar a que la aplicación se cuelgue. El paso final es determinar el descriptor de contexto de aplicación correspondiente para el descriptor de contexto de transacción 2. Emita el mandato db2pd -transactions para el descriptor de contexto de transacción 2.

```
db2pd -db sample -transactions 2
```

La salida tendrá un aspecto similar al que se muestra a continuación:

Address	AppHandl [nod-index]	TranHdl	Locks	State	
Tflag	Tflag2	Firstlsn	Lastlsn	LogSpace	
SpaceReserved	TID	AxRegCnt	GXID		
0x07000000302A2080	12	[000-00012]	2	6	WRITE
0x00000000	0x00000000	0x000002EE000C	0x000002EE005E	232	
396	0x00000000ABB	1	0		

A partir de la salida, vemos que el descriptor de contexto de transacción 2 corresponde al descriptor de contexto de aplicación 12. Al volver a consultar los resultados de la función de tabla WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES, puede ver que la aplicación 12 hace referencia a app1.db2. Esta aplicación retiene un bloqueo de fila que app2.db2 requiere. Para que app2.db2 pueda continuar, puede confirmar, retrotraer o terminar la unidad de trabajo o el proceso desde la ventana en la que se ejecuta app1.db2. De otro modo, también puede forzar la desactivación de app1.db2 emitiendo FORCE APPLICATION en el descriptor de contexto de aplicación 12.

```
db2 force application (12)
```

Información adicional: otra forma de diagnosticar aplicaciones que se han colgado debido a una contención de bloqueo consiste en utilizar la función de tabla SNAPSHOT\_LOCKWAIT del supervisor. Esta función de tabla proporciona información acerca de los poseedores de bloqueos y de los que esperan bloqueos.

Para utilizar esta función de tabla, debe activarse el parámetro de configuración de conmutador de supervisor DFT\_MON\_LOCK antes de iniciarse la base de datos. Este conmutador afecta a todas las bases de datos de una instancia.

---

## Ejercicio 7: Cancelación de una actividad en curso

Este ejercicio muestra cómo cancelar una actividad que actualmente está activa mediante la utilización del procedimiento WLM\_CANCEL\_ACTIVITY.

Tiempo estimado: 5-10 minutos

### Paso 1: Emitir una consulta de ejecución larga

Desde una ventana del CLP, ejecute el siguiente script, que emite una consulta de ejecución larga:

```
db2 -tvf longquery.db2
```

### Paso 2: obtener el descriptor de contexto de aplicación

Desde otra ventana del CLP, llame a WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para obtener el descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad de la actividad de cursor.

```
SELECT T.APPLICATION_HANDLE, T.UOW_ID, T.ACTIVITY_ID, T.ACTIVITY_TYPE
FROM SYSIBMADM.APPLICATIONS A,
     TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES
            (CAST(NULL AS BIGINT), -2)) T
WHERE (A.AGENT_ID = T.APPLICATION_HANDLE) AND
      (A.COORD_MEMBER = T.COORD_MEMBER) AND
      (A.MEMBER = T.MEMBER) AND
      (T.MEMBER = T.COORD_MEMBER) AND
      (A.TPMON_CLIENT_APP = 'CLP longquery.db2')
```

Mediante la unión de los resultados de la función de tabla a la vista administrativa APPLICATIONS, podemos determinar qué actividad de cursor se ejecuta desde longquery.db2. La salida tendría un aspecto similar al que se muestra a continuación:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	ACTIVITY_TYPE
-----			
----	267	1	1 READ_DML

1 registro(s) seleccionado(s).

### Paso 3: Cancelar la actividad

Desde la misma ventana del CLP, llame al procedimiento almacenado WLM\_CANCEL\_ACTIVITY para cancelar la actividad de cursor obtenida anteriormente, mediante la utilización del descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad obtenidos en el paso anterior:

```
CONNECT TO SAMPLE

CALL WLM_CANCEL_ACTIVITY (267, 1, 1)

CONNECT RESET
```

Tenga en cuenta que, en su caso, el descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad serán distintos.

En la primera ventana del CLP, verá que la consulta de ejecución larga que `longquery.db2` ha emitido devolverá la salida siguiente.  
SQL4725N Se ha cancelado la actividad. SQLSTATE=57014

---

## Ejercicio 8: Descubrimiento de qué tipos de actividades se ejecutan en el sistema

Este ejercicio muestra cómo puede utilizar las funciones de tabla de la gestión de carga de trabajo de DB2 y los conjuntos de acciones de trabajo para descubrir qué tipos de actividades se ejecutan en el sistema.

Tiempo estimado: 15-20 minutos

Puede que desee saber el número de actividades grandes o de programas de utilidad de carga que se ejecutan simultáneamente en el sistema, por ejemplo. El hecho de entender los tipos de trabajo que se ejecutan en el sistema es importante, pues distintos tipos de trabajo tendrán requisitos de recursos distintos y un impacto en el sistema distinto.

### Paso 1: Determinar el número de actividades de cada tipo que se ejecutan en el sistema

Antes de empezar, puede que desee que se muestre el número de actividades de un tipo determinado que están ejecutándose simultáneamente mediante la utilización de la función de tabla

`WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES`:

`CONNECT TO SAMPLE`

```
SELECT ACTIVITY_TYPE,  
       COUNT(*) AS NUMBER_RUNNING  
FROM TABLE (  
  WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(CAST(NULL AS BIGINT), -2)) AS T  
GROUP BY ACTIVITY_TYPE
```

La salida de esta consulta tendrá un aspecto similar al que se muestra a continuación:

ACTIVITY_TYPE	NUMBER_RUNNING
-----	-----
READ_DML	1

Para obtener información acerca de los distintos tipos de actividades que se han ejecutado en el sistema a lo largo de un período de tiempo determinado, puede utilizar conjuntos de clases de trabajo y acciones de trabajo.

### Paso 2: Crear un conjunto de acciones de trabajo de base de datos con acciones de trabajo de recuento de actividad

Para obtener el recuento del número de veces que una actividad de un tipo específico se ha ejecutado a lo largo de un período de tiempo, debe crearse un conjunto de acciones de trabajo. En este ejemplo, puesto que nos interesan las actividades que se ejecutan en todo el sistema, el conjunto de acciones de trabajo se creará en el nivel de la base de datos y se asociará al conjunto de clases de trabajo `all_class_types` que se ha creado en el Ejercicio 4, Paso 1. Este conjunto de clases de trabajo contiene clases de trabajo para todos los tipos de actividades reconocidas. Si sólo nos interesaran las actividades que se ejecutan en una clase de servicio específica, crearíamos un conjunto de acciones de trabajo en el nivel de la

clase de servicio. Para este ejemplo, también nos interesa la información para todos los tipos de actividades con el fin de que el conjunto de acciones de trabajo contenga una acción de trabajo COUNT ACTIVITY para cada clase de trabajo del conjunto de clases de trabajo all\_class\_types.

```
CREATE WORK ACTION SET work1_was FOR DATABASE
  USING WORK CLASS SET all_class_types
  (WORK ACTION count_read_wa ON WORK CLASS read_wc COUNT ACTIVITY,
   WORK ACTION count_write_wa ON WORK CLASS write_wc COUNT ACTIVITY,
   WORK ACTION count_ddl_wa ON WORK CLASS ddl_wc COUNT ACTIVITY,
   WORK ACTION count_call_wa ON WORK CLASS call_wc COUNT ACTIVITY,
   WORK ACTION count_load_wa ON WORK CLASS load_wc COUNT ACTIVITY,
   WORK ACTION count_all_wa ON WORK CLASS all_wc COUNT ACTIVITY)
```

Información adicional: siempre que se ha aplicado una o más acciones de trabajo a una actividad que corresponde a una clase de trabajo, el contador para la clase de trabajo se incrementa en uno. La acción de trabajo COUNT ACTIVITY proporciona una manera eficiente de garantizar que se actualiza el contador. Si no desea realizar ninguna otra acción en una actividad distinta de contar el número de actividades de ese tipo que se han ejecutado, la acción de trabajo COUNT ACTIVITY es el mejor enfoque.

### Paso 3: Ejecutar varias actividades

Ejecute una vez el script work1.db2.

```
db2 -tvf work1.db2
```

### Paso 4: Ver las estadísticas del conjunto de acciones de trabajo

Puede utilizar la función de tabla WLM\_GET\_WORK\_ACTION\_SET\_STATS para acceder a las estadísticas del conjunto de acciones de trabajo en memoria con el fin de obtener el número de veces que se han ejecutado tipos de actividades específicas. Por ejemplo, la cola siguiente le indicará el número de actividades que se han asignado a cada una de las clases de trabajo del conjunto de clases de trabajo al que se asocia una acción de trabajo:

```
CONNECT TO SAMPLE

SELECT SUBSTR(WORK_ACTION_SET_NAME, 1, 12) AS WORK_ACTION_SET_NAME,
       SUBSTR(WORK_CLASS_NAME, 1, 12) AS WORK_CLASS_NAME,
       LAST_RESET,
       SUBSTR(CHAR(ACT_TOTAL), 1, 12) AS TOTAL_ACTS
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS(' ', -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME, PART
```

Información adicional: el espacio en blanco que se incluye con la sentencia significa que el argumento no ha de restringir el resultado (en este ejemplo, deseamos obtener información para todos los conjuntos de acciones de trabajo). El valor del último argumento, member, es el carácter comodín -2, que significa que han de devolverse datos de todos los miembros de base de datos.

La salida de esta consulta tendrá un aspecto similar al siguiente, donde "\*" representa todas las actividades a las que no corresponde ninguna de las clases de trabajo definidas o a las que corresponden clases de trabajo que no tienen ninguna acción de trabajo.

```
WORK_ACTION_SET_NAME WORK_CLASS_NAME LAST_RESET
TOTAL_ACTS
-----
-
WORK1_WAS          *                2007-08-14-13.55.30.725886 0
```

WORK1_WAS	ALL_WC	2007-08-14-13.55.30.725886	2
WORK1_WAS	CALL_WC	2007-08-14-13.55.30.725886	4
WORK1_WAS	DDL_WC	2007-08-14-13.55.30.725886	12
WORK1_WAS	LOAD_WC	2007-08-14-13.55.30.725886	1
WORK1_WAS	READ_WC	2007-08-14-13.55.30.725886	12
WORK1_WAS	WRITE_WC	2007-08-14-13.55.30.725886	6

7 registro(s) seleccionado(s).

## Paso 5: Diferenciar actividades por otra información adicional además del tipo y de otros atributos

Puede separar las actividades por algo más que sólo sus tipos. Por ejemplo, puede que desee saber cuántas consultas grandes están ejecutándose.

Altere el conjunto de clases de trabajo para añadir una nueva clase de trabajo de lectura que representará las consultas grandes. Para este ejemplo, una consulta grande es cualquier consulta que tenga una cardinalidad superior a 40.

```
ALTER WORK CLASS SET all_class_types
  ADD WORK CLASS large_wc WORK TYPE READ FOR CARDINALITY FROM 41 POSITION AT 1
```

Información adicional: observe que hemos colocado esta clase de trabajo en la posición 1. Si no se hubiera especificado la cláusula POSITION AT, la clase de trabajo se ubicaría al final del conjunto de clases de trabajo. Al determinar a qué clase de trabajo pertenece una actividad, las clases de trabajo se comprueban en el orden en el que se han colocado, y la primera clase de trabajo cuyos atributos coinciden con los atributos de actividad es la clase a la que se asigna la actividad. En este caso, si large\_wc se hubiera ubicado al final de la lista, las actividades grandes se habrían asignado a read\_wc porque se ha colocado delante de large\_wc.

Altere el conjunto de acciones de trabajo para añadir una nueva acción de trabajo COUNT ACTIVITY y aplique ésta a la nueva clase de trabajo.

```
ALTER WORK ACTION SET work1_was
  ADD WORK ACTION count_large_reads ON WORK CLASS large_wc COUNT ACTIVITY
```

## Paso 6: Restablecer estadísticas y ejecutar varias actividades

Llame al procedimiento almacenado WLM\_COLLECT\_STATS para restablecer las estadísticas que se almacenan en memoria para empezar desde cero y, cuando opte por consultar esa información estadística de gestión de carga de trabajo que se almacena en memoria, ésta contendrá información para las actividades que se han ejecutado a partir de este punto.

```
CALL WLM_COLLECT_STATS()
```

Ejecute una vez el script work1.db2.

```
db2 -tvf work1.db2
```

## Paso 7: Ver las estadísticas del conjunto de acciones de trabajo

Vuelva a utilizar la función de tabla WLM\_GET\_WORK\_ACTION\_SET\_STATS para acceder a las estadísticas del conjunto de acciones de trabajo en memoria con el fin de obtener el número de veces que se han ejecutado tipos de actividades específicas.

```
CONNECT TO SAMPLE
```

```
SELECT SUBSTR(WORK_ACTION_SET_NAME, 1, 12) AS WORK_ACTION_SET_NAME,
  SUBSTR(CHAR(MEMBER), 1, 4) AS MEMB,
```

```

SUBSTR(WORK_CLASS_NAME, 1, 12) AS WORK_CLASS_NAME,
LAST_RESET,
SUBSTR(CHAR(ACT_TOTAL), 1, 12) AS TOTAL_ACTS
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS(' ', -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME, MEMB

```

La salida tendrá un aspecto similar al que se muestra a continuación:

```

WORK_ACTION_SET_NAME MEMB WORK_CLASS_NAME LAST_RESET
TOTAL_ACTS
-----
-----
-----
WORK1_WAS          0      *          2007-08-14-13.55.35.650685 0
WORK1_WAS          0    ALL_WC          2007-08-14-13.55.35.650685 2
WORK1_WAS          0    CALL_WC          2007-08-14-13.55.35.650685 4
WORK1_WAS          0    DDL_WC          2007-08-14-13.55.35.650685 12
WORK1_WAS          0   LARGE_WC          2007-08-14-13.55.35.650685 4
WORK1_WAS          0    LOAD_WC          2007-08-14-13.55.35.650685 1
WORK1_WAS          0    READ_WC          2007-08-14-13.55.35.650685 8
WORK1_WAS          0    WRITE_WC         2007-08-14-13.55.35.650685 6

```

8 registro(s) seleccionado(s).

Observe que, esta vez, son cuatro las actividades del script que se consideran actividades grandes.

## Paso 8: Restablecer para el siguiente ejercicio

Descarte el conjunto de acciones de trabajo:

```

ALTER WORK ACTION SET WORK1_WAS
ALTER WORK ACTION COUNT_READ_WA DISABLE
ALTER WORK ACTION COUNT_WRITE_WA DISABLE
ALTER WORK ACTION COUNT_DDL_WA DISABLE
ALTER WORK ACTION COUNT_CALL_WA DISABLE
ALTER WORK ACTION COUNT_LOAD_WA DISABLE
ALTER WORK ACTION COUNT_ALL_WA DISABLE
ALTER WORK ACTION COUNT_LARGE_READS DISABLE;
ALTER WORK ACTION SET WORK1_WAS DISABLE;
DROP WORK ACTION SET WORK1_WAS;

```

---

## Ejercicio 9: Captura de información detallada acerca de una actividad en ejecución

Este ejercicio muestra cómo puede utilizar el procedimiento `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` para capturar información detallada acerca de una actividad actualmente en ejecución para su posterior análisis histórico.

Tiempo estimado: 5-10 minutos

La información de actividad que captura se envía al supervisor de sucesos para actividades activo. En las tareas anteriores se mostraba cómo utilizar la cláusula `COLLECT ACTIVITY DATA` para capturar información de actividad detallada para las cargas de trabajo, clases de servicio, acciones de trabajo y umbrales. Esta cláusula debe especificarse con anterioridad al inicio de la ejecución de una actividad, y la información se envía al supervisor de sucesos de actividades cuando la actividad se ha completado. El procedimiento `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` le permite capturar información de forma reactiva cuando observa que existe un problema relacionado con una actividad que ya está en progreso. Cuando se utiliza este procedimiento, la

información relacionada con una actividad se envía inmediatamente al supervisor de sucesos de actividades. Se recopila información de actividad básica y de sentencia, pero no datos de entrada.

### Paso 1: Habilitar el supervisor de sucesos de actividades

Habilite el supervisor de sucesos para actividades existente que ha creado en el Ejercicio 1.

```
CONNECT TO SAMPLE  
  
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

### Paso 2: Emitir una consulta de ejecución larga

Desde el CLP, ejecute el siguiente script, que emite una consulta de ejecución larga con un cursor problemático:

```
db2 -tvf longquery.db2
```

### Paso 3: obtener el descriptor de contexto de aplicación

Desde una segunda ventana del CLP, llame a `WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES` para obtener el descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de la actividad. Mediante la unión de los resultados de la función de tabla a la vista administrativa `APPLICATIONS`, puede determinar qué actividad de cursor se ejecuta desde `longquery.db2`.

```
CONNECT TO SAMPLE  
  
SELECT T.APPLICATION_HANDLE, T.UOW_ID, T.ACTIVITY_ID, T.ACTIVITY_TYPE  
FROM SYSIBMADM.APPLICATIONS A,  
     TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES  
            (CAST(NULL AS BIGINT), -2)) T  
WHERE (A.AGENT_ID = T.APPLICATION_HANDLE) AND  
      (A.COORD_MEMBER = T.COORD_MEMBER) AND  
      (A.MEMBER = T.MEMBER) AND  
      (T.MEMBER = T.COORD_MEMBER) AND  
      (A.TPMON_CLIENT_APP = 'CLP longquery.db2')
```

La salida tiene un aspecto similar al que se muestra a continuación:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	ACTIVITY_TYPE
-----	-----	-----	-----
----			
	267	1	1 READ_DML

1 registro(s) seleccionado(s).

### Paso 4: Capturar información acerca de la actividad

Desde la misma ventana del CLP, llame al procedimiento almacenado `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` mediante la utilización del descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad obtenidos en el paso anterior:

```
CONNECT TO SAMPLE  
  
CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS (267, 1, 1)  
  
CONNECT RESET
```

Este paso envía información acerca de la actividad al supervisor de sucesos para actividades activo. Tenga en cuenta que, en su caso, el descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad podrían ser distintos.

### Paso 5: Ver la información de actividad

Observe la información que se ha recopilado para la actividad mediante la utilización de una sentencia como la siguiente:

```
SELECT VARCHAR(A.APPL_NAME, 15) as APPL_NAME,
        VARCHAR(A.TPMON_CLIENT_APP, 20) AS CLIENT_APP_NAME,
        VARCHAR(A.APPL_ID, 30) as APPL_ID,
        A.ACTIVITY_ID,
        A.UOW_ID,
        A.PARTIAL_RECORD,
        A.TIME_STARTED,
        A.TIME_COMPLETED,
        VARCHAR(S.STMT_TEXT, 300) AS STMT_TEXT
FROM ACTIVITY_DB2ACTIVITIES AS A,
     ACTIVITYSTMT_DB2ACTIVITIES AS S
WHERE A.APPL_ID = S.APPL_ID AND
      A.ACTIVITY_ID = S.ACTIVITY_ID AND
      A.UOW_ID = S.UOW_ID
```

La salida tiene un aspecto similar al siguiente:

```
APPL_NAME      CLIENT_APP_NAME  APPL_ID
ACTIVITY_ID    UOW_ID          PARTIAL_RECORD  TIME_STARTED
TIME_COMPLETED STMT_TEXT
-----
db2bp          CLP longquery.db2 *LOCAL.swa1kty.070928151408
1              1                1 2007-09-28-11.14.09.334636 0000-00-00-
00.00.00.000000 SELECT COUNT(*) FROM SYSCAT.TABLES, SYSCAT.TABLES,
SYSCAT.TABLES, SYSCAT.TABLES, SYSCAT.TABLES
```

Nota: las actividades que se capturan con el procedimiento `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` contienen algo menos de información que si se utilizara la cláusula `COLLECT ACTIVITY DATA` porque la actividad se captura antes de que haya completado su ejecución. En particular, no se aplican campos como la indicación de fecha y hora de finalización (que sólo muestra ceros) y el sqlcode. Puede determinar si una actividad se ha recopilado mediante el procedimiento `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` consultando la columna `PARTIAL_RECORD` de la tabla `ACTIVITY`. Si la columna `PARTIAL_RECORD` tiene el valor 1 (como se muestra en la salida anterior), la información de actividad se ha recopilado mediante `WLM_CAPTURE_ACTIVITY_IN_PROGRESS`. Si la columna `PARTIAL_RECORD` tiene el valor 0, la información de actividad se ha recopilado antes de que ésta se completara con la cláusula `COLLECT ACTIVITY DATA`.

---

## Ejercicio 10: Generación de datos históricos y de informes

Este ejercicio muestra cómo utilizar el ejemplo de la herramienta de análisis histórico del WLM

Tiempo estimado: 20-25 minutos

Los productos de base de datos DB2 incluyen scripts Perl de ejemplo que proporcionan la información que ha capturado el supervisor de sucesos de actividad de WLM para la realización del análisis histórico. Puede revisar o incluso modificar estos scripts para producir informes de análisis histórico adicionales que se ajusten a sus necesidades. Los scripts Perl son:

- wlmhist.pl: genera datos históricos.
- wlmhistrep.pl: produce informes a partir de los datos históricos.

## Paso 1: Crear las tablas de Explain

Para poder generar datos históricos, las tablas de Explain deben existir bajo el esquema del usuario que ejecuta la herramienta. Para crear las tablas de Explain, vaya al directorio /sqlib/misc y ejecute lo siguiente:

```
db2 CONNECT TO SAMPLE
```

```
db2 -tvf EXPLAIN.DDL
```

## Paso 2: Alterar la clase de servicio para recopilar datos de actividad

Habilite la recopilación de actividades especificando la cláusula COLLECT ACTIVITY DATA en el objeto de WLM de interés. Para este ejercicio, deseamos generar datos históricos para las actividades que se ejecutan en la subclase de servicio por omisión de la superclase de servicio de usuario por omisión:

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS  
COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS
```

## Paso 3: Habilitar el supervisor de sucesos de actividades

Puesto que el supervisor de sucesos de actividades se ha creado en el Ejercicio 1, Paso 1, habilítelo ahora si todavía no está habilitado.

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

## Paso 4: Ejecutar varias actividades

Ejecute varias actividades para que se recopilen datos de actividad en los que basar la generación de datos históricos.

```
db2 -tvf work1.db2
```

```
db2 -tvf work2.db2
```

## Paso 5: Inhabilitar el supervisor de sucesos para actividades

Se recomienda especialmente desactivar el supervisor de sucesos para las actividades antes de generar datos históricos. Si no lo hace, las actividades de DML que se ejecuten como resultado de la ejecución del generador de datos históricos también podrían capturarse y colocarse en las tablas de actividades del supervisor de sucesos de DB2 y, de esta forma, incrementar considerablemente el número de actividades reales para las que han de generarse datos de actividad.

```
CONNECT TO SAMPLE
```

```
SET EVENT MONITOR DB2ACTIVITIES STATE 0
```

## Paso 6: Generar datos históricos

Ejecute el script del generador de datos históricos, `wlmhist.pl`, para generar datos históricos para las actividades que han de capturarse en las tablas del supervisor de sucesos de actividades. El formato es el siguiente:

```
wlmhist.pl nombrebd usuario contraseña [desdeHora hastaHora IDcargaTrabajo  
nombreClaseServicio nombreSubclaseServicio tablaActividades tablaSentenciasActividad]
```

Utilice un guión (-) para eludir cualquier parámetro opcional.

Información adicional: el script del generador de datos históricos (`wlmhist.pl`) sólo generará datos históricos para DML. Si anteriormente ha ejecutado el script del generador de datos históricos (`wlmhist.pl`) una o más veces, se recomienda que, antes de volver a ejecutarlo, borre las tablas indicadas como `tablaActividades` y `tablaSentenciasActividad` para evitar la aparición de datos duplicados. Si opta por no borrar estas dos tablas, asegúrese de utilizar los parámetros de entrada `desdeHora` y `hastaHora` para estar seguro de que no se generan datos históricos para actividades cuyos datos ya se han generado.

Para este ejercicio, genere datos históricos para todas las actividades que se han capturado en el supervisor de sucesos de actividades.

```
Perl wlmhist.pl sample db2inst1 password
```

Puede que observe algunos errores similares a los siguientes:

```
Error running explain [IBM][CLI Driver][DB2/LINUX8664] SQL0418N Una  
sentencia contiene un marcador de parámetro que no es válido. SQLSTATE=42610  
for statement VALUES (TABLE_SCHEMA(:H00002 , :H00003 )) INTO :H00007
```

```
DBD::DB2::db do failed: [IBM][CLI Driver][DB2/LINUX8664] SQL0418N Una  
sentencia  
contiene un marcador de parámetro que no es válido. SQLSTATE=42610
```

Cuando se ejecuten datos históricos, Explain ha de ejecutarse en la sentencia real. En algunos casos, Explain no puede ejecutarse en algunas sentencias con marcadores de parámetro y se devuelve un error. No se generarán datos históricos para ninguna actividad que muestre un error de este tipo.

Cuando la herramienta haya completado la generación de datos históricos, le indicará para cuántas actividades ha generado datos históricos correctamente.

## Paso 7: Generar informes de datos históricos

Ejecute el script de generación de informes de datos históricos, `wlmhistrep.pl`, para generar informes basados en los datos que se han generado en el Paso 1. El formato es el siguiente:

```
wlmhistrep.pl aliasbd IDusuario contraseña [archivoSalida informe nombreEsquema desdeHora hastaHora emisor]
```

Utilice un guión (-) para eludir parámetros opcionales.

El parámetro **informe** puede ser cualquier combinación de las letras siguientes:

- A: Tablas con coincidencias
- B: Tablas sin coincidencias
- C: Índices con coincidencias
- D: Índices sin coincidencias
- E: Emisores

Si el parámetro **IDusuario** que especifica no es igual al que se ha utilizado para ejecutar el script `wlmhist.pl` cuando se creó la tabla `wlmhist`, debe especificar el valor `nombreEsquema` correcto. Los parámetros **desdeHora** y **hastaHora** deben especificarse en formato de indicación de fecha y hora (por ejemplo, `2007-06-06-17.00.00`).

Para este ejercicio, genere informes para las tablas con coincidencias y los índices sin coincidencias.

```
Perl wlmhistrep.pl sample db2inst1 password - AD
```

La salida tendrá un aspecto similar al que se muestra a continuación:

TABLES HIT REPORT FOR DATABASE sample

TABLE NAME	TABLE SCHEMA	% HITS	TOTAL HITS
EMPLOYEE	KARENAM	7.14285714	2
INVENTORY	KARENAM	14.28571429	4
ORG	KARENAM	28.57142857	8
SALES	KARENAM	14.28571429	4
SYSROUTINES	SYSIBM	7.14285714	2
SYSTABLES	SYSIBM	21.42857143	6
SYSTABLESPACES	SYSIBM	7.14285714	2

INDEXES NOT HIT REPORT FOR DATABASE sample

TABLE NAME	TABLE SCHEMA	INDEX NAME	INDEX SCHEMA	INDEX TYPE
EXPLAIN_ARGUMENT	KARENAM	ARG_I1	KARENAM	REG
HMON_ATM_INFO	SYSTOOLS	ATM_UNIQ	SYSTOOLS	REG
CUSTOMER	KARENAM	CUST_CID_XMLIDX	KARENAM	XVIL
CUSTOMER	KARENAM	CUST_NAME_XMLIDX	KARENAM	XVIL
CUSTOMER	KARENAM	CUST_PHONES_XMLIDX	KARENAM	XVIL
CUSTOMER	KARENAM	CUST_PHONET_XMLIDX	KARENAM	XVIL
EXPLAIN_DIAGNOSTIC	KARENAM	EXP_DIAG_DAT_I1	KARENAM	REG
HMON_COLLECTION	SYSTOOLS	HI_OBJ_UNIQ	SYSTOOLS	REG
ADVISE_INDEX	KARENAM	IDX_I1	KARENAM	REG
ADVISE_INDEX	KARENAM	IDX_I2	KARENAM	REG
SYSATTRIBUTES	SYSIBM	INDATTRIBUTES01	SYSIBM	REG
SYSATTRIBUTES	SYSIBM	INDATTRIBUTES02	SYSIBM	REG
:				
:				

### Paso 8: Restablecer para el siguiente ejercicio

Inhabilite la recopilación de actividades para la subclase de servicio por omisión de la superclase de servicio de usuario por omisión y borre las tablas de actividades.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
COLLECT ACTIVITY DATA NONE
```

```
DELETE FROM ACTIVITY_DB2ACTIVITIES
DELETE FROM ACTIVITYSTMT_DB2ACTIVITIES
```

---

## Ejercicio 11: Utilización de agregados ampliados para clases de servicio

Este ejercicio muestra cómo utilizar la opción COLLECT AGGREGATE ACTIVITY DATA EXTENDED en una clase de servicio para producir histogramas de tiempos entre llegadas de actividades de coordinador y costes estimados.

Tiempo estimado: 25-30 minutos

El tiempo entre llegadas es el tiempo transcurrido entre la llegada al sistema de una actividad y la llegada de la siguiente actividad. El coste estimado de una actividad son los recursos estimados del sistema que se utilizarán en la ejecución de la actividad y sólo se aplica a las actividades de DML. Un histograma de tiempo entre llegadas puede ser útil para correlacionarlo con un histograma de vida útil o con otras estadísticas de vida útil con el fin de determinar si un cambio en las estadísticas de vida útil ha sido el resultado de un cambio en la velocidad de llegada de la carga de trabajo o bien ha sido el resultado de un cambio en la complejidad de la carga de trabajo (consultas más complejas) o un cambio en el sistema. El histograma de coste estimado puede ser útil para correlacionarlo con los histogramas de tiempo entre llegadas y de vida útil con el fin de determinar si un cambio en el histograma de vida útil puede deberse a un cambio en la complejidad de la carga de trabajo (consultas más complejas con envío de costes estimados más altos), debido a un cambio en la velocidad de llegada de las actividades (determinado a partir de la distribución del tiempo entre llegadas) o debido a un cambio en el propio sistema, como la especificación de un nuevo umbral, un cambio en la prioridad asignada a una clase de servicio o un cambio en el hardware.

Para obtener más información acerca de los histogramas, consulte “Histogramas en la gestión de carga de trabajo” en la página 278.

El acceso a los histogramas se realiza por medio del supervisor de sucesos de estadísticas. En este ejercicio vuelve a utilizarse el supervisor de sucesos de estadísticas que se ha creado en el Ejercicio 1, Paso 1.

### Paso 1: Crear vistas para visualizar las estadísticas de histograma

Cree varias vistas para facilitar la realización de consultas de la tabla HISTOGRAMBIN\_DB2STATISTICS. En la primera vista se visualizarán todos los tipos de histogramas disponibles. En este ejercicio, sólo se informará de los tres tipos básicos: vida útil, tiempo de ejecución y tiempo en cola.

```
CREATE VIEW HISTOGRAMTYPES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) HISTOGRAM_TYPE
  FROM HISTOGRAMBIN_DB2STATISTICS
```

Una segunda vista facilitará la averiguación de para qué clases de servicio han de recopilarse histogramas. La tabla HISTOGRAMBIN\_DB2STATISTICS informa acerca de la clase de servicio para la que se recopila el histograma mediante la especificación del ID de clase de servicio. La unión de esta tabla a la tabla de catálogo SERVICECLASSES permitirá presentar la información de clase de servicio con el nombre de la superclase de servicio y el nombre de la subclase de servicio en lugar del ID de clase de servicio.

```
CREATE VIEW HISTOGRAMSERVICECLASSES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) HISTOGRAM_TYPE,
    SUBSTR(PARENTSERVICECLASSNAME,1,24) SERVICE_SUPERCLASS,
```

```

        SUBSTR(SERVICECLASSNAME,1,24) SERVICE_SUBCLASS
FROM HISTOGRAMBIN_DB2STATISTICS H,
    SYSCAT.SERVICECLASSES S
WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID

```

En la tercera vista se visualizarán todas las horas a las que se ha recopilado un histograma de un tipo determinado para una clase de servicio determinada. Como la vista HISTOGRAMSERVICECLASSES, une la tabla HISTOGRAMBIN\_DB2STATISTICS a la tabla de catálogo SERVICECLASSES. La diferencia está en la columna STATISTICS\_TIMESTAMP, que se incluye como una de las columnas de esta vista.

```

CREATE VIEW HISTOGRAMTIMES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) HISTOGRAM_TYPE,
    SUBSTR(PARENTSERVICECLASSNAME,1,24) SERVICE_SUPERCLASS,
    SUBSTR(SERVICECLASSNAME,1,24) SERVICE_SUBCLASS,
    STATISTICS_TIMESTAMP TIMESTAMP
FROM HISTOGRAMBIN_DB2STATISTICS H,
    SYSCAT.SERVICECLASSES S
WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID

```

La cuarta y última vista se utilizará para mostrar los propios histogramas. También muestra una tarea común que ha de realizarse cuando se utilizan histogramas, que es agregar éstos a lo largo del tiempo. Esta vista muestra el valor superior de cada intervalo y el número de actividades que se ha contado para cada intervalo. En los dos histogramas siguientes, el campo BIN\_TOP mide el número de milisegundos del tiempo entre llegadas de actividades y el número de timerons en el coste estimado. Cuando el valor de BIN\_TOP es, por ejemplo, de 3.000 milisegundos y el valor de BIN\_TOP del intervalo anterior es de 2.000 milisegundos y el valor de NUMBER\_IN\_BIN es diez para un histograma de tiempo entre llegadas, puede determinar que diez actividades han llegado al sistema, individualmente, entre 2 y 3 segundos después de la llegada de la actividad anterior, por ejemplo.

```

CREATE VIEW HISTOGRAMS(HISTOGRAM_TYPE, SERVICE_SUPERCLASS,
  SERVICE_SUBCLASS, BIN_TOP, NUMBER_IN_BIN) AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) HISTOGRAM_TYPE,
    SUBSTR(PARENTSERVICECLASSNAME,1,24) SERVICE_SUPERCLASS,
    SUBSTR(SERVICECLASSNAME,1,24) SERVICE_SUBCLASS,
    TOP AS BIN_TOP,
    SUM(NUMBER_IN_BIN) AS NUMBER_IN_BIN
FROM HISTOGRAMBIN_DB2STATISTICS H,
    SYSCAT.SERVICECLASSES S
WHERE H.SERVICE_CLASS_ID = S.SERVICECLASSID
GROUP BY HISTOGRAM_TYPE, PARENTSERVICECLASSNAME, SERVICECLASSNAME, TOP

```

## Paso 2: Activar la recopilación de histogramas

Para la clase de usuario de servicio por omisión, la activación de la recopilación de histogramas se realiza alterando su subclase por omisión para recopilar datos de actividad agregada con la opción EXTENDED. Con ello, los tres histogramas están disponibles en la opción BASE (vida útil, tiempo de ejecución y tiempo en cola) y los dos histogramas sólo están disponibles cuando se utiliza la opción EXTENDED (tiempo entre llegadas y coste estimado).

```

ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA EXTENDED

```

## Paso 3: Activar el supervisor de sucesos de estadísticas

Si todavía no está activo, active el supervisor de sucesos de estadísticas que se ha creado anteriormente para que éste pueda recibir los datos agregados siempre que se recopilen.

## Paso 4: Ejecutar actividades y enviar estadísticas al supervisor de sucesos de estadísticas

En primer lugar, ejecute varias actividades; después de que las actividades se hayan completado, se llama al procedimiento almacenado WLM\_COLLECT\_STATS para enviar las estadísticas de clase de servicio al supervisor de sucesos de estadísticas activo (incluidos los histogramas de vida útil de actividad, de tiempo de ejecución, de tiempo en cola, de tiempo entre llegadas y de coste estimado para la clase de servicio de usuario por omisión). Estos histogramas contienen datos acerca de todas las actividades que se han ejecutado en la clase de servicio de usuario por omisión desde que se habilitaran las estadísticas de actividades agregadas. La llamada al procedimiento almacenado también restablece las estadísticas. Para mostrar los cambios en la actividad de base de datos a lo largo del tiempo, se crean tres intervalos de recopilación.

En el primer intervalo, ejecute dos scripts, work1.db2 y work2.db2, y recopile y restablezca las estadísticas.

```
db2 -o- -tvf work1.db2
db2 -o- -tvf work2.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS
```

En el segundo intervalo, sólo ejecute el script work1.db2 una vez y recopile y restablezca las estadísticas.

```
db2 -o- -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS
```

En el tercer intervalo, ejecute el script work1.db2 dos veces y ejecute el script work2.db2 una vez y recopile y restablezca las estadísticas.

```
db2 -o- -tvf work1.db2
db2 -o- -tvf work2.db2
db2 -o- -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS
```

La recopilación periódica de datos de esta forma le permite observar cómo cambia el trabajo en el sistema a lo largo del tiempo.

Información adicional: la recopilación periódica de datos no ha de ser necesariamente una operación manual. Mediante la utilización del parámetro de configuración de base de datos WLM\_COLLECT\_INT, puede establecerse el intervalo en minutos transcurrido el cual habrá de producirse automáticamente la recopilación y el restablecimiento de las estadísticas.

## Paso 5: Consultar vistas para ver estadísticas

Ahora que se han recopilado estadísticas, las vistas anteriormente creadas pueden utilizarse para ver las estadísticas. La vista HISTOGRAMTYPES sólo devuelve los tipos de histogramas disponibles.

```
SELECT * FROM HISTOGRAMTYPES
```

```
HISTOGRAM_TYPE
-----
CoordActEstCost
CoordActExecTime
CoordActInterArrivalTime
CoordActLifetime
CoordActQueueTime
```

3 registro(s) seleccionado(s).

Puesto que la opción EXTENDED se ha utilizado al alterar la clase de servicio, existen cinco histogramas.

La vista HISTOGRAMSERVICECLASSES le permite ver las clases de servicio para las que se ha recopilado un histograma. El ejemplo siguiente limita la salida a sólo la salida del histograma CoordActInterArrivalTime. Puesto que la recopilación de actividades agregadas sólo se ha activado para la subclase por omisión de la clase de servicio de usuario por omisión, sólo se muestra esa clase al realizar la selección desde la vista HISTOGRAMSERVICECLASSES.

```
SELECT * FROM HISTOGRAMSERVICECLASSES
WHERE HISTOGRAM_TYPE = 'CoordActInterArrivalTime'
ORDER BY SERVICE_SUPERCLASS, SERVICE_SUBCLASS
```

```
HISTOGRAM_TYPE          SERVICE_SUPERCLASS      SERVICE_SUBCLASS
-----
-
CoordActInterArrivalTime SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS
```

1 registro(s) seleccionado(s).

La vista HISTOGRAMTIMES muestra las horas a las que se han recopilado los histogramas. Puesto que el procedimiento WLM\_COLLECT\_STATS se ha ejecutado tres veces, existen tres indicaciones de fecha y hora para el histograma de tiempo entre llegadas que se muestra.

```
SELECT * FROM HISTOGRAMTIMES
WHERE HISTOGRAM_TYPE = 'CoordActInterArrivalTime'
AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
ORDER BY TIMESTAMP
```

```
HISTOGRAM_TYPE          SERVICE_SUPERCLASS      SERVICE_SUBCLASS
TIMESTAMP
-----
-
CoordActInterArrivalTime SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS
2007-08-08-13.41.38.870298
CoordActInterArrivalTime SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS
2007-08-08-13.41.42.802855
CoordActInterArrivalTime SYSDEFAULTUSERCLASS      SYSDEFAULTSUBCLASS
2007-08-08-13.41.53.577835
```

La última vista, HISTOGRAMS, se destina a la visualización de los propios histogramas. A diferencia de la vista HISTOGRAMTIMES, que muestra cada intervalo de recopilación en su propia fila, esta vista agrega datos de histograma de los diversos intervalos para producir un único histograma de un tipo determinado para una clase de servicio determinada.

```
SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
WHERE HISTOGRAM_TYPE = 'CoordActInterArrivalTime'
AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
```

ORDER BY BIN\_TOP

BIN_TOP	NUMBER_IN_BIN
-1	0
1	10
2	6
3	7
5	14
8	7
12	32
19	2
29	9
44	24
68	11
103	8
158	8
241	9
369	1
562	10
858	5
1309	5
1997	0
3046	0
4647	0
7089	0
10813	2
16493	2
25157	0
38373	0
58532	0
89280	0
136181	0
207720	0
316840	0
483283	0
737162	0
1124409	0
1715085	0
2616055	0
3990325	0
6086529	0
9283913	0
14160950	0
21600000	0

41 registro(s) seleccionado(s).

La ejecución de esta consulta produce salida que no será exactamente igual a la que se muestra en el ejemplo anterior porque los tiempos entre llegadas de actividad dependen del rendimiento del sistema. En la salida anterior, existen 41 intervalos y todos los intervalos más grandes están vacíos. En la parte superior existe un intervalo cuyo valor BIN\_TOP es -1. Este intervalo representa todas las actividades cuyo tiempo entre llegadas era demasiado grande para poder representarse en el histograma. El hecho de ver un NUMBER\_OF\_BIN mayor que cero cuando BIN\_TOP es -1 indica que probablemente debe incrementar el valor de intervalo alto del histograma. En la salida anterior, NUMBER\_IN\_BIN es 0 y por lo tanto no es necesario realizar este cambio. La mayoría de las actividades han llegado a menos de 1.309 milisegundos entre sí. Cuatro actividades han llegado entre 7.089 milisegundos y 16.493 milisegundos entre sí.

La misma consulta puede repetirse con un tipo de histograma CoordActEstCost en lugar de CoordActInterArrivalTime.

```

SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
  WHERE HISTOGRAM_TYPE = 'CoordActEstCost'
    AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
    AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
  ORDER BY BIN_TOP

```

BIN_TOP	NUMBER_IN_BIN
-1	0
1	39
2	0
3	0
5	0
8	30
12	0
19	30
29	0
44	0
68	0
103	0
158	0
241	0
369	0
562	0
858	0
1309	0
1997	0
3046	0
4647	0
7089	0
10813	0
16493	0
25157	0
38373	0
58532	0
89280	0
136181	0
207720	0
316840	0
483283	0
737162	0
1124409	0
1715085	0
2616055	0
3990325	0
6086529	0
9283913	0
14160950	0
21600000	0

41 registro(s) seleccionado(s).

Este tipo de histograma es habitual para una carga de trabajo pequeña. Con una carga de trabajo pequeña, no existe mucha variación en el tamaño de las actividades, por lo tanto, sólo existen tres intervalos distintos para los que se han contado actividades. Algo más del 60% de las actividades tenían un coste estimado comprendido entre 5 y 19 timerons, y el resto tenían costes estimados inferiores a 1 timeron.

## Paso 6: Restablecer para otros ejercicios

El último paso consiste en desactivar la recopilación de actividades agregadas en la clase de servicio de usuario por omisión, descartar las vistas y suprimir la información de las tablas de estadísticas.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS  
COLLECT AGGREGATE ACTIVITY DATA NONE
```

```
DROP VIEW HISTOGRAMS  
DROP VIEW HISTOGRAMTIMES  
DROP VIEW HISTOGRAMSERVICECLASSES  
DROP VIEW HISTOGRAMTYPES
```

```
SET EVENT MONITOR DB2STATISTICS STATE 0
```

```
DELETE FROM HISTOGRAMBIN_DB2STATISTICS  
DELETE FROM SCSTATS_DB2STATISTICS
```



---

## Capítulo 7. Casos de ejemplo de gestión de carga de trabajo

---

### Aplicación de ejemplo de gestión de carga de trabajo

En el servidor de datos de DB2 se han integrado completas funciones de gestión de carga de trabajo con la gestión de carga de trabajo de DB2, que le proporcionan un control más preciso sobre las actividades, los recursos y el rendimiento, y una perspectiva más exhaustiva de la ejecución del sistema. Ahora existe una aplicación de ejemplo de gestión de carga de trabajo en developerWorks.

La aplicación de ejemplo de gestión de carga de trabajo muestra cómo puede utilizar las funciones de la gestión de carga de trabajo de DB2 para lograr los objetivos siguientes:

#### **Proteger el sistema de consultas de escape**

Las consultas de escape son costosas y empobrecen el rendimiento. La aplicación de ejemplo de gestión de carga de trabajo identifica las consultas que potencialmente puede ser consultas de escape y, a continuación, detiene la ejecución de esas consultas después de que éstas han infringido un umbral especificado.

#### **Limitar el consumo de recursos simultáneos de aplicaciones individuales**

La aplicación de ejemplo muestra cómo utilizar las funciones de la gestión de carga de trabajo de DB2 para evitar que las aplicaciones que envían grandes cantidades de trabajo simultáneo afecten negativamente al rendimiento de las demás aplicaciones.

#### **Obtener un tiempo de respuesta específico**

Las funciones de gestión de carga de trabajo le permiten lograr un objetivo de tiempo de respuesta específico de tipo "la transacción X de la aplicación Y debe completarse en 1 segundo en el 90% de los casos", con independencia de cualquier otro tipo de actividad que esté ejecutándose simultáneamente en el sistema. La aplicación de ejemplo mostrará cómo lograr un objetivo de tiempo de respuesta.

#### **Tiempo de respuesta coherente para consultas breves**

Las consultas que generalmente tienen un tiempo de respuesta de menos de 1 segundo deberían tener un tiempo de respuesta relativamente coherente, con independencia de cualquier otro tipo de carga de trabajo que esté ejecutándose en el sistema. La aplicación de ejemplo utiliza el histograma de tiempo de ejecución de consulta para supervisar la coherencia.

#### **Proteger el sistema durante los períodos de máxima demanda**

Las funciones de la política de gestión de carga de trabajo protegen el sistema contra una sobrecarga de la capacidad durante los picos de máxima demanda colocando el trabajo en cola cuando la carga del sistema es suficientemente alta.

#### **Habilitar el proceso de extracción, transformación y carga (ETL) de lotes simultáneos y las consultas de usuario**

Las funciones de gestión de carga de trabajo le permiten ejecutar trabajos ETL (como cargar datos en tablas) a la vez que controla el impacto en el rendimiento para los usuarios que ejecutan consultas simultáneamente.

Para obtener la aplicación de ejemplo, consulte Workload management sample en developerWorks.

## Caso de ejemplo: Investigación de una ralentización del sistema relacionada con la carga de trabajo

Si observa una ralentización del sistema (por ejemplo, algunas aplicaciones tardan mucho más de lo esperado en completarse) y no está seguro de si el problema tiene relación con la configuración de las cargas de trabajo, puede utilizar los datos defunción de tabla para investigar y, si es necesario, corregir el problema.

En primer lugar, cree una consulta que agregue los datos de las diferentes clases de servicio y miembros de base de datos utilizando los datos de la función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS. Establezca series vacías para el primer y el segundo argumento, y establezca el tercer argumento en -2 para indicar que los datos deben recopilarse para todas las clases de servicio en todos los miembros de base de datos.

Su consulta puede parecerse a la siguiente:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(SUM(COORD_ACT_COMPLETED_TOTAL)),1,13) AS ACTSCOMPLETED,
       SUBSTR(CHAR(SUM(COORD_ACT_ABORTED_TOTAL)),1,11) AS ACTSABORTED,
       SUBSTR(CHAR(MAX(CONCURRENT_ACT_TOP)),1,6) AS ACTSHW,
       CAST(CASE WHEN SUM(COORD_ACT_COMPLETED_TOTAL) = 0 THEN 0
              ELSE SUM(COORD_ACT_COMPLETED_TOTAL * COORD_ACT_LIFETIME_AVG)
                 / SUM(COORD_ACT_COMPLETED_TOTAL) END / 1000 AS DECIMAL(9,3))
       AS ACTAVGLIFETIME
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS('', '', -2)) AS SCSTATS
GROUP BY SERVICE_SUPERCLASS_NAME, SERVICE_SUBCLASS_NAME
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME
```

SUPERCLASS_NAME	SUBCLASS_NAME	ACTSCOMPLETED	ACTSABORTED	ACTSHW	ACTAVGLIFETIME
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	20	0	1	3.750
SUP1	SUB1	40	0	8	7.223

En los datos del ejemplo anterior, la subclase de servicio SUB1 en la superclase de servicio SUP1 está ejecutando más actividades simultáneas que de costumbre. Para investigarlo más a fondo, es posible que desee examinar las estadísticas de las cargas de trabajo que se correlacionan con esta clase de servicio. Su consulta puede parecerse a la siguiente:

```
SELECT SUBSTR(WLSTATS.WORKLOAD_NAME,1,22) AS WL_NAME,
       SUBSTR(CHAR(WLSTATS.MEMBER),1,4) AS MEMB,
       CONCURRENT_WLO_TOP AS WLO_HIGH_WTRMRK,
       CONCURRENT_WLO_ACT_TOP AS WLO_ACT_HIGH_WTRMRK
FROM TABLE(WLM_GET_WORKLOAD_STATS('', -2)) AS WLSTATS,
     TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES('', '', -2)) AS SCWLOS
WHERE WLSTATS.WORKLOAD_NAME = SCWLOS.WORKLOAD_NAME
AND SCWLOS.SERVICE_SUPERCLASS_NAME = 'SUP1'
AND SCWLOS.SERVICE_SUBCLASS_NAME = 'SUB1'
ORDER BY WL_NAME, MEMB;
```

WL_NAME	MEMB	WLO_HIGH_WTRMRK	WLO_ACT_HIGH_WTRMRK
LYNNSALES	0	2	8
LYNNSALES	1	0	0
SYSDEFAULTUSERWORKLOAD	0	1	1
SYSDEFAULTUSERWORKLOAD	1	0	0

La salida muestra que una aplicación en la carga de trabajo LYNN SALES ha enviado 8 actividades al mismo tiempo. Plántese añadir un umbral para limitar la simultaneidad de actividades de coordinador para cada aparición de la carga de trabajo.

---

## Caso de ejemplo: Agregación de métricas de actividad a través de actividades hijas

Cada fila de un supervisor de sucesos de actividades contiene información y métricas de supervisión que describen la ejecución de una actividad específica. La actividad incluye como información los elementos de supervisor **parent\_uow\_id** y **parent\_activity\_id**. Estos elementos de supervisor tienen un valor distinto de cero si la actividad es hija de alguna otra actividad (si está anidada) y si los elementos identifican a la actividad padre.

Mediante el empleo del SQL recursivo, se pueden agregar métricas de supervisor en un supervisor de sucesos de actividades a su actividad padre a través de todas las actividades hijas. La observación de las métricas agregadas correspondientes al padre resulta útil para identificar qué partes de un procedimiento almacenado pueden beneficiarse de un ajuste. Por ejemplo, si un procedimiento almacenado P1 llama a los procedimientos almacenados P2 y P3, las métricas agregadas pueden mostrar que el 90% del uso de la CPU durante una llamada a P1 se produce durante el proceso de las sentencias en el procedimiento anidado P3. A partir de ello, puede enfocar su investigación en consecuencia.

Puede utilizar lo siguiente como ejemplo de obtención de agregados de métricas anidadas desde un supervisor de sucesos de actividad.

**Nota:** En este ejemplo, el carácter @ se utiliza como carácter de terminación de sentencia.

Las sentencias siguientes pueden copiarse en un archivo (por ejemplo, test.clp) y pueden ejecutarse utilizando db2 -td@ -f *nombre\_archivo*, tal como se muestra en el mandato de ejemplo siguiente:

```
db2 -td@ -f test.clp
```

Para ilustrar este ejemplo, supongamos que existen las tablas y los procedimientos almacenados siguientes:

```
CREATE TABLE T1 (ONE INT)@
```

```
DROP PROCEDURE TEST.P1@
```

```
DROP PROCEDURE TEST.P2@
```

```
DROP PROCEDURE TEST.P3@
```

```
DROP PROCEDURE TEST.P4@
```

```
CREATE PROCEDURE TEST.P4()
```

```
LANGUAGE SQL
```

```
BEGIN
```

```
    INSERT INTO T1 VALUES(5);
```

```
    INSERT INTO T1 VALUES(6);
```

```
    INSERT INTO T1 VALUES(7);
```

```
END@
```

```
CREATE PROCEDURE TEST.P3()
```

```
LANGUAGE SQL
```

```
BEGIN
```

```
    DECLARE V INTEGER;
```

```
    INSERT INTO T1 VALUES(1);
```

```
    CALL TEST.P4();
```

```

SELECT COUNT(*) INTO V FROM T1;
END@

```

```

CREATE PROCEDURE TEST.P2()
LANGUAGE SQL
BEGIN
  INSERT INTO T1 VALUES(2);
  INSERT INTO T1 VALUES(3);
END@

```

```

CREATE PROCEDURE TEST.P1()
LANGUAGE SQL
BEGIN
  CALL TEST.P3();
  CALL TEST.P2();
  INSERT INTO T1 VALUES(4);
END@

```

## Procedimiento

1. Crear una supervisión de sucesos de actividades y habilitar la captación de actividades. En este ejemplo, la captación de actividades se habilita para la conexión actual mediante el procedimiento WLM\_SET\_CONN\_ENV. Ejecute el procedimiento TEST.P1 utilizando la misma conexión y luego inhabilite el supervisor de sucesos de actividades. La información de la actividad se capturará para el procedimiento TEST.P1 y para todas sus actividades hijas.

```

CREATE EVENT MONITOR A FOR ACTIVITIES WRITE TO TABLE@
SET EVENT MONITOR A STATE 1@

```

```

CALL WLM_SET_CONN_ENV(NULL, '<collectactdata>WITH DETAILS</collectactdata>
<collectactpartition>ALL</collectactpartition>')@

```

```

CALL TEST.P1()@

```

```

SET EVENT MONITOR A STATE 0@

```

```

CALL WLM_SET_CONN_ENV(NULL, '<collectactdata>NONE</collectactdata>')@

```

2. Ejecute la consulta siguiente para mostrar las sentencias capturadas, junto con la información de identificación padre, el consumo de CPU individual y el consumo de CPU agregado. El consumo de CPU agregado se suma de forma recurrente por todas las actividades hijas. Esta consulta puede extenderse fácilmente para dar soporte a cualquier métrica de supervisión capturada mediante el supervisor de sucesos de actividades.

**Nota:** Para mejorar la legibilidad, solo se informará aquí sobre el consumo de CPU.

```

WITH ACT( APPL_ID,
          UOW_ID,
          ACTIVITY_ID,
          PARENT_UOW_ID,
          PARENT_ACTIVITY_ID,
          CPU )
AS ( SELECT APPL_ID,
          UOW_ID,
          ACTIVITY_ID,
          MAX(PARENT_UOW_ID),
          MAX(PARENT_ACTIVITY_ID),
          SUM(METRICS.TOTAL_CPU_TIME)
FROM ACTIVITY_A AS A,
XMLTABLE (XMLNAMESPACES( DEFAULT 'http://www.ibm.com/xmlns/prod/db2/mon'),
          '$actmetrics/activity_metrics' PASSING XMLPARSE(DOCUMENT A.DETAILS_XML) as "actmetrics"
          COLUMNS
          TOTAL_CPU_TIME BIGINT PATH 'total_cpu_time' ) AS METRICS

```

```

WHERE A.PARTIAL_RECORD = 0
GROUP BY APPL_ID,
        UOW_ID,
        ACTIVITY_ID ),
TMP( BASE_APPL_ID,
      BASE_UOW_ID,
      BASE_ACTIVITY_ID,
      APPL_ID,
      UOW_ID,
      ACTIVITY_ID,
      PARENT_UOW_ID,
      PARENT_ACTIVITY_ID,
      CPU,
      LEVEL )
AS ( SELECT APPL_ID,
          UOW_ID,
          ACTIVITY_ID,
          APPL_ID,
          UOW_ID,
          ACTIVITY_ID,
          PARENT_UOW_ID,
          PARENT_ACTIVITY_ID,
          CPU,
          1
      FROM ACT
      UNION ALL
      SELECT T.BASE_APPL_ID,
            T.BASE_UOW_ID,
            T.BASE_ACTIVITY_ID,
            A.APPL_ID,
            A.UOW_ID,
            A.ACTIVITY_ID,
            A.PARENT_UOW_ID,
            A.PARENT_ACTIVITY_ID,
            A.CPU,
            T.LEVEL + 1
      FROM ACT AS A, TMP AS T
      WHERE A.APPL_ID = T.APPL_ID AND
            A.PARENT_UOW_ID = T.UOW_ID AND
            A.PARENT_ACTIVITY_ID = T.ACTIVITY_ID AND
            T.LEVEL < 128 ),
AGG( APPL_ID,
      UOW_ID,
      ACTIVITY_ID,
      CPU )
AS ( SELECT BASE_APPL_ID,
          BASE_UOW_ID,
          BASE_ACTIVITY_ID,
          SUM(CPU)
      FROM TMP
      GROUP BY BASE_APPL_ID,
              BASE_UOW_ID,
              BASE_ACTIVITY_ID )
SELECT
A.UOW_ID,
A.ACTIVITY_ID,
A.PARENT_UOW_ID,
A.PARENT_ACTIVITY_ID,
A.CPU AS STMT_CPU,
B.CPU AS AGG_CPU,
SUBSTR(CONCAT(REPEAT(' ',C.STMT_NEST_LEVEL),
              C.STMT_TEXT),
1, 30) AS STMT_TEXT
FROM ACT AS A,
AGG AS B,
ACTIVITYSTMT_A AS C,
ACTIVITY_A AS D

```

```

WHERE A.APPL_ID = B.APPL_ID AND
      A.UOW_ID = B.UOW_ID AND
      A.ACTIVITY_ID = B.ACTIVITY_ID AND
      D.COORD_PARTITION_NUM = D.PARTITION_NUMBER AND
      A.APPL_ID = C.APPL_ID AND
      A.UOW_ID = C.UOW_ID AND
      A.ACTIVITY_ID = C.ACTIVITY_ID AND
      A.APPL_ID = D.APPL_ID AND
      A.UOW_ID = D.UOW_ID AND
      A.ACTIVITY_ID = D.ACTIVITY_ID AND
      D.PARTIAL_RECORD = 0
ORDER BY D.TIME_CREATED ASC@

```

Esta consulta genera la salida siguiente. La columna STMT\_CPU informa sobre el consumo de CPU correspondiente a la sentencia sin incluir el uso de la CPU de ninguna actividad hija. La columna AGG\_CPU informa sobre el consumo de CPU agregado correspondiente a la actividad y a todas sus hijas.

**Nota:** Si la velocidad en la ejecución de las consultas es tal que no permite su medición, STMT\_CPU y AGG\_CPU podrían tener un valor cero.

UOW_ID	ACTIVITY_ID	PARENT_UOW_ID	PARENT_ACTIVITY_ID	STMT_CPU	AGG_CPU	STMT_TEXT
576	1	0	0	0	5353	84064 CALL TEST.P1()
576	2	576	1	1	7444	52043 CALL TEST.P3()
576	3	576	2	2	1869	1869 INSERT INTO T1 VALUES(1)
576	4	576	2	2	11727	26935 CALL TEST.P4()
576	5	576	4	4	2017	2017 INSERT INTO T1 VALUES(5)
576	6	576	4	4	6602	6602 INSERT INTO T1 VALUES(6)
576	7	576	4	4	6589	6589 INSERT INTO T1 VALUES(7)
576	8	576	2	2	15795	15795 SELECT COUNT(*) INTO :HV00
576	9	576	1	1	11727	20314 CALL TEST.P2()
576	10	576	9	9	1941	1941 INSERT INTO T1 VALUES(2)
576	11	576	9	9	6646	6646 INSERT INTO T1 VALUES(3)
576	12	576	1	1	6354	6354 INSERT INTO T1 VALUES(4)

12 registro(s) seleccionado(s).

## Caso de ejemplo: Identificación de actividades que tardan demasiado tiempo en completarse

Las funciones de tabla de gestión de carga de trabajo simplifican la tarea de identificar una actividad específica dentro del servidor de datos y, si es necesario, cancelarla sin tener que finalizar toda la aplicación.

### Identificación de una actividad que tarda demasiado en completarse

A continuación se muestra un ejemplo de cómo identificar una consulta de ejecución larga. Supongamos que un usuario del departamento de ventas que ejecuta la aplicación SalesReport se queja de que la aplicación tarda demasiado en completarse.

Después de identificar el descriptor de contexto de la aplicación, utilice la función de tabla WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para buscar todas las actividades actualmente en ejecución en esta aplicación. Por ejemplo, si el descriptor de contexto de la aplicación es 1, su consulta puede parecerse a la siguiente:

```

SELECT SUBSTR(CHAR(COORD_MEMBER),1,5) AS COORD,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(UOW_ID),1,5) AS UOWID,
       SUBSTR(CHAR(ACTIVITY_ID),1,5) AS ACTID,
       SUBSTR(CHAR(PARENT_UOW_ID),1,8) AS PARUOWID,
       SUBSTR(CHAR(PARENT_ACTIVITY_ID),1,8) AS PARACTID,
       SUBSTR(CHAR(ACTIVITY_TYPE),1,8) AS ACTTYPE,

```

```

SUBSTR(CHAR(NESTING_LEVEL),1,7) AS NESTING
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(1, -2))
AS WLOACTS
ORDER BY MEMB, UOWID, ACTID

```

COORD	MEMB	UOWID	ACTID	PARUOWID	PARACTID	ACTTYPE	NESTING
0	0	2	3	-	-	CALL	0
0	0	2	5	2	3	READ_DML	1

Se identifica que la actividad tiene el ID de unidad de trabajo 2 y el ID de actividad 5. A continuación, puede utilizar la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS para descubrir qué hacen los agentes que trabajan en esta actividad:

```

SELECT APPLICATION_HANDLE, UOW_ID, ACTIVITY_ID,
SUBSTR(REQUEST_TYPE,1,8) AS REQUEST_TYPE,
SUBSTR(EVENT_TYPE,1,8) AS EVENT_TYPE,
SUBSTR(EVENT_OBJECT,1,8) AS EVENT_OBJECT
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS(' ', ' ', CAST(NULL AS BIGINT),-2))
AS AGENTS
WHERE APPLICATION_HANDLE = 1
AND UOW_ID = 2
AND ACTIVITY_ID = 5

```

Por ejemplo, la actividad puede estar en cola, en ejecución o a la espera de un bloqueo. Si la actividad se hubiera colocado en cola, el resultado sería:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	REQUEST_TYPE	EVENT_TYPE	EVENT_OBJECT
1	2	5	OPEN	WAIT	WLM_QUEUE

Si la actividad se estuviera ejecutando, el resultado sería:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	REQUEST_TYPE	EVENT_TYPE	EVENT_OBJECT
1	2	5	OPEN	PROCESS	REQUEST

Si la actividad estuviera a la espera de un bloqueo, el resultado sería:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	REQUEST_TYPE	EVENT_TYPE	EVENT_OBJECT
1	2	5	OPEN	ACQUIRE	LOCK

Cuando ya sepa lo que está haciendo la actividad, puede proceder en consonancia:

- Si la actividad está en cola y el usuario indica que la consulta tarda tanto que ya no le importan los resultados, o si considera que la consulta consume demasiados recursos, puede cancelarla.
- Si la actividad es importante y está en cola, plantéese cancelar algún otro trabajo menos importante que esté en ejecución actualmente (de forma que, cuando las actividades abandonen la cola, se reduzca la simultaneidad). O bien, el usuario puede darse por satisfecho al saber que el trabajo no se ha colgado y que espera el momento de ejecutarse.
- Si la actividad está a la espera de un bloqueo, puede utilizar el supervisor de instantáneas para investigar qué bloqueo espera la aplicación.
- Si la actividad está a la espera de un bloqueo mantenido por una actividad de menor prioridad, puede plantearse cancelar esa actividad.

También le puede resultar útil saber la sentencia de DML que está ejecutando esa actividad 5. Suponiendo que tenga un supervisor de sucesos de actividad activo, puede ejecutar el procedimiento WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS para capturar información sobre la sentencia de DML y otra información sobre la

actividad 5 mientras se está ejecutando. A diferencia del supervisor de sucesos de sentencias, el procedimiento `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` le permite capturar información sobre una consulta específica, en contraposición con todas las consultas que se ejecutan en ese momento. También puede obtener el texto de la sentencia si utiliza `MON_GET_ACTIVITY_DETAILS`.

Si decide que debe cancelar la actividad, puede utilizar la rutina `WLM_CANCEL_ACTIVITY` para cancelar la actividad sin tener que finalizar la aplicación que la emitió:

```
CALL WLM_CANCEL_ACTIVITY (1, 2, 5)
```

La aplicación que emitió la actividad recibirá un error `SQL4725N`. Cualquier aplicación que maneje códigos de SQL negativos puede manejar este código de SQL.

## Identificación de una actividad colgada a causa de una contención de bloqueo

Supongamos una situación en la que un usuario se queja de una aplicación que tarda demasiado. Supongamos también que tiene el nombre de la aplicación o el ID de autorización de la aplicación que tarda tanto en ejecutarse. Con esta información, puede utilizar el mandato `LIST APPLICATIONS` para obtener el descriptor de contexto de la aplicación. Suponiendo que el descriptor de contexto de la aplicación que ha devuelto el mandato `LIST APPLICATIONS` es 2, puede utilizar la función de tabla `WLM_GET_SERVICE_CLASS_AGENTS` para determinar qué agentes trabajan en esta actividad. Su consulta puede parecerse a la siguiente:

```
SELECT SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHANDLE,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(AGENT_TID),1,9) AS AGENT_TID,
       SUBSTR(AGENT_TYPE,1,11) AS AGENTTYPE,
       SUBSTR(EVENT_OBJECT,1,11) AS EVENTOBJECT,
       SUBSTR(REQUEST_TYPE,1,7) AS REQTYPE,
       SUBSTR(CHAR(UOW_ID),1,6) AS UOW_ID,
       SUBSTR(CHAR(ACTIVITY_ID),1,6) AS ACT_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS(' ', ' ', 2, -2)) AS SCDETAILS
ORDER BY APPHANDLE, MEMB, AGENT_TID
```

APPHANDLE	MEMB	AGENT_TID	AGENTTYPE	EVENTOBJECT	REQTYPE	UOW_ID	ACT_ID
2	0	1	COORDINATOR	REQUEST	OPEN	2	1
2	1	3	SUBAGENT	LOCK	-	2	1

El resultado indica que el agente 1 está esperando una respuesta remota. Al examinar el agente en el miembro remoto que trabaja en la misma actividad, el campo `EVENTOBJECT` indica que el agente está esperando a obtener un bloqueo.

El siguiente paso consiste en determinar quién es el propietario del bloqueo. Puede obtener esta información si activa los conmutadores del supervisor y utiliza la función de tabla del supervisor de instantáneas, como se muestra en el ejemplo siguiente:

```
SELECT AGENT_ID AS WAITING_FOR_LOCK,
       SUBSTR(APPL_ID_HOLDING_LK,1,40) AS HOLDING_LOCK,
       CAST(LOCK_MODE_REQUESTED AS SMALLINT) AS WANTED,
       CAST(LOCK_MODE AS SMALLINT) AS HELD
FROM TABLE(SNAPSHOT_LOCKWAIT('SAMPLE',-1)) AS SLW
```

WAITING_FOR_LOCK	HOLDING_LOCK	WANTED	HELD
2	*LOCAL.DB2.060131021547	9	5

También puede determinar el propietario del bloqueo utilizando la secuencia de mandatos siguiente:

```
db2pd -db alias_base_datos -locks
db2pd -db alias_base_datos -transactions
```

Si desea cancelar la actividad que tarda tanto en ejecutarse, puede utilizar el procedimiento WLM\_CANCEL\_ACTIVITY. Si la finalización correcta de la aplicación que está tardando en ejecutarse es más importante que la finalización correcta de la aplicación propietaria del bloqueo, puede forzar la aplicación propietaria del bloqueo.

---

## Caso de ejemplo: Cómo cancelar actividades que han estado en cola durante más de una hora

Si utiliza los scripts de ejemplo aquí descritos, puede crear un procedimiento para cancelar las actividades que han estado en cola durante más de una hora. Además, se proporciona un script de ejemplo que puede utilizarse para planificar el procedimiento de cancelación de actividad en cola para que se ejecute cada 10 minutos utilizando el planificador de tareas administrativas de DB2.

El procedimiento de cancelación de actividades en cola también captura información sobre las actividades canceladas (si hay un supervisor de sucesos de actividades activo), y mantiene una pequeña tabla histórica de actividades canceladas. Estos dos componentes informativos son opcionales y los comentarios de este script de ejemplo indican dónde deben eliminarse las marcas de comentario de los componentes, si no son necesarios.

Las sentencias del procedimiento de ejemplo son actividades propiamente dichas y están sujetas al control del umbral (dependiendo de cómo estén configurados los umbrales en el sistema). Considere la posibilidad de ejecutar el procedimiento de cancelación de actividad en cola de ejemplo en una clase de servicio a la que no se haya aplicado ningún umbral de cola.

1. Copie el script de ejemplo siguiente, que crea el procedimiento para cancelar actividades que han estado en cola durante más de una hora, en un archivo que haya creado (por ejemplo, uno llamado `x.clp`):

```
--
Breve tabla histórica para rastrear las
-- actividades canceladas

CREATE TABLE SAMPLE.CANCELED_ACTIVITIES(
  APPLICATION_HANDLE BIGINT,
  UOW_ID BIGINT,
  ACTIVITY_ID BIGINT )@

-- Cancelar las actividades que hayan estado en cola
-- durante más de una hora

CREATE PROCEDURE SAMPLE.CANCEL_QUEUED_ACTIVITIES()
LANGUAGE SQL
BEGIN
  DECLARE APPHANDLE    BIGINT;
  DECLARE UOWID        BIGINT;
  DECLARE ACTIVITYID   BIGINT;
  DECLARE QUEUETIME    BIGINT;
  DECLARE AT_END       INT DEFAULT 0;
```

```

DECLARE QUEUEDAPPS CURSOR WITH HOLD FOR SELECT APPLICATION_HANDLE,
        UOW_ID, ACTIVITY_ID
        FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(NULL,-2)) AS T
        WHERE ACTIVITY_STATE = 'QUEUED' AND LOCAL_START_TIME IS NULL;

DECLARE QTIMECUR CURSOR FOR SELECT TIMESTAMPDIF(8, CHAR
        (CURRENT_TIMESTAMP - TIMESTAMP(VALUE)))
        FROM TABLE(WLM_GET_ACTIVITY_DETAILS(APPHANDLE ,
        UOWID , ACTIVITYID , -2)) AS T WHERE NAME = 'ENTRY_TIME';

DECLARE CONTINUE HANDLER FOR NOT FOUND
        SET AT_END = 1;

-- Pasar por alto errores de actividad no encontrada y de supervisor
-- de sucesos de actividades que no existe.
DECLARE CONTINUE HANDLER FOR SQLSTATE '5U035', SQLSTATE '01H53'
        BEGIN
        END;

-- Buscar todas las actividades que los umbrales de WLM
-- han puesto en cola donde (ACTIVITY_STATE = 'QUEUED')
OPEN QUEUEDAPPS;
FETCH QUEUEDAPPS INTO APPHANDLE, UOWID, ACTIVITYID;

WHILE AT_END = 0 DO

-- Ahora utilice la hora de entrada de la actividad para calcular el
        tiempo que ha pasado en cola.
-- La colocación en cola se produce antes de que una actividad empiece a ejecutarse,
-- por lo que el tiempo en cola se calcula de forma aproximada
-- como: hora actual menos hora de entrada
        OPEN QTIMECUR;
        FETCH QTIMECUR INTO QUEUETIME;
        CLOSE QTIMECUR;

        IF ( QUEUETIME >= 1 ) THEN

-- Opcional: inserte un registro en una tabla para registrar la
-- cancelación de la sentencia (para fines de supervisión, para
-- saber cuántas sentencias se cancelaron). Modifique este
-- registro según proceda para capturar más información, como el nombre
-- de la aplicación que emitió la consulta cancelada. Quite la marca de
-- comentario de estas dos líneas si no necesita la supervisión.
                INSERT INTO SAMPLE.CANCELED_ACTIVITIES VALUES ( APPHANDLE,
                        UOWID, ACTIVITYID );

-- Opcional: Enviar detalles sobre la actividad a cualquier supervisor
-- de sucesos de actividad antes de cancelar. Quite la marca de
-- comentario de esta línea si no le interesan los detalles de las
-- sentencias canceladas
                CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS( APPHANDLE, UOWID,
                        ACTIVITYID );

-- Cancelar la actividad
                CALL WLM_CANCEL_ACTIVITY( APPHANDLE, UOWID, ACTIVITYID );

-- Confirmación explícita necesaria para la sentencia de inserción anterior. El
-- planificador de tareas administrativas no llevará a cabo la confirmación. Quite
-- la marca de comentario de esta línea si se elimina la sentencia de inserción.
                COMMIT;

        END IF;

        FETCH QUEUEDAPPS INTO APPHANDLE, UOWID, ACTIVITYID;

END WHILE;

```

```
CLOSE QUEUEDAPPS;
```

```
END@
```

2. Cree el procedimiento de cancelación de actividades en cola ejecutando el script `x.clp` con el mandato siguiente:

```
db2 -td@ -f x.clp
```

3. Ejecute el procedimiento de cancelación de actividades en cola emitiendo este mandato:

```
db2 "call sample.cancel_queued_activities()"
```

Cualquier actividad que haya estado en cola durante más de una hora se cancelará.

4. El siguiente script de ejemplo planifica el procedimiento de cancelación de actividades en cola para que se ejecute cada 10 minutos utilizando el Planificador de tareas administrativas de DB2. Copie el script de ejemplo en un archivo de su creación (por ejemplo, uno llamado `y.clp`):

```
-----  
-- Habilitar planificador de tareas administrativas de DB2 si  
-- no estaba ya habilitado.  
-----  
  
!db2set DB2_ATS_ENABLE=YES@  
  
-----  
-- Crear espacio de tablas SYSTOOLSPACE.  
-- Habilitar si SYSTOOLSPACE todavía  
-- no existe en su base de datos.  
-----  
  
-- CREATE TABLESPACE SYSTOOLSPACE IN IBMCATGROUP MANAGED BY AUTOMATIC STORAGE  
-- EXTENTSIZE 4@  
  
-----  
-- Añadir una tarea para cancelar automáticamente  
-- actividades que hayan estado en cola  
-- más de una hora. La tarea se planifica  
-- para que se ejecute cada 10 minutos. Ajuste  
-- la planificación según proceda mediante  
-- el parámetro de entrada de planificación  
-- (especificado en formato cron).  
-----  
  
CALL SYSPROC.ADMIN_TASK_ADD(  
    'CANCEL ACTIVITIES QUEUED FOR MORE 1 HOUR',  
    NULL,  
    NULL,  
    NULL,  
    '* / 10 * * * *',  
    'SAMPLE',  
    'CANCEL_QUEUED_ACTIVITIES',  
    NULL,  
    NULL,  
    NULL )@
```

5. Planifique el procedimiento de cancelación de actividades en cola para que se ejecute cada 10 minutos mediante el script `y.clp` utilizando el mandato siguiente:

```
db2 -td@ -f y.clp
```

---

## Caso de ejemplo: Identificación de actividades con un coste estimado bajo y un tiempo de ejecución alto

En el ejemplo siguiente se muestra cómo puede utilizar clases de trabajo, conjuntos de acciones de trabajo, umbrales y la recopilación de actividades para identificar las actividades que tienen un coste estimado bajo pero un tiempo de ejecución alto. Esta situación podría indicar que el coste estimado (en timerons) no es exacto debido a la existencia de estadísticas de tabla y de índice obsoletas.

El primer paso consiste en crear un conjunto de clases de trabajo con una clase de trabajo que se utilizará para identificar las actividades que tienen un coste estimado bajo. Por ejemplo:

```
CREATE WORK CLASS SET WCS1  
(WORK CLASS SMALLDML WORK TYPE DML FOR TIMERONCOST FROM 0 TO 500)
```

A continuación, ha de crear un conjunto de acciones de trabajo de base de datos con una acción de trabajo que aplique un umbral de tiempo total de actividad a la clase de trabajo SMALLDML. La acción de umbral es CONTINUE, y la opción COLLECT ACTIVITY DATA se especifica para que una actividad que infrinja el umbral se envíe al supervisor de sucesos de actividades al completarse:

```
CREATE WORK ACTION SET WAS1 FOR DATABASE USING WORK CLASS SET WCS1  
(WORK ACTION WA1 ON WORK CLASS SMALLDML WHEN ACTIVITYTOTALTIME > 15 MINUTES  
COLLECT ACTIVITY DATA WITH DETAILS CONTINUE)
```

Para finalizar, ha de crear y activar un supervisor de sucesos de violaciones de umbral y un supervisor de sucesos de actividades:

```
CREATE EVENT MONITOR THVIOLATIONS FOR THRESHOLD VIOLATIONS WRITE TO TABLE  
SET EVENT MONITOR THVIOLATIONS STATE 1
```

```
CREATE EVENT MONITOR DB2ACTIVITIES FOR ACTIVITIES WRITE TO TABLE  
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

Ahora, cuando una actividad de DML con un coste estimado de menos de 500 timerons se ejecute durante más de 15 minutos, se grabará un registro de violación de umbral en el supervisor de sucesos THVIOLATIONS (para indicar que se ha infringido el umbral de tiempo total) y se recopilarán detalles acerca de la actividad de DML cuando la actividad se complete y se envíe al supervisor de sucesos DB2ACTIVITIES. Puede utilizar la información recopilada acerca de la actividad en el supervisor de sucesos DB2ACTIVITIES para realizar una investigación más exhaustiva. Por ejemplo, puede ejecutar la sentencia EXPLAIN en la consulta y examinar el plan de acceso. También debe considerar la carga del sistema y consultar la hora a la que se ha recopilado la actividad, pues una vida útil prolongada puede ser el resultado de una insuficiencia de recursos del sistema o de la colocación en cola de la actividad. La vida útil prolongada no indica necesariamente que las estadísticas estén obsoletas.

---

## Caso de ejemplo: Cancelación de todas las actividades que se ejecutan en una subclase de servicio

A continuación se presenta un ejemplo de un procedimiento almacenado que puede utilizarse para cancelar todas las actividades que se estén ejecutando actualmente en una subclase de servicio particular.

Para crear el procedimiento CANCELALL, ejecute los pasos siguientes:

1. Copie la sentencia siguiente CREATE PROCEDURE en un archivo (por ejemplo, cancelall.ddl):

```
CREATE PROCEDURE CANCELALL ( IN INSCID BIGINT )
    SPECIFIC CANCELALL
    LANGUAGE SQL
BEGIN
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE APPHNDL BIGINT;
    DECLARE UOWID INTEGER;
    DECLARE ACTIVITYID INTEGER;

    DECLARE C1 CURSOR FOR (SELECT APPLICATION_HANDLE,
        UOW_ID, ACTIVITY_ID
        FROM TABLE(SYSPROC.WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES
            ( NULL, -2 ))
        AS T WHERE T.SERVICE_CLASS_ID = INSCID);

    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    BEGIN
        END;
    OPEN C1;

    FETCH_LOOP:
    LOOP

        FETCH C1 INTO APPHNDL, UOWID, ACTIVITYID;

        IF (SQLSTATE <> '00000')
        THEN LEAVE FETCH_LOOP;
        END IF;

        CALL WLM_CANCEL_ACTIVITY( APPHNDL, UOWID, ACTIVITYID );

    END LOOP FETCH_LOOP;

END@
```

2. Ejecute el siguiente mandato CLP:

```
db2 -td@ -f cancelall.ddl
```

Una vez creado el procedimiento, ejecútelo (por ejemplo, cancelando todas las actividades de la subclase de servicio con un ID = 15) utilizando la sentencia siguiente:

```
CALL CANCELALL( 15 )
```

**Nota:** El procedimiento CANCELALL debe ejecutarse en una subclase de servicio diferente desde el destino que se pasa como entrada; de lo contrario el procedimiento se cancelará por sí solo.

---

## Caso de ejemplo: Desconexión de todas las aplicaciones correlacionadas o ejecución de actividades en una clase de servicio

A continuación se muestra un ejemplo de un procedimiento almacenado que puede utilizarse para desconectar (forzar) todas las aplicaciones que se correlacionan en una clase de servicio en particular o que actualmente ejecutan actividades en una clase de servicio concreta.

Para crear el procedimiento FORCEALLINSC, ejecute los pasos siguientes:

1. Copie la sentencia siguiente CREATE PROCEDURE en un archivo (por ejemplo, forceall.ddl):

```

CREATE PROCEDURE FORCEALLINSC ( IN INSCID BIGINT )
    SPECIFIC FORCEALLINSC
    LANGUAGE SQL
BEGIN
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE APPHNDL BIGINT;
    DECLARE UOWID INTEGER;
    DECLARE ACTIVITYID INTEGER;

    DECLARE C1 CURSOR FOR (SELECT APPLICATION_HANDLE
        FROM TABLE(SYSPROC.WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES
            ( NULL, -2 ))
        AS T WHERE T.SERVICE_CLASS_ID = INSCID);

    DECLARE C2 CURSOR FOR (SELECT APPLICATION_HANDLE
        FROM TABLE(SYSPROC.WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES
            (NULL,NULL, -2 ))
        AS T, SYSCAT.SERVICECLASSES AS S
        WHERE T.SERVICE_SUPERCLASS_NAME = S.PARENTSERVICECLASSNAME AND
        T.SERVICE_SUBCLASS_NAME = S.SERVICECLASSNAME AND
        S.SERVICECLASSID = INSCID);

    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    BEGIN
    END;

-- Primero fuerce las aplicaciones que tengan una actividad en ejecución en
-- la clase de servicio especificada

    OPEN C1;

    FETCH_LOOP:
    LOOP

    FETCH C1 INTO APPHNDL;

    IF (SQLSTATE <> '00000')
    THEN LEAVE FETCH_LOOP;
    END IF;

    CALL ADMIN_CMD( 'FORCE APPLICATION (' || CHAR(APPHNDL) || ')' );

    END LOOP FETCH_LOOP;

-- Luego fuerce las conexiones que se correlacionan con la clase de servicio,
-- pero que no tienen actualmente ninguna actividad en ejecución

    OPEN C2;

    FETCH_LOOP2:
    LOOP

    FETCH C2 INTO APPHNDL;

    IF (SQLSTATE <> '00000')
    THEN LEAVE FETCH_LOOP2;
    END IF;

    CALL ADMIN_CMD( 'FORCE APPLICATION (' || CHAR(APPHNDL) || ')' );

    END LOOP FETCH_LOOP2;

END@

```

2. Ejecute el siguiente mandato CLP:

```
db2 -td@ -f forceall.dd1
```

Una vez creado el procedimiento, ejecútelo (por ejemplo, desconectando todas las aplicaciones que están correlacionadas o que actualmente ejecutan actividades en una clase de servicio particular con el ID = 15) utilizando la sentencia siguiente:  
 CALL FORCEALLINSC( 15 )

**Nota:** El procedimiento FORCEALLINSC debe ejecutarse en una clase de servicio diferente del destino que se pasa como entrada; de lo contrario el procedimiento se cancelará automáticamente.

## Caso de ejemplo: Ajuste de una configuración de gestión de carga de trabajo de DB2 cuando hay información de planificación de capacidad disponible

Si ha realizado la planificación de la capacidad, debería disponer de información relacionada con los tipos de usuarios y con los tiempos de respuesta esperados de éstos. Puede utilizar esta información para construir, determinar la efectividad de la configuración de la gestión de carga de trabajo de DB2 y ajustar ésta.

Supongamos que ha realizado la planificación de la capacidad y que la información de la tabla siguiente representa los resultados de este ejercicio de determinación de los tipos de trabajo y de los objetivos de tiempo de respuesta:

Tabla 63. Resultados de la planificación de la capacidad

Tipo de trabajo	Aplicación	Objetivo	Importancia	Rendimiento esperado
Entrada de pedido	orderentryapp.exe	Obtener un tiempo de respuesta medio de < 1 segundo	Alta	10.000 (inserciones y actualizaciones) por día
Consultas de Business Intelligence	businessobjects.exe	Obtener un tiempo de respuesta medio de < 10 segundos	Alta	100 consultas por día
Proceso por lotes	batchapp.exe	Maximizar el rendimiento	Baja	5.000 actualizaciones por día
Otros	Todas las demás aplicaciones	Mejorar esfuerzos	Baja	100 actividades por día

En función de la información de la tabla anterior, puede crear tres clases de servicio (ORDER\_ENTRY\_SC, BI\_QUERIES\_SC y BATCH\_SC) y tres cargas de trabajo (ORDER\_ENTRY\_WL, BI\_QUERIES\_WL y BATCH\_WL) para asignar trabajo a las clases de servicio. Tras la creación de las clases de servicio y las cargas de trabajo, puede crear un supervisor de sucesos estadísticos para recopilar información de actividad agregada, como el histograma de vida útil de actividades de cada clase de servicio. Supongamos que los datos de la siguiente tabla comparan el promedio diario de las actividades de cada clase de servicio (calculado a partir del histograma de vida útil de actividades) con los volúmenes que se han previsto en el ejercicio de planificación de la capacidad:

Tabla 64. Actividades por día

Clase de servicio	Actividades previstas por día	Actividades reales por día
ORDER_ENTRY_SC	10.000	9.700
BI_QUERIES_SC	100	115
BATCH_SC	5.000	5.412

Tabla 64. Actividades por día (continuación)

Clase de servicio	Actividades previstas por día	Actividades reales por día
SYSDEFAULTUSERCLASS	100	85

Los datos que se observan indican que las estimaciones de la planificación de la capacidad eran precisas. Los datos de la tabla siguiente comparan los promedios de vida útil de actividades (obtenidos del histograma de vida útil de actividades) con los objetivos de tiempo de respuesta determinados durante la planificación de la capacidad y muestran qué actividades de la clase de servicio BI\_QUERIES\_SC no cumplen sus objetivos de tiempo de respuesta.

Tabla 65. Tiempos de respuesta

Clase de servicio	Objetivo de tiempo de respuesta	Promedio de vida útil real
ORDER_ENTRY_SC	< 1 segundo	0,8 segundos
BI_QUERIES_SC	< 10 segundos	30 segundos
BATCH_SC		2 segundos
SYSDEFAULTUSERCLASS		10 minutos

Con la gestión de carga de trabajo de DB2, puede utilizar distintos enfoques para solucionar el problema que representan las consultas de Business Intelligence que no cumplen sus objetivos de tiempo de respuesta:

- Limitación de la simultaneidad de clases de servicio menos importantes
- Establecimiento del gestor de carga de trabajo del sistema operativo para que proporcione menos recursos de procesador a las clases de servicio menos importantes
- Modificación de las prioridades del agente y del buscador de captación previa de E/S para las clases de servicio
- Utilización de cualquier combinación de los tres enfoques anteriores

Supongamos que el tiempo de procesador es el recurso que da lugar a que las consultas de Business Intelligence no cumplan sus objetivos. Supongamos también que utiliza el gestor de carga de trabajo del sistema operativo de modo que éste proporcione a la clase de servicio SYSDEFAULTUSERCLASS menos recursos de procesador que las demás clases de servicio. A continuación, puede capturar la información de actividad agregada correspondiente a un período de días para observar si los cambios realizados en la asignación de CPU proporcionan los resultados que espera. Los datos de la tabla siguiente muestran otra comparación entre los objetivos de tiempo de respuesta y los promedios de vida útil reales calculados a partir de los histogramas tras haber realizado los cambios en el gestor de carga de trabajo del sistema operativo. Ahora, todas las clases de servicio cumplen los objetivos de tiempo de respuesta y, con la reasignación del tiempo de procesador, las actividades de la clase de servicio SYSDEFAULTUSERCLASS han doblado sus tiempos de respuesta.

Tabla 66. Tiempos de respuesta tras la reconfiguración

Clase de servicio	Objetivo de tiempo de respuesta	Promedio de vida útil real
ORDER_ENTRY_SC	< 1 segundo	0,6 segundos
BI_QUERIES_SC	< 10 segundos	9,5 segundos
BATCH_SC		1,5 segundos
SYSDEFAULTUSERCLASS		20 minutos

---

## Caso de ejemplo: Ajuste de una configuración de la gestión de carga de trabajo de DB2 cuando no hay información de planificación de capacidad disponible

Puede utilizar las herramientas de la gestión de carga de trabajo de DB2 como ayuda para diseñar, supervisar y ajustar la configuración de la gestión de carga de trabajo, aunque no disponga de datos de análisis de la capacidad que pueda utilizar para diseñar la configuración.

Supongamos que inicialmente no sabe qué cargas de trabajo y qué clases de servicio han de crearse porque no dispone de completos conocimientos acerca de las cargas de trabajo existentes en el sistema o porque todavía no sabe qué cargas de trabajo son necesarias para obtener resultados de ejecución estables. Supongamos también que sabe que algunas aplicaciones tienen requisitos de tiempo de respuesta pero que todavía no sabe cuántas son las otras aplicaciones que compiten por la obtención de recursos con estas aplicaciones con requisitos de tiempo de respuesta críticos. Puede utilizar las funciones de la supervisión de la gestión de carga de trabajo para determinarlo.

Para realizar una configuración de la gestión de carga de trabajo de DB2 basándose en los datos de supervisión:

1. Clasifique las aplicaciones que sabe que son importantes. Debe aislar estas aplicaciones y asignar a éstas la parte adecuada de recursos del sistema.
2. Para el resto de la carga de trabajo, recopile estadísticas de las actividades que hacen un uso más intensivo de la carga de trabajo, pues estas actividades son las que tienen mayor impacto por actividad en el sistema.
3. Analice la información de la actividad que ha recopilado en el paso 2.
4. Repita los pasos 1 a 3 en la parte de la carga de trabajo que sigue sin clasificar. Repita este paso hasta que pueda determinar que el resto del trabajo no clasificado no merece clasificación.

En los apartados siguientes se proporciona información acerca de la realización de estos pasos:

### **Paso 1. Aislar las aplicaciones que se sabe que son importantes y asignar a éstas una parte adecuada de recursos**

Supongamos que tiene dos aplicaciones importantes de Business Intelligence, BI1 y BI2, y que necesita minimizar los tiempos de respuesta de estas aplicaciones. Puede crear cargas de trabajo para estas dos aplicaciones y correlacionar éstas con una clase de servicio denominada MOSTIMPORTANT para la que puede asignar recursos del sistema.

En el sistema operativo AIX, ha de utilizar AIX Workload Manager para crear una clase de servicio denominada MOSTIMPORTANT y asignar a esta clase de servicio un conjunto de recursos garantizado.

En el servidor de datos de DB2, ha de crear las clases de servicio y las cargas de trabajo necesarias:

```
CREATE SERVICE CLASS MOSTIMPORTANT OUTBOUND CORRELATOR 'MOSTIMPORTANT'  
CREATE WORKLOAD BI1WORKLOAD APPLNAME ('BI1') SERVICE CLASS MOSTIMPORTANT  
CREATE WORKLOAD BI2WORKLOAD APPLNAME ('BI2') SERVICE CLASS MOSTIMPORTANT
```

Para este ejemplo, supongamos que incluso tras haber justificado la carga de trabajo de las aplicaciones conocidas, existe una parte de la carga de trabajo del sistema que no se ha justificado. Por lo tanto, deberá completar sus conocimientos acerca de esta carga de trabajo y, posiblemente, controlarla.

## **Paso 2. Para el resto de la carga de trabajo no clasificada, recopile estadísticas de las actividades que hacen un uso más intensivo de la carga de trabajo**

Una actividad de larga ejecución tiene más impacto individual en el sistema que una actividad de corta ejecución porque la primera utiliza recursos del sistema durante un período de tiempo más prolongado. Sin embargo, la recopilación de información acerca de una actividad de larga ejecución no conlleva más actividad general que la que implicaría la recopilación de información acerca de una actividad de corta ejecución. Como resultado de ello, la mejor forma de recopilar información acerca de la parte más grande de la carga de trabajo consiste en recopilar primero la información de las actividades de más larga ejecución.

Para iniciar la recopilación de la información de actividad, primero decida a partir de qué vida útil de actividades ha de recopilarse información de actividad. Para realizar esta tarea simplemente ha de seleccionar una parte de las actividades no clasificadas que han de recopilarse, como el 30%, y observar el histograma de vida útil de actividades de esas actividades. Permita que el sistema se ejecute para que se actualicen las estadísticas y, a continuación, ejecute el procedimiento `WLM_COLLECT_STATS` para enviar las estadísticas a un supervisor de sucesos estadísticos activo.

Utilice la consulta siguiente para obtener el histograma de vida útil de actividades para la clase de servicio `SYSDEFAULTUSERCLASS` en forma de tabla que represente la parte de las actividades totales que se encuentran dentro de cada rango de vida útil. Esta consulta se ha escrito dando por supuesto que la base de datos no tiene varios miembros.

```
WITH TOTAL AS (
SELECT PARENTSERVICECLASSNAME,
      SERVICECLASSNAME,
      HIST.HISTOGRAM_TYPE,
      SUM(NUMBER_IN_BIN) AS NUMBER_IN_BIN
FROM HISTOGRAMBIN_DB2STATISTICS AS HIST,
      SYSCAT.SERVICECLASSES SC
WHERE
      HIST.SERVICE_CLASS_ID = SC.SERVICECLASSID
      AND HIST.TOP >= 0
      AND SC.PARENTSERVICECLASSNAME = 'SYSDEFAULTUSERCLASS'
      AND SC.SERVICECLASSNAME = 'SYSDEFAULTSUBCLASS'
      AND HIST.HISTOGRAM_TYPE = 'CoordActLifetime'
GROUP BY PARENTSERVICECLASSNAME, SERVICECLASSNAME, HISTOGRAM_TYPE)
SELECT CAST(CAST(TOP AS DOUBLE) / 60000 AS DECIMAL(14,3)) AS TOP_IN_MINUTES,
      CAST(100 * CAST(SUM(HIST.NUMBER_IN_BIN) AS DOUBLE) / TOTAL.NUMBER_IN_BIN AS DECIMAL(4,2))
      AS PERCENT_IN_BIN
FROM HISTOGRAMBIN_DB2STATISTICS AS HIST,
      SYSCAT.SERVICECLASSES SC,
      TOTAL
WHERE HIST.SERVICE_CLASS_ID = SC.SERVICECLASSID
      AND HIST.TOP >= 0
      AND TOTAL.NUMBER_IN_BIN > 0
      AND SC.PARENTSERVICECLASSNAME = 'SYSDEFAULTUSERCLASS'
      AND SC.SERVICECLASSNAME = 'SYSDEFAULTSUBCLASS'
      AND HIST.HISTOGRAM_TYPE = 'CoordActLifetime'
      AND TOTAL.PARENTSERVICECLASSNAME = SC.PARENTSERVICECLASSNAME
      AND TOTAL.SERVICECLASSNAME = SC.SERVICECLASSNAME
```

```

AND TOTAL.HISTOGRAM_TYPE = HIST.HISTOGRAM_TYPE
GROUP BY TOP, SC.PARENTSERVICECLASSNAME, SC.SERVICECLASSNAME, HIST.HISTOGRAM_TYPE, TOTAL.NUMBER_IN_BIN;

```

TOP_IN_MINUTES	PERCENT_IN_BIN
0,000	0,00
0,000	0,00
0,000	0,00
0,000	0,00
0,000	0,00
0,000	0,00
0,000	0,00
0,000	0,00
0,000	0,00
0,001	0,00
0,001	0,00
0,002	0,00
0,004	0,00
0,006	0,00
0,009	0,00
0,014	0,00
0,021	0,00
0,033	0,00
0,050	0,00
0,077	0,00
0,118	0,00
0,180	0,00
0,274	0,00
0,419	0,00
0,639	0,00
0,975	0,00
1,488	0,00
2,269	0,00
3,462	0,00
5,280	0,00
8,054	0,00
12,286	0,00
18,740	0,00
28,584	10,00
43,600	15,00
66,505	45,00
101,442	23,00
154,731	5,00
236,015	2,00
360,000	0,00

En la siguiente figura se muestran los resultados de la consulta anterior trazados en forma de gráfico:

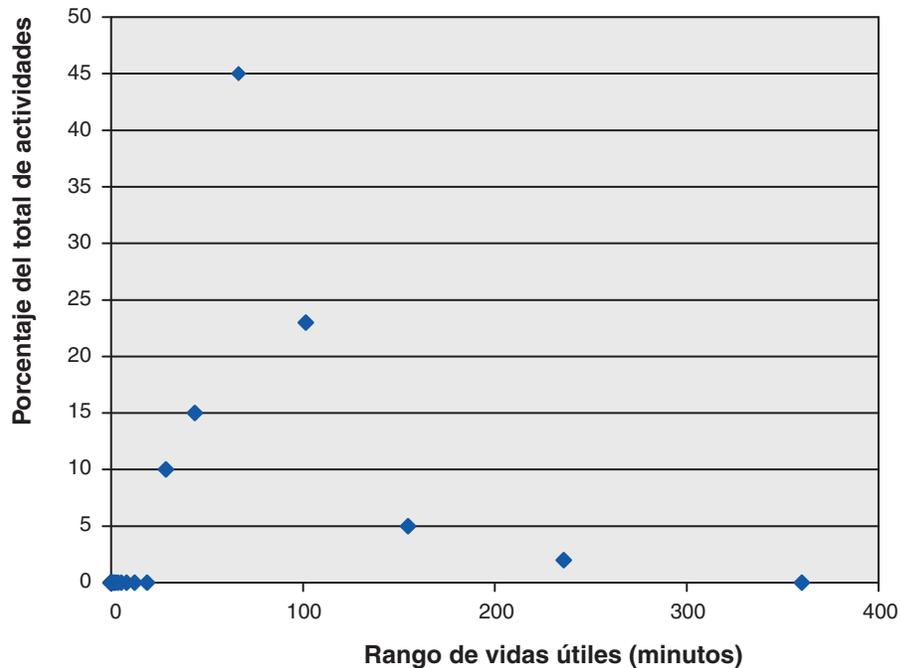


Figura 56. Histograma de vida útil de actividades de las actividades no clasificadas

En este ejemplo, el 30% de las actividades se encuentran dentro de un rango de vida útil de 101 minutos o superior. Para capturar la información relacionada con estas actividades, cree un umbral de vida útil de actividades de 100 minutos con las opciones CONTINUE y COLLECT ACTIVITY DATA, tal como se muestra en el ejemplo siguiente. Si se infringe este umbral, la información de actividad se envía a un supervisor de sucesos de actividades activo.

```
CREATE THRESHOLD COLLECTLONGESTRUNNING30PERCENT
FOR SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
ACTIVITIES ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 100 MINUTES COLLECT ACTIVITY DATA CONTINUE
```

Permita que el sistema se ejecute para que tenga lugar la recopilación de datos.

Si se da por supuesto que la actividad general de recopilación de información del 30% de las actividades de ejecución más larga es aceptable, puede permitir que la recopilación de datos continúe durante varias horas o durante varios días. Puede utilizar los datos recopilados para determinar qué usuarios y aplicaciones producen la ejecución más larga del 30% de las actividades de DML que siguen sin clasificar. Puede que estas actividades incluyan actividades con requisitos de tiempo críticos. Los resultados podrían sorprenderle, como observar que existen aplicaciones de baja prioridad que ejecutan un número significativo de grandes actividades. Cuando haya completado la recopilación y análisis de los datos, puede descartar el umbral.

### Paso 3. Analizar la información de las actividades que se ha recopilado en el paso anterior

Puede analizar la información de las actividades que ha recopilado en el paso anterior en función de la aplicación que las ha enviado. Puede especificar la consulta siguiente:

```

SELECT SUBSTR (APPL_NAME, 1,16) APPLICATION_NAME,
       AVG(TIMESTAMPDIFF(4, CHAR(TIME_COMPLETED - TIME_CREATED)))
       AS AVG_LIFETIME_MINUTES
       COUNT(*) AS ACTIVITY_COUNT
FROM ACTIVITY_DB2ACTIVITIES
GROUP BY APPL_NAME
ORDER BY APPL_NAME

```

```

APPLICATION_NAME  AVG_LIFETIME_MINUTES  ACTIVITY_COUNT
=====
MOSTLYSMALL1      120                    21
MOSTLYSMALL2      110                    15
UNIMPORTANTAPP    150                    10213

```

Un análisis de las actividades en función de la aplicación que realiza el envío muestra que gran número de las actividades de ejecución más larga las ha enviado la aplicación UNIMPORTANTAPP, que es una aplicación relativamente poco importante. Puede utilizar una carga de trabajo para aislar esta aplicación de las demás aplicaciones no clasificadas y correlacionar ésta con una clase de servicio denominada BESTEFFORT, que recibe recursos sólo cuando todas las demás actividades han satisfecho sus necesidades de recursos.

De acuerdo con los resultados anteriores, las demás aplicaciones de la clase de servicio por omisión parecen enviar pocas actividades grandes. Puede que el parezca oportuno y conveniente repetir el proceso de recopilación de las actividades que se ejecutan en la clase de servicio por omisión sin limitar la recopilación a las actividades de larga ejecución.

#### **Paso 4. Repetir los pasos 1 a 3 en la parte de la carga de trabajo que sigue sin clasificar hasta que pueda determinarse que el resto del trabajo no clasificado no merece clasificación**

Ahora que las dos aplicaciones importantes se ejecutan en la clase de servicio MOSTIMPORTANT y que la aplicación poco importante se ejecuta en la clase de servicio BESTEFFORT, mucho menos trabajo se ejecuta en la clase de servicio de usuario por omisión. En esta situación, posiblemente podrá dedicarse poco esfuerzo a recopilar información acerca de cada actividad de esta clase de servicio. Asimismo, puede que no necesite seguir subdividiendo el trabajo, por lo que puede concluir aquí. Supongamos que desea recopilar información acerca del resto de las actividades, en caso de que la carga de trabajo restante le haya sorprendido. Puede realizar esta tarea estableciendo COLLECT ACTIVITY DATA para la clase de servicio de usuario por omisión y creando un supervisor de sucesos de actividades:

```

ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
COLLECT ACTIVITY DATA ON COORDINATOR WITHOUT DETAILS

```

Permita que el sistema se ejecute para que tenga lugar la recopilación de datos. Puede analizar los resultados como en el paso 3.

```

SELECT SUBSTR (APPL_NAME,1,16) APPLICATION_NAME,
       AVG(TIMESTAMPDIFF(4, CHAR(TIME_COMPLETED - TIME_CREATED)))
       AS AVG_LIFETIME_MINUTES
       COUNT(*) AS ACTIVITY_COUNT
FROM ACTIVITY_DB2ACTIVITIES
GROUP BY APPL_NAME
ORDER BY APPL_NAME

```

```

APPLICATION_NAME  AVG_LIFETIME_MINUTES  ACTIVITY_COUNT
=====

```

MOSTLYSMALL1	5	1501
MOSTLYSMALL2	7	124
ONLYSMALL	2	10123

Los resultados muestran que la aplicación ONLYSMALL produce la mayor parte de las actividades no clasificadas. Puesto que esta aplicación no estaba incluida en los resultados que obtuvo cuando recopiló información acerca de las actividades más grandes, puede dar por supuesto que ONLYSMALL no ha producido ninguna consulta grande durante el período de recopilación de datos.

---

## Capítulo 8. Consulta

---

### Procedimientos y funciones de tabla

#### WLM\_CANCEL\_ACTIVITY - Cancelar una actividad

Este procedimiento cancela una actividad determinada. Si la cancelación tiene lugar, se devolverá un mensaje de error a la aplicación que ha enviado la actividad que se ha cancelado.

##### Sintaxis

```
►►—WLM_CANCEL_ACTIVITY—(—descriptor_contexto_aplicación—,—id_uow—,—id_actividad—)————►◄
```

El esquema es SYSPROC.

##### Parámetros del procedimiento

###### *descriptor\_contexto\_aplicación*

Argumento de entrada de tipo BIGINT que especifica el descriptor de contexto de aplicación cuya actividad se va a cancelar. Si el argumento es nulo, no se encontrará ninguna actividad y se devolverá SQL4702N con SQLSTATE 5U035.

###### *id\_uow*

Argumento de entrada de tipo INTEGER que especifica el ID de la unidad de trabajo de la actividad que se va a cancelar. Si el argumento es nulo, no se encontrará ninguna actividad y se devolverá SQL4702N con SQLSTATE 5U035.

###### *id\_actividad*

Argumento de entrada de tipo INTEGER que especifica el ID de actividad que identifica de forma exclusiva la actividad dentro de la unidad de trabajo que se va a cancelar. Si el argumento es nulo, no se encontrará ninguna actividad y se devolverá SQL4702N con SQLSTATE 5U035.

##### Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

##### Privilegio PUBLIC por omisión

Ninguna

##### Ejemplo

Un administrador puede utilizar la función de tabla WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para buscar el descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad de una

actividad. Para cancelar una actividad con el descriptor de contexto de aplicación 1, el ID de unidad de trabajo 2 y el ID de actividad 3:

```
CALL WLM_CANCEL_ACTIVITY(1, 2, 3)
```

### Notas de uso

- Si no se encuentra ninguna actividad, se devuelve un SQL4702N con SQLSTATE 5U035.
- Si la actividad no se puede cancelar porque no está en el estado correcto (no inicializada), se devuelve un SQL4703N (código de razón 1) con SQLSTATE 5U016.
- Si la actividad se cancela satisfactoriamente, se devuelve un SQL4725N con SQLSTATE 57014 a la aplicación cancelada.
- Si, en el momento de la cancelación, el coordinador está procesando una petición para otra actividad o está desocupado, la actividad se coloca en el estado CANCEL\_PENDING y se cancelará cuando el coordinador procese la siguiente petición.

## WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS - Recopilar información para supervisor de sucesos de actividades

El procedimiento WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS recopila información sobre una actividad especificada y la graba en el supervisor de sucesos de actividades activo.

Cuando se aplica este procedimiento a una actividad con actividades hijo, el procedimiento genera de forma recurrente un registro correspondiente a cada actividad hijo. Esta información se recopila y se envía al invocar el procedimiento; el procedimiento no espera hasta que la actividad padre finalice la ejecución. El registro de la actividad se marca como un registro parcial en el supervisor de sucesos.

### Sintaxis

```
►► WLM_CAPTURE_ACTIVITY_IN_PROGRESS (—descriptor_contexto_aplicación—, —————►  
► —id_uow—, —id_actividad—) —————►◄
```

El esquema es SYSPROC.

### Parámetros del procedimiento

Si no se especifica ninguno de los parámetros siguientes, no se encuentra ninguna actividad y se devuelve SQL4702N con SQLSTATE 5U035.

*descriptor\_contexto\_aplicación*

Argumento de entrada de tipo BIGINT que especifica el descriptor de contexto de aplicación cuya información de actividad se va a capturar.

*id\_uow*

Argumento de entrada de tipo INTEGER que especifica el ID de la unidad de trabajo de la actividad que se va a capturar.

*id\_actividad*

Argumento de entrada de tipo INTEGER que especifica el ID de actividad que identifica de forma exclusiva la actividad dentro de la unidad de trabajo cuya información se va a capturar.

## Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplo

Supongamos que un usuario se queja de que el procedimiento almacenado MYSHEMA.MYSLOWSTP parece estar ejecutándose más lentamente de lo habitual. El administrador desea investigar la causa de la lenta ejecución. Investigar mientras se está ejecutando el procedimiento almacenado puede resultar poco práctico, así que el administrador decide capturar la información de la actividad del procedimiento almacenado y cualquiera de las actividades anidadas en la misma.

Se ha activado un supervisor de sucesos para las actividades de DB2 llamadas DB2ACTIVITIES. El administrador utiliza la función WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES para buscar el descriptor de contexto de aplicación, el ID de unidad de trabajo y el ID de actividad para la llamada de este procedimiento almacenado. Suponiendo que la actividad se identifica mediante el descriptor de contexto de aplicación 1, el ID de unidad de trabajo 2 y el ID de actividad 3, ahora el administrador puede emitir la llamada a WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS del siguiente modo:

```
CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS(1,2,3)
```

Una vez completado el procedimiento, el administrador puede utilizar la siguiente función de tabla para ver dónde ha empleado su tiempo la actividad. La función recupera la información del supervisor de sucesos DB2ACTIVITIES.

```
CREATE FUNCTION SHOWCAPTUREDACTIVITY(APPHNDL BIGINT,  
                                     UOWID INTEGER,  
                                     ACTIVITYID INTEGER)  
  RETURNS TABLE (UOW_ID INTEGER, ACTIVITY_ID INTEGER, STMT_TEXT VARCHAR(40),  
                 LIFE_TIME DOUBLE)  
  LANGUAGE SQL  
  READS SQL DATA  
  NO EXTERNAL ACTION  
  DETERMINISTIC  
  RETURN WITH RAH (LEVEL, APPL_ID, PARENT_UOW_ID, PARENT_ACTIVITY_ID,  
                  UOW_ID, ACTIVITY_ID, STMT_TEXT, ACT_EXEC_TIME) AS  
(SELECT 1, ROOT.APPL_ID, ROOT.PARENT_UOW_ID,  
        ROOT.PARENT_ACTIVITY_ID, ROOT.UOW_ID, ROOT.ACTIVITY_ID,  
        ROOTSTMT.STMT_TEXT, ACT_EXEC_TIME  
 FROM ACTIVITY_DB2ACTIVITIES ROOT, ACTIVITYSTMT_DB2ACTIVITIES ROOTSTMT  
 WHERE ROOT.APPL_ID = ROOTSTMT.APPL_ID AND ROOT.AGENT_ID = APPHNDL  
       AND ROOT.UOW_ID = ROOTSTMT.UOW_ID AND ROOT.UOW_ID = UOWID  
       AND ROOT.ACTIVITY_ID = ROOTSTMT.ACTIVITY_ID AND ROOT.ACTIVITY_ID = ACTIVITYID  
 UNION ALL  
 SELECT PARENT.LEVEL +1, CHILD.APPL_ID, CHILD.PARENT_UOW_ID,  
        CHILD.PARENT_ACTIVITY_ID, CHILD.UOW_ID,  
        CHILD.ACTIVITY_ID, CHILDESTMT.STMT_TEXT, CHILD.ACT_EXEC_TIME
```

```

FROM RAH PARENT, ACTIVITY_DB2ACTIVITIES CHILD,
ACTIVITYSTMT_DB2ACTIVITIES CHILDSTMT
WHERE PARENT.APPL_ID = CHILD.APPL_ID AND
CHILD.APPL_ID = CHILDSTMT.APPL_ID AND
PARENT.UOW_ID = CHILD.PARENT_UOW_ID AND
CHILD.UOW_ID = CHILDSTMT.UOW_ID AND
PARENT.ACTIVITY_ID = CHILD.PARENT_ACTIVITY_ID AND
CHILD.ACTIVITY_ID = CHILDSTMT.ACTIVITY_ID AND
PARENT.LEVEL < 64
)
SELECT UOW_ID, ACTIVITY_ID, SUBSTR(STMT_TEXT,1,40),
ACT_EXEC_TIME AS
LIFE_TIME
FROM RAH

```

La siguiente consulta de ejemplo utiliza la función de tabla:

```

SELECT * FROM TABLE(SHOWCAPTUREDACTIVITY(1, 2, 3))
AS ACTS ORDER BY UOW_ID, ACTIVITY_ID

```

### Notas de uso

Si no hay ningún supervisor de sucesos de actividades activo, se devuelve un SQL1633W con SQLSTATE 01H53.

La información de actividad sólo se recopila en el miembro coordinador para la actividad.

## WLM\_COLLECT\_STATS - Coleccionar y restaurar estadísticas de gestión de carga de trabajo

El procedimiento WLM\_COLLECT\_STATS reúne estadísticas para clases de servicio, cargas de trabajo, clases de trabajo y colas de umbral y las graba en el supervisor de sucesos estadísticos. El procedimiento también restablece las estadísticas para clases de servicio, cargas de trabajo, clases de trabajo y colas de umbral. Si no hay ningún supervisor de sucesos estadísticos activos, el procedimiento sólo restaura las estadísticas.

### Sintaxis

```

▶▶ WLM_COLLECT_STATS ( [espera—, —indicación_fecha_hora_estadísticas] ) ▶▶

```

El esquema es SYSPROC.

### Parámetros del procedimiento

*wait*

Argumento de entrada opcional de tipo CHAR que especifica si este procedimiento vuelve inmediatamente después de iniciar una recopilación y restauración de estadísticas. Si se especifica 'Y', el procedimiento no volverá hasta que se hayan grabado y vaciado todas las estadísticas en las tablas del supervisor de sucesos estadísticos. De lo contrario, el procedimiento volverá inmediatamente después de iniciar una recopilación y restauración de estadísticas.

*indicación\_fecha\_hora\_estadísticas*

Argumento de salida opcional de tipo TIMESTAMP que devuelve el valor de indicación de fecha y hora para iniciar la recopilación de estadísticas.

## Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplos

*Ejemplo 1:* Llamar a WLM\_COLLECT\_STATS para iniciar la recopilación y restauración de estadísticas.

```
CALL WLM_COLLECT_STATS()
```

El ejemplo siguiente muestra los datos de salida de la consulta.

```
Estado de devolución = 0
```

*Ejemplo 2:* Llamar a WLM\_COLLECT\_STATS para recopilar y restaurar las estadísticas, pero no volver hasta que los datos se hayan grabado en las tablas del supervisor de sucesos estadísticos.

```
CALL WLM_COLLECT_STATS('Y', ::indicación_fecha_hora_colección)
```

El ejemplo siguiente muestra los datos de salida de la consulta.

```
Estado de devolución = 0
```

*Ejemplo 3:* Llamar a WLM\_COLLECT\_STATS para recopilar y restaurar estadísticas mientras otra llamada está en proceso.

```
CALL WLM_COLLECT_STATS()
```

El ejemplo siguiente muestra los datos de salida de la consulta.

```
SQL1632W La petición de reunir y restaurar las estadísticas se ha ignorado porque ya hay otra petición para reunir y restaurar estadísticas en proceso.
```

## Notas de uso

El procedimiento WLM\_COLLECT\_STATS realiza la misma tarea de recolección (enviar estadísticas al supervisor de sucesos estadísticos activo) y restauración de las operaciones que se da de forma automática en el intervalo definido por el parámetro de configuración de bases de datos **wlm\_collect\_int**.

Si llama al procedimiento mientras otra petición de recopilación y restauración está en proceso (por ejemplo, mientras otra invocación del procedimiento se está ejecutando o mientras se realiza la recopilación automática), se devuelve SQL1632W con SQLSTATE 01H53 y se pasa por alto la nueva petición.

En modalidad asíncrona, el procedimiento WLM\_COLLECT\_STATS sólo inicia el proceso de recopilación y restauración. Es posible que el procedimiento vuelva al que llama antes de que se graben todas las estadísticas en el supervisor de sucesos estadísticos activo. En función de la rapidez con que se realice la recolección y

restauración de las estadísticas, el procedimiento WLM\_COLLECT\_STATS (que es en sí una actividad y se incluirá en la estadística de actividades) puede que se cuente en las estadísticas del intervalo de recopilación previo o bien del nuevo intervalo de recopilación que se acaba de iniciar.

En modalidad síncrona, el procedimiento WLM\_COLLECT\_STATS no vuelve hasta que la recopilación de estadísticas ha finalizado y todas las estadísticas se han grabado en las tablas de los supervisores de sucesos estadísticos activos. La indicación de fecha y hora a la que se ha iniciado la reunión de estadísticas se devuelve a través del parámetro de salida *indicación\_fecha\_hora\_estadísticas*.

## WLM\_GET\_ACTIVITY\_DETAILS - Devolver información detallada sobre una actividad específica

Esta función devuelve información detallada sobre una actividad específica identificada por su descriptor de contexto de aplicación, su ID de unidad de trabajo y su ID de actividad. Esta información incluye detalles sobre cualquier umbral que haya violado la actividad.

**Nota:** Esta función de tabla ha quedado obsoleta y se ha sustituido por la Función de tabla MON\_GET\_ACTIVITY\_DETAILS .

Esta función devuelve estadísticas básicas de una o varias subclases de servicio.

### Sintaxis

```
►►WLM_GET_ACTIVITY_DETAILS(—descriptor_contexto_aplicación—,—id_uow—,—id_actividad—,—miembro—)
```

El esquema es SYSPROC.

### Parámetros de la función de tabla

#### *descriptor\_contexto\_aplicación*

Es un argumento de entrada de tipo BIGINT que especifica un descriptor de contexto de aplicación válido. Si el argumento es nulo, esta función no devuelve ninguna fila. Si el argumento es nulo, se devuelve un error SQL171N.

#### *id\_uow*

Argumento de entrada de tipo INTEGER que especifica un identificador de unidad de trabajo válido exclusivo dentro de la aplicación. Si el argumento es nulo, esta función no devuelve ninguna fila. Si el argumento es nulo, se devuelve un error SQL171N.

#### *id\_actividad*

Argumento de entrada de tipo INTEGER que especifica un ID de actividad válido exclusivo dentro de la unidad de trabajo. Si el argumento es nulo, esta función no devuelve ninguna fila. Si el argumento es nulo, se devuelve un error SQL171N.

#### *miembro*

Argumento de entrada de tipo INTEGER que especifica un número de miembro válido en la misma instancia que la base de datos conectada actualmente al llamar a esta función. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros de base de datos. Si se especifica un valor nulo, se establece -1 de forma implícita.

## Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplo

Se puede obtener información detallada sobre una actividad individual mediante la función de tabla WLM\_GET\_ACTIVITY\_DETAILS. Esta función de tabla devuelve información sobre la actividad como pares nombre-valor para cada miembro. Este ejemplo se limita a mostrar sólo un subconjunto de once miembros de los pares nombre-valor para cada miembro de una actividad identificada mediante el descriptor de contexto de aplicación 1, el ID de unidad de trabajo 1 y el ID de actividad 5. Para ver una lista completa de pares nombre-valor, consulte la Tabla 68 en la página 404 y la Tabla 69 en la página 406.

```
SELECT SUBSTR(CHAR(DBPARTITIONNUM),1,4) AS PART,
       SUBSTR(NAME, 1, 20) AS NAME,
       SUBSTR(VALUE, 1, 30) AS VALUE
FROM TABLE(WLM_GET_ACTIVITY_DETAILS(1, 1, 5, -2)) AS ACTDETAIL
WHERE NAME IN ('APPLICATION_HANDLE',
              'COORD_PARTITION_NUM',
              'LOCAL_START_TIME',
              'UOW_ID',
              'ACTIVITY_ID',
              'PARENT_UOW_ID',
              'PARENT_ACTIVITY_ID',
              'ACTIVITY_TYPE',
              'NESTING_LEVEL',
              'INVOCATION_ID',
              'ROUTINE_ID')
ORDER BY PART
```

El ejemplo siguiente muestra los datos de salida de la consulta.

PART	NAME	VALUE
0	APPLICATION_HANDLE	1
0	COORD_PARTITION_NUM	0
0	LOCAL_START_TIME	2005-11-25-18.52.49.343000
0	UOW_ID	1
0	ACTIVITY_ID	5
0	PARENT_UOW_ID	1
0	PARENT_ACTIVITY_ID	3
0	ACTIVITY_TYPE	READ_DML
0	NESTING_LEVEL	0
0	INVOCATION_ID	1
0	ROUTINE_ID	0
1	APPLICATION_HANDLE	1
1	COORD_PARTITION_NUM	0
1	LOCAL_START_TIME	2005-11-25-18.52.49.598000
1	UOW_ID	1
1	ACTIVITY_ID	5
1	PARENT_UOW_ID	

```

1 PARENT_ACTIVITY_ID
1 ACTIVITY_TYPE      READ_DML
1 NESTING_LEVEL      0
1 INVOCATION_ID      1
1 ROUTINE_ID          0

```

## Nota sobre uso

El valor QUEUED de ACTIVITY\_STATE significa que la actividad de coordinación ha realizado un RPC sobre el miembro de catálogo para obtener tíquets de umbral y aún no se ha recibido una respuesta. Este estado puede indicar que WLM ha colocado la actividad en cola o, durante breves periodos de tiempo, puede indicar que la actividad está en proceso de obtener sus tíquets. Para obtener una imagen más precisa de si la actividad se está colocando realmente en cola o no, se puede determinar qué agente está trabajando en la actividad (mediante la función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS) y buscar si el event\_object del agente de miembro de catálogo tiene el valor WLM\_QUEUE.

## Información devuelta

Tabla 67. Información devuelta para WLM\_GET\_ACTIVITY\_DETAILS

Nombre de columna	Tipo de datos	Descripción
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
NAME	VARCHAR(256)	Nombre del elemento. Consulte la Tabla 68 y Tabla 69 en la página 406 para ver los valores posibles.
VALUE	VARCHAR(1024)	Valores de elemento. Consulte la Tabla 68 y Tabla 69 en la página 406 para ver los valores posibles.

Tabla 68. Elementos devueltos

Nombre de elemento	Descripción
ACTIVITY_ID	activity_id - Elemento de supervisor de ID de actividad
ACTIVITY_STATE	activity_state - Elemento de supervisor de estado de actividad
ACTIVITY_TYPE	activity_type - Elemento de supervisor de tipo de actividad
APPLICATION_HANDLE	application_handle - Elemento de supervisor de descriptor de contexto de aplicación
COORD_PARTITION_NUM	coord_partition_num - Elemento de supervisor de número de partición coordinadora
DATABASE_WORK_ACTION_SET_ID	Si esta actividad se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a la base de datos, esta columna contiene el ID del conjunto de acciones de trabajo. Esta columna contiene 0 si la actividad aún no se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a la base de datos.
DATABASE_WORK_CLASS_ID	Si esta actividad se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a la base de datos, esta columna contiene el ID de la clase de trabajo de esta actividad. Esta columna contiene 0 si la actividad aún no se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a la base de datos.

Tabla 68. Elementos devueltos (continuación)

Nombre de elemento	Descripción
EFFECTIVE_ISOLATION	effective_isolation - Elemento de supervisor de aislamiento efectivo
EFFECTIVE_LOCK_TIMEOUT	effective_lock_timeout - Elemento de supervisor de tiempo de espera de bloqueo efectivo
EFFECTIVE_QUERY_DEGREE	effective_query_degree - Elemento de supervisor de grado de consulta efectivo
ENTRY_TIME	entry_time - Elemento de supervisor de hora de entrada
INVOCATION_ID	invocation_id - Elemento de supervisor de ID de invocación
LAST_REFERENCE_TIME	last_reference_time - Elemento de supervisor de hora de última referencia
LOCAL_START_TIME	local_start_time - Elemento de supervisor de hora de inicio local
NESTING_LEVEL	nesting_level - Elemento de supervisor de nivel de anidamiento
PACKAGE_NAME	package_name - Elemento de supervisor de nombre de paquete.
PACKAGE_SCHEMA	package_schema - Elemento de supervisor de esquema de paquete.
PACKAGE_VERSION_ID	package_version_id - Elemento de supervisor de versión del paquete
PARENT_ACTIVITY_ID	parent_activity_id - Elemento de supervisor de ID de actividad padre
PARENT_UOW_ID	Identificador exclusivo de unidad de trabajo dentro de una aplicación. Se refiere a la unidad de trabajo original en la que se ha iniciado esta actividad padre. Devuelve una serie vacía si la actividad no tiene actividad padre o si está en un miembro remoto.
QUERY_COST_ESTIMATE	query_cost_estimate - Elemento de supervisor de estimación del coste de la consulta
ROUTINE_ID	routine_id - Elemento de supervisor de ID de rutina
ROWS_FETCHED	rows_fetched - Elemento de supervisor de filas captadas
ROWS_MODIFIED	rows_modified - Elemento de supervisor de filas modificadas
SECTION_NUMBER	section_number - Elemento de supervisor de número de sección
SERVICE_CLASS_ID	service_class_id - Elemento de supervisor de ID de clase de servicio
SERVICE_CLASS_WORK_ACTION_SET_ID	Si esta actividad se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a una clase de servicio, esta columna contiene el ID del conjunto de acciones de trabajo. Esta columna contiene 0 si la actividad aún no se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a una clase de servicio.

Tabla 68. Elementos devueltos (continuación)

Nombre de elemento	Descripción
SERVICE_CLASS_WORK_CLASS_ID	Si esta actividad se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a una clase de servicio, esta columna contiene el ID de la clase de trabajo de esta actividad. Esta columna contiene 0 si la actividad aún no se ha correlacionado con un conjunto de acciones de trabajo que se ha aplicado a una clase de servicio.
STMT_PKG_CACHE_ID	stmt_pkgcache_id - Elemento de supervisor de identificador de antememoria del paquete de la sentencia
STMT_TEXT	stmt_text - Elemento de supervisor de texto de sentencia de SQL
SYSTEM_CPU_TIME	system_cpu_time - Elemento de supervisor de tiempo de CPU del sistema
UOW_ID	uow_id - Elemento de supervisor de ID de unidad de trabajo
USER_CPU_TIME	user_cpu_time - Elemento de supervisor de tiempo de CPU del usuario
UTILITY_ID	utility_id - Elemento de supervisor de ID de programa de utilidad

**Importante:** La función de tabla WLM\_GET\_ACTIVITY\_DETAILS muestra únicamente los umbrales que se están aplicando actualmente a una actividad.

Se devuelven los elementos siguientes únicamente si los umbrales correspondientes se aplican a la actividad.

Tabla 69. Elementos devueltos si se aplica

Nombre de elemento	Descripción
ACTIVITYTOTALTIME_THRESHOLD_ID	activitytotaltime_threshold_id - Elemento de supervisor de ID de umbral de tiempo total de actividad
ACTIVITYTOTALTIME_THRESHOLD_VALUE	activitytotaltime_threshold_value - Elemento de supervisor de valor de umbral de tiempo total de actividad
ACTIVITYTOTALTIME_THRESHOLD_VIOLATED	activitytotaltime_threshold_violated - Elemento de supervisor de umbral de tiempo total de actividad violado
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_ID	concurrentdbcoordactivities_db_threshold_id - Elemento de supervisor de ID de umbral de base de datos de las actividades del coordinador de bases de datos simultáneas
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_QUEUED	concurrentdbcoordactivities_db_threshold_queued - Elemento de supervisor de umbral de base de datos de actividades del coordinador de bases de datos simultáneas en cola
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VALUE	concurrentdbcoordactivities_db_threshold_value - Elemento de supervisor de valor de umbral de base de datos de las actividades del coordinador de bases de datos simultáneas

Tabla 69. Elementos devueltos si se aplica (continuación)

Nombre de elemento	Descripción
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VIOLATED	concurrentdbcoordactivities_db_threshold_violated - Elemento de supervisor de umbral de base de datos de las actividades del coordinador de bases de datos simultáneas violado
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_ID	concurrentdbcoordactivities_subclass_threshold_id - Elemento de supervisor de ID de umbral de la subclase de servicio de actividades del coordinador de bases de datos simultáneas
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_QUEUED	'Yes' indica que el umbral CONCURRENTDBCOORDACTIVITIES_SUBCLASS ha colocado la actividad en cola. 'No' indica que la actividad no se ha colocado en cola.
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_VALUE	Elemento de supervisor concurrentdbcoordactivities_subclass Elemento de supervisor _threshold_value - Valor de umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_VIOLATED	'Yes' indica que la actividad ha violado el umbral CONCURRENTDBCOORDACTIVITIES_SUBCLASS. No' indica que la actividad todavía no ha violado el umbral.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_ID	El ID del umbral CONCURRENTDBCOORDACTIVITIES_SUPERCLASS que se aplicó a la actividad.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_QUEUED	'Yes' indica que el umbral CONCURRENTDBCOORDACTIVITIES_SUPERCLASS ha puesto la actividad en cola. No' indica que la actividad no se ha colocado en cola.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_VALUE	El límite superior del umbral CONCURRENTDBCOORDACTIVITIES_SUPERCLASS que se aplicó a la actividad.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_VIOLATED	'Yes' indica que la actividad ha violado el umbral CONCURRENTDBCOORDACTIVITIES_SUPERCLASS ha puesto la actividad en cola. No' indica que la actividad todavía no ha violado el umbral.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_ID	El ID del umbral CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET que se aplicó a la actividad.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_QUEUED	'Yes' indica que el umbral CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET puso a la actividad en cola. 'No' indica que la actividad no se ha colocado en cola.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VALUE	El límite superior del umbral CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET que se aplicó a la actividad.

Tabla 69. Elementos devueltos si se aplica (continuación)

Nombre de elemento	Descripción
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VIOLATED	'Yes' indica que la actividad ha violado el umbral CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET. 'No' indica que la actividad todavía no ha violado el umbral.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_ID	El ID del umbral CONCURRENTWORKLOADACTIVITIES que se aplicó a la actividad.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VALUE	El límite superior del umbral CONCURRENTWORKLOADACTIVITIES que se aplicó a la actividad.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VIOLATED	'Yes' indica que la actividad ha violado el umbral CONCURRENTWORKLOADACTIVITIES. 'No' indica que la actividad todavía no ha violado el umbral.
ESTIMATEDSQLCOST_THRESHOLD_ID	estimatedsqlcost_threshold_id - Elemento de supervisor de ID de umbral de coste de SQL estimado
ESTIMATEDSQLCOST_THRESHOLD_VALUE	estimatedsqlcost_threshold_value - Elemento de supervisor de valor de umbral de coste de SQL estimado
ESTIMATEDSQLCOST_THRESHOLD_VIOLATED	estimatedsqlcost_threshold_violated - Elemento de supervisor de umbral de coste de SQL estimado violado
SQLROWSRETURNED_THRESHOLD_ID	sqlrowsreturned_threshold_id - Elemento de supervisor de ID de umbral de filas de SQL leídas devueltas
SQLROWSRETURNED_THRESHOLD_VALUE	sqlrowsreturned_threshold_value - Elemento de supervisor de valor de umbral de filas de SQL leídas devueltas
SQLROWSRETURNED_THRESHOLD_VIOLATED	sqlrowsreturned_threshold_violated - Elemento de supervisor de umbral de filas de SQL leídas devueltas violado
SQLTEMPSPACE_THRESHOLD_ID	sqltempespace_threshold_id - Elemento de supervisor de ID de umbral de espacio temporal de SQL
SQLTEMPSPACE_THRESHOLD_VALUE	sqltempespace_threshold_value - Elemento de supervisor de valor de umbral de espacio temporal de SQL
SQLTEMPSPACE_THRESHOLD_VIOLATED	sqltempespace_threshold_violated - Elemento de supervisor de umbral de espacio temporal de SQL violado

## Función de tabla WLM\_GET\_QUEUE\_STATS - Devolver estadísticas de cola de umbral

La función WLM\_GET\_QUEUE\_STATS devuelve estadísticas básicas de una o más colas de umbral en todos los miembros activos. Esta función devuelve una fila de estadísticas para cada cola de umbral.

## Sintaxis

```
► WLM_GET_QUEUE_STATS(—predicado_umbral—,—dominio_umbral—,—  
► nombre_umbral—,—id_umbral—)
```

El esquema es SYSPROC.

## Parámetros de la función de tabla

### *predicado\_umbral*

Argumento de entrada de tipo VARCHAR(27) que especifica un predicado de umbral. Los valores posibles son los siguientes:

*CONCDBC*

Umbral de actividades de coordinador de base de datos simultáneo

*DBCONN*

Umbral total de conexiones de miembro de base de datos

*SCCONN*

Umbral total de conexiones de partición de clase de miembro

Si el argumento es nulo o la serie está vacía, se devuelven datos para todos los umbrales que cumplan los demás criterios.

Los valores de *predicado\_umbral* coinciden con los de la columna THRESHOLDPREDICATE en la vista SYSCAT.THRESHOLDS.

### *dominio\_umbral*

Argumento de entrada de tipo VARCHAR(18) que especifica un dominio de umbral. Los valores posibles son los siguientes:

*DB* Base de datos

*SB* Subclase de servicio

*SP* Superclase de servicio

*WA* Conjunto de acciones de trabajo

Si el argumento es nulo o la serie está vacía, se devuelven datos para todos los umbrales que cumplan los demás criterios.

Los valores de *dominio\_umbral* coinciden con los de la columna DOMAIN en la vista SYSCAT.THRESHOLDS.

### *nombre\_umbral*

Argumento de entrada de tipo VARCHAR(128) que especifica un nombre de umbral. Si el argumento es nulo o la serie está vacía, se devuelven datos para todos los umbrales que cumplan los demás criterios. Los valores de *nombre\_umbral* coinciden con los de la columna THRESHOLDNAME en la vista SYSCAT.THRESHOLDS.

### *id\_umbral*

Argumento de entrada de tipo INTEGER que especifica un ID de umbral. Si el argumento es -1, se devuelven datos para todos los umbrales que cumplan los demás criterios. Los valores de *id\_umbral* coinciden con los de la columna THRESHOLDID en la vista SYSCAT.THRESHOLDS.

## Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplo

La siguiente consulta muestra las estadísticas básicas correspondientes a todas las colas de un sistema, en todos los miembros:

```
SELECT substr(THRESHOLD_NAME, 1, 6) THRESHNAME,
       THRESHOLD_PREDICATE,
       THRESHOLD_DOMAIN,
       MEMBER MEMB,
       QUEUE_SIZE_TOP,
       QUEUE_TIME_TOTAL,
       QUEUE_ASSIGNMENTS_TOTAL QUEUE_ASSIGN
FROM table(WLM_GET_QUEUE_STATS('', '', '', -1)) as QSTATS
```

Un ejemplo de la salida es el siguiente:

```
THRESHNAME THRESHOLD_PREDICATE THRESHOLD_DOMAIN ...
-----
LIMIT1     CONCDBC                      DB                      ...
LIMIT2     SCCONN                       SP                      ...
LIMIT3     DBCONN                       DB                      ...
... MEMB QUEUE_SIZE_TOP QUEUE_TIME_TOTAL QUEUE_ASSIGN
... -----
... 0         12          1238540          734
... 0         4           741249           24
... 0         7           412785           128
```

## Nota sobre uso

La función no realiza ninguna agregación de datos entre colas (de un miembro) ni entre miembros (para una o varias colas). Sin embargo, puede utilizar consultas de SQL para agregar datos, como se muestra en el ejemplo anterior.

## Información devuelta

Tabla 70. Información devuelta para WLM\_GET\_QUEUE\_STATS

Nombre de columna	Tipo de datos	Descripción
THRESHOLD_PREDICATE	VARCHAR(27)	threshold_predicate - Elemento de supervisor de predicado de umbral
THRESHOLD_DOMAIN	VARCHAR(18)	threshold_domain - Elemento de supervisor de dominio de umbral
THRESHOLD_NAME	VARCHAR(128)	threshold_name - Elemento de supervisor de nombre de umbral

Tabla 70. Información devuelta para WLM\_GET\_QUEUE\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
THRESHOLD_ID	INTEGER	thresholdid - Elemento de supervisor de ID de umbral
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Elemento de supervisor de nombre de superclase de servicio
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Elemento de supervisor de nombre de subclase de servicio
WORK_ACTION_SET_NAME	VARCHAR(128)	work_action_set_name - Elemento de supervisor de nombre de conjunto de acciones de trabajo
WORK_CLASS_NAME	VARCHAR(128)	work_class_name - Elemento de supervisor de clase de trabajo
WORKLOAD_NAME	VARCHAR(128)	workload_name - Elemento de supervisor de nombre de carga de trabajo
LAST_RESET	TIMESTAMP	last_reset - elemento de supervisor de indicación de fecha y hora de último restablecimiento
QUEUE_SIZE_TOP	INTEGER	queue_size_top - Elemento de supervisor de máximo tamaño de cola
QUEUE_TIME_TOTAL	BIGINT	queue_time_total - Elemento de supervisor de tiempo total en cola
QUEUE_ASSIGNMENTS_TOTAL	BIGINT	queue_assignments_total - Elemento de supervisor de total de asignaciones de cola
QUEUE_SIZE_CURRENT	INTEGER	Número de conexiones o actividades en la cola.
QUEUE_TIME_LATEST	BIGINT	Tiempo transcurrido en la cola por la última conexión o actividad para dejar la cola. Las unidades son milisegundos.
QUEUE_EXIT_TIME_LATEST	TIMESTAMP	Hora en que la última conexión o actividad dejó la cola.
THRESHOLD_CURRENT_CONCURRENCY	INTEGER	Número de conexiones o actividades que se están ejecutando actualmente según el umbral.
THRESHOLD_MAX_CONCURRENCY	INTEGER	Número máximo de conexiones o actividades que se está ejecutando actualmente con el permiso del umbral.
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

## **Función de tabla WLM\_GET\_SERVICE\_CLASS\_AGENTS - Listar los agentes que se ejecutan en una clase de servicio**

La función WLM\_GET\_SERVICE\_CLASS\_AGENTS devuelve la lista de agentes, procesos en modalidad delimitada (procesos db2fmp) y entidades del sistema de un miembro especificado que se están ejecutando en una clase de servicio dada o en nombre de una aplicación dada. Las entidades del sistema son procesos y hebras sin agente, como limpiadores de páginas y captadores previos.

Consulte la Tabla 71 en la página 414 para ver una lista completa de la información que se puede devolver.

### **Sintaxis**

```
►—WLM_GET_SERVICE_CLASS_AGENTS—(—nombre_superclase_servicio—,—————►  
►—nombre_subclase_servicio—,—descriptor_contexto_aplicación—,—miembro—)————►
```

El esquema es SYSPROC.

### **Parámetros de la función de tabla**

#### *nombre\_superclase\_servicio*

Argumento de entrada de tipo VARCHAR(128) que especifica el nombre de una superclase de servicio de la base de datos actualmente conectada. Si el argumento es nulo o una serie vacía, se recuperan datos correspondientes a todas las superclases de la base de datos.

#### *nombre\_subclase\_servicio*

Argumento de entrada de tipo VARCHAR(128) que hace referencia a una determinada subclase dentro de una superclase. Si el argumento es nulo o una serie vacía, se recuperan datos correspondientes a todas las subclases de la base de datos.

#### *descriptor\_contexto\_aplicación*

Argumento de entrada de tipo BIGINT que especifica el descriptor de contexto de aplicación para el que se debe devolver información del agente. Si el argumento es nulo, se recuperan datos correspondientes a todas las aplicaciones de la base de datos. El descriptor de contexto de aplicación 0 sólo devolverá entidades del sistema.

#### *miembro*

Argumento de entrada de tipo INTEGER que especifica el número de miembro en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros de base de datos. Si se especifica un valor nulo, se establece -1 de forma implícita.

### **Autorización**

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

### Ejemplo 1

La consulta siguiente devuelve una lista de agentes asociados al descriptor de contexto de aplicación 1 para todos los miembros de la base de datos. Puede determinar el descriptor de contexto de aplicación mediante el mandato **LIST APPLICATIONS** o la función de tabla

WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES.

```
SELECT SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHANDLE,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(AGENT_TID),1,9) AS AGENT_TID,
       SUBSTR(AGENT_TYPE,1,11) AS AGENTTYPE,
       SUBSTR(AGENT_STATE,1,10) AS AGENTSTATE,
       SUBSTR(REQUEST_TYPE,1,12) AS REQTYPE,
       SUBSTR(CHAR(UOW_ID),1,6) AS UOW_ID,
       SUBSTR(CHAR(ACTIVITY_ID),1,6) AS ACT_ID
FROM TABLE(WLM_GET_SERVICE_CLASS_AGENTS(CAST(NULL AS VARCHAR(128)),
    CAST(NULL AS VARCHAR(128)), 1, -2)) AS SCDETAILS
ORDER BY APPHANDLE, MEMB, AGENT_TID
```

Un ejemplo de la salida es el siguiente:

APPHANDLE	MEMB	AGENT_TID	AGENTTYPE	AGENTSTATE	REQTYPE	UOW_ID	ACT_ID
1	0	3	COORDINATOR	ACTIVE	FETCH	1	5
1	0	4	SUBAGENT	ACTIVE	SUBSECTION:1	1	5
1	1	2	SUBAGENT	ACTIVE	SUBSECTION:2	1	5

La salida muestra un agente coordinador y un subagente en el miembro 0, así como un subagente en el miembro 1 que operan en nombre de una actividad con el ID de UOW 1 y el ID de actividad 5. La columna AGENTTYPE con un valor de COORDINATOR tiene un valor FETCH para la columna REQTYPE (que indica el tipo de petición principal o inicial). Esto significa que el tipo de petición es una petición de captación para el agente coordinador.

### Ejemplo 2

La consulta siguiente determina el bloqueo en el que espera un agente:

```
db2 select event_object, event_type, event_state, varchar(event_object_name, 30)
as event_object_name
from table(WLM_GET_SERVICE_CLASS_AGENTS('','cast(NULL as bigint), -1)) as t
```

Un ejemplo de la salida es el siguiente:

EVENT_OBJECT	EVENT_TYPE	EVENT_STATE	EVENT_OBJECT_NAME
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	WAIT	IDLE	-
LOCK	ACQUIRE	IDLE	020005000000000000000000000054
ROUTINE	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-

```

REQUEST      PROCESS      EXECUTING    -

```

21 registro(s) seleccionado(s).

Utilizando la misma consulta posteriormente, determina qué umbral de WLM ha puesto en cola un agente:

```

EVENT_OBJECT  EVENT_TYPE  EVENT_STATE  EVENT_OBJECT_NAME
-----
REQUEST      PROCESS      EXECUTING    -
WLM_QUEUE    WAIT         IDLE         MYCONCDBCOORDTH
ROUTINE      PROCESS      EXECUTING    -
REQUEST      PROCESS      EXECUTING    -

```

21 registro(s) seleccionado(s).

### Nota sobre uso

Los parámetros tienen el efecto de sumarse (operación AND). Es decir, si se especifican parámetros de entrada conflictivos como una superclase de servicio SUP\_A y una subclase de servicio SUB\_B de modo que SUB\_B no es una subclase de SUP\_A, no se devuelve ninguna fila.

### Información devuelta

Tabla 71. Información devuelta por WLM\_GET\_SERVICE\_CLASS\_AGENTS

Nombre de columna	Tipo de datos	Descripción
SERVICE_SUPERCLASS_NAME	VARCHAR (128)	service_superclass_name - Elemento de supervisor de nombre de superclase de servicio
SERVICE_SUBCLASS_NAME	VARCHAR (128)	service_subclass_name - Elemento de supervisor de nombre de subclase de servicio
APPLICATION_HANDLE	BIGINT	application_handle - Elemento de supervisor de descriptor de contexto de aplicación
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos

Tabla 71. Información devuelta por WLM\_GET\_SERVICE\_CLASS\_AGENTS (continuación)

Nombre de columna	Tipo de datos	Descripción
ENTITY	VARCHAR (32)	Uno de los valores siguientes: <ul style="list-style-type: none"> <li>• Si el tipo de entidad es un agente, el valor es db2agent.</li> <li>• Si el tipo de entidad es un proceso en modalidad delimitada, el valor es db2fmp (<i>pid</i>) donde <i>pid</i> es el ID del proceso en modalidad delimitada.</li> <li>• De lo contrario, el valor es el nombre de la entidad del sistema.</li> </ul>
WORKLOAD_NAME	VARCHAR (128)	workload_name - Elemento de supervisor de nombre de carga de trabajo
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - Elemento de supervisor de identificador de la aparición de carga de trabajo
UOW_ID	INTEGER	uow_id - Elemento de supervisor de ID de unidad de trabajo
ACTIVITY_ID	INTEGER	activity_id - Elemento de supervisor de ID de actividad
PARENT_UOW_ID	INTEGER	parent_uow_id - Elemento de supervisor de ID de unidad de trabajo padre
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - Elemento de supervisor de ID de actividad padre
AGENT_TID	BIGINT	agent_tid - Elemento de supervisor de ID de hebra de agente
AGENT_TYPE	VARCHAR (32)	Tipo de agente. Los tipos de agente son los siguientes: <ul style="list-style-type: none"> <li>• COORDINATOR</li> <li>• OTHER</li> <li>• PDBSUBAGENT</li> <li>• SMPSUBAGENT</li> </ul> Si el valor es COORDINATOR, el ID de agente puede variar en entorno de concentrador.
SMP_COORDINATOR	INTEGER	Indica si el agente es o no de coordinación SMP; 1 indica que lo es, 0 que no lo es.
AGENT_SUBTYPE	VARCHAR (32)	Subtipo de agente. Los subtipos posibles son los siguientes: <ul style="list-style-type: none"> <li>• DSS</li> <li>• OTHER</li> <li>• RPC</li> <li>• SMP</li> </ul>
AGENT_STATE	VARCHAR (32)	Indica si el agente está asociado o activo. Los valores posibles son: <ul style="list-style-type: none"> <li>• ASSOCIATED</li> <li>• ACTIVE</li> </ul>
EVENT_TYPE	VARCHAR (32)	El tipo del último suceso que ha procesado este agente. Los valores posibles son los siguientes: <ul style="list-style-type: none"> <li>• ACQUIRE</li> <li>• PROCESS</li> <li>• WAIT</li> </ul> Consulte la Tabla 72 en la página 419 para obtener más información sobre los valores posibles para esta columna.

Tabla 71. Información devuelta por WLM\_GET\_SERVICE\_CLASS\_AGENTS (continuación)

Nombre de columna	Tipo de datos	Descripción
EVENT_OBJECT	VARCHAR (32)	<p>El objeto del último suceso que ha procesado este agente. Los valores posibles son los siguientes:</p> <ul style="list-style-type: none"> <li>• COMPRESSION_DICTIONARY_BUILD</li> <li>• IMPLICIT_REBIND</li> <li>• INDEX_RECREATE</li> <li>• LOCK</li> <li>• LOCK_ESCALATION</li> <li>• QP_QUEUE</li> <li>• REMOTE_REQUEST</li> <li>• REQUEST</li> <li>• ROUTINE</li> <li>• WLM_QUEUE</li> </ul> <p>Consulte la Tabla 72 en la página 419 para obtener más información sobre los valores posibles para esta columna.</p>
EVENT_STATE	VARCHAR (32)	<p>El estado del último suceso que ha procesado este agente. Los valores posibles son los siguientes:</p> <ul style="list-style-type: none"> <li>• EXECUTING</li> <li>• IDLE</li> </ul> <p>Consulte la Tabla 72 en la página 419 para obtener más información sobre los valores posibles para esta columna.</p>
REQUEST_ID	VARCHAR(64)	<p>ID de petición. Este valor es único solamente si se combina con el valor de <i>descriptor_contexto_aplicación</i>. Puede utilizar esta combinación para distinguir entre una petición que tarda mucho tiempo y varias peticiones; por ejemplo, para distinguir varias captaciones de una captación larga.</p>

Tabla 71. Información devuelta por WLM\_GET\_SERVICE\_CLASS\_AGENTS (continuación)

Nombre de columna	Tipo de datos	Descripción
REQUEST_TYPE	VARCHAR (32)	<p>Tipo de petición. Los valores posibles son los siguientes:</p> <ul style="list-style-type: none"> <li>• Para agentes coordinadores: <ul style="list-style-type: none"> <li>– CLOSE</li> <li>– COMMIT</li> <li>– COMPILE</li> <li>– DESCRIBE</li> <li>– EXCSQLSET</li> <li>– EXECIMMD</li> <li>– EXECUTE</li> <li>– FETCH</li> <li>– INTERNAL <i>número</i>, donde <i>número</i> es el valor de la constante interna</li> <li>– OPEN</li> <li>– PREPARE</li> <li>– REBIND</li> <li>– REDISTRIBUTE</li> <li>– REORG</li> <li>– ROLLBACK</li> <li>– RUNSTATS</li> </ul> </li> <li>• Para subagentes con un AGENT_SUBTYPE de DSS o SMP: <ul style="list-style-type: none"> <li>– Si el número de subsección es distinto de cero, el número de subsección está en el formato SUBSECCIÓN:<i>número subsección</i>; de lo contrario, devuelve NULL.</li> </ul> </li> </ul>

Tabla 71. Información devuelta por WLM\_GET\_SERVICE\_CLASS\_AGENTS (continuación)

Nombre de columna	Tipo de datos	Descripción
REQUEST_TYPE (continuación)	VARCHAR (32)	<ul style="list-style-type: none"> <li>• Para subagentes con un AGENT_SUBTYPE de RPC:                             <ul style="list-style-type: none"> <li>- ABP</li> <li>- CATALOG</li> <li>- INTERNAL</li> <li>- REORG</li> <li>- RUNSTATS</li> <li>- WLM</li> </ul> </li> <li>• Para subagentes con un SUBTYPE de OTHER:                             <ul style="list-style-type: none"> <li>- ABP</li> <li>- APP_RBSVPT</li> <li>- APP_RELSVPT</li> <li>- BACKUP</li> <li>- CLOSE</li> <li>- EXTERNAL_RBSVPT</li> <li>- EVMON</li> <li>- FORCE</li> <li>- FORCE_ALL</li> <li>- INTERNAL <i>número</i>, donde <i>número</i> es el valor de la constante interna</li> <li>- INTERRUPT</li> <li>- NOOP (si no hay petición)</li> <li>- QP</li> <li>- REDISTRIBUTE</li> <li>- STMT_RBSVPT</li> <li>- STOP_USING</li> <li>- UPDATE_DBM_CFG</li> <li>- WLM</li> </ul> </li> </ul>
NESTING_LEVEL	INTEGER	nesting_level - Elemento de supervisor de nivel de anidamiento
INVOCATION_ID	INTEGER	invocation_id - Elemento de supervisor de ID de invocación
ROUTINE_ID	INTEGER	routine_id - Elemento de supervisor de ID de rutina
EVENT_OBJECT_NAME	VARCHAR(1024)	Nombre de objeto de suceso. Si el valor de EVENT_OBJECT es LOCK, el valor de esta columna es el nombre del bloqueo en el que espera el agente. Si el valor de EVENT_OBJECT es WLM_QUEUE, el valor de la columna es el nombre del umbral de WLM que el agente ha puesto en cola. De lo contrario, el valor es NULL.
APPLICATION_NAME	VARCHAR (128)	appl_name - Nombre de aplicación
APPLICATION_ID	VARCHAR (128)	appl_id - ID de aplicación
CLIENT_PID	BIGINT	client_pid - ID de proceso de cliente
SESSION_AUTH_ID	VARCHAR (128)	session_auth_id - ID de autorización de sesión
REQUEST_START_TIME	TIMESTAMP	Hora a la que el agente ha iniciado el proceso de la petición en la que está trabajando actualmente

Tabla 71. Información devuelta por WLM\_GET\_SERVICE\_CLASS\_AGENTS (continuación)

Nombre de columna	Tipo de datos	Descripción
AGENT_STATE_LAST_UPDATE_TIME	TIMESTAMP	Hora a la que se ha cambiado por última vez el suceso que está procesando el agente. El suceso que está procesando actualmente el agente se identifica mediante las columnas EVENT_TYPE, EVENT_OBJECT y EVENT_STATE.
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - Elemento de supervisor de ID ejecutable
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

**Nota:** En la tabla siguiente se indican las combinaciones posibles de los valores de columna EVENT\_STATE, EVENT\_TYPE, EVENT\_OBJECT y EVENT\_OBJECT\_NAME.

Tabla 72. Combinaciones posibles para los valores de columna EVENT\_STATE, EVENT\_TYPE, EVENT\_OBJECT y EVENT\_OBJECT\_NAME

Descripción del suceso	Valor de EVENT_STATE	Valor de EVENT_TYPE	Valor de EVENT_OBJECT	Valor de EVENT_OBJECT_NAME
Obtener bloqueo	IDLE	ACQUIRE	LOCK	Nombre de bloqueo
Escalar bloqueo	EXECUTING	PROCESS	LOCK_ESCALATION	NULL
Procesar petición	EXECUTING	PROCESS	REQUEST	NULL
Esperar una petición nueva	IDLE	WAIT	REQUEST	NULL
Esperar a que una petición se procese en un miembro remoto	IDLE	WAIT	REMOTE_REQUEST	NULL
Esperar en una cola de umbral de WLM	IDLE	WAIT	WLM_QUEUE	Nombre de umbral
Procesar una rutina	EXECUTING	PROCESS	ROUTINE	NULL
Volver a crear un índice	EXECUTING	PROCESS	INDEX_RECREATE	NULL
Crear diccionario de compresión	EXECUTING	PROCESS	COMP_DICT_BLD	NULL
Revinculación implícita	EXECUTING	PROCESS	IMPLICIT_REBIND	NULL

## **WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES - Lista de instancias de carga de trabajo**

La función WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES devuelve la lista de todas las apariciones de carga de trabajo que se ejecutan en una determinada clase de servicio de un miembro determinado. Una aparición de carga de trabajo es una conexión de base de datos específica cuyos atributos coinciden con la definición de una carga de trabajo y por lo tanto está asociada a la carga de trabajo o asignada a la misma.

Consulte la Tabla 73 en la página 421 para ver una lista completa de la información que se puede devolver.

## Sintaxis

►—WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES—(—*nombre\_superclase\_servicio*—,—►  
►—*nombre\_subclase\_servicio*—,—*miembro*—)—►

El esquema es SYSPROC.

## Parámetros de la función de tabla

### *nombre\_superclase\_servicio*

Argumento de entrada de tipo VARCHAR(128) que especifica el nombre de una superclase de servicio de la base de datos actualmente conectada. Si el argumento es nulo o una serie vacía, se recuperan datos correspondientes a todas las superclases de la base de datos que coinciden con los valores de los otros parámetros.

### *nombre\_subclase\_servicio*

Subclase de servicio de destino para la aparición de la carga de trabajo. Cualquier trabajo emitido por esta aparición de la carga de trabajo se ejecutará en esta subclase de servicio bajo la superclase de servicio de destino, excepto las actividades que se correlacionan o se vuelven a correlacionar con una subclase distinta.

### *miembro*

Argumento de entrada de tipo INTEGER que especifica el número de miembro en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros de base de datos. Si se especifica el valor nulo, se establece -1 de forma implícita.

## Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplo

Si un administrador desea ver qué apariciones de carga de trabajo se están ejecutando en el sistema como un conjunto, puede llamar a la función WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES especificando un valor nulo o una serie vacía para *nombre\_superclase\_servicio* y *nombre\_subclase\_servicio* y -2 para *miembro*:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,  
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,  
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,  
       SUBSTR(CHAR(COORD_MEMBER),1,4) AS COORDMEMB,  
       SUBSTR(CHAR(APPLICATION_HANDLE),1,7) AS APPHNDL,  
       SUBSTR(WORKLOAD_NAME,1,22) AS WORKLOAD_NAME,  
       SUBSTR(CHAR(WORKLOAD_OCCURRENCE_ID),1,6) AS WLO_ID
```

```

FROM TABLE(WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES
             (CAST(NULL AS VARCHAR(128)), CAST(NULL AS VARCHAR(128)), -2))
             AS SCINFO
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB, APPHNDL,
         WORKLOAD_NAME, WLO_ID

```

Si el sistema tiene cuatro miembros de base de datos y está ejecutando dos cargas de trabajo en este momento, la consulta anterior genera resultados como los siguientes:

```

SUPERCLASS_NAME      SUBCLASS_NAME      MEMB COORDMEMB ...
-----
SYSDEFAULTMAINTENAN  SYSDEFAULTSUBCLASS 0      0      ...
SYSDEFAULTSYSTEMCLA  SYSDEFAULTSUBCLASS 0      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 0      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 0      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 1      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 1      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 2      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 2      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 3      0      ...
SYSDEFAULTUSERCLASS  SYSDEFAULTSUBCLASS 3      0      ...
... APPHNDL WORKLOAD_NAME      WLO_ID
... -----
... -          -                -
... -          -                -
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2

```

### Nota sobre uso

Los parámetros tienen el efecto de sumarse (operación AND). Es decir, si se especifican parámetros de entrada conflictivos como una superclase de servicio SUP\_A y una subclase de servicio SUB\_B de modo que SUB\_B no es una subclase de SUP\_A, no se devuelve ninguna fila.

**Nota:** Las estadísticas notificadas para la aparición de carga de trabajo (por ejemplo, coord\_act\_completed\_total) se restablecen al principio de cada unidad de trabajo cuando se combinan con las estadísticas correspondientes de carga de trabajo.

### Información devuelta

Tabla 73. Información devuelta para WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES

Nombre de columna	Tipo de datos	Descripción
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Elemento de supervisor de nombre de superclase de servicio
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Elemento de supervisor de nombre de subclase de servicio
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - Elemento de supervisor de número de partición coordinadora

Tabla 73. Información devuelta para WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES (continuación)

Nombre de columna	Tipo de datos	Descripción
APPLICATION_HANDLE	BIGINT	application_handle - Elemento de supervisor de descriptor de contexto de aplicación
WORKLOAD_NAME	VARCHAR(128)	workload_name - Elemento de supervisor de nombre de carga de trabajo
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - Elemento de supervisor de identificador de la aparición de carga de trabajo
UOW_ID	INTEGER	uow_id - Elemento de supervisor de ID de unidad de trabajo
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - Elemento de supervisor de estado de la aparición de carga de trabajo
SYSTEM_AUTH_ID	VARCHAR(128)	system_auth_id - Elemento de supervisor de identificador de autorización del sistema
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - Elemento de supervisor de ID de autorización de sesión
APPLICATION_NAME	VARCHAR(128)	appl_name - Elemento de supervisor de nombre de aplicación
CLIENT_WRKSTNNAME	VARCHAR(255)	client_wrkstnname - Elemento de supervisor de nombre de estación de trabajo cliente
CLIENT_ACCTNG	VARCHAR(255)	client_acctng - Elemento de supervisor de serie de contabilidad cliente
CLIENT_USER	VARCHAR(255)	Valor actual del registro especial CLIENT_USERID correspondiente a esta instancia de carga de trabajo.
CLIENT_APPLNAME	VARCHAR(255)	client_applname - Elemento de supervisor de nombre de aplicación cliente
COORD_ACT_COMPLETED_TOTAL	INTEGER	coord_act_completed_total - Elemento de supervisor de total de actividades de coordinador completadas
COORD_ACT_ABORTED_TOTAL	INTEGER	coord_act_aborted_total - Elemento de supervisor de total de actividades de coordinador finalizadas anormalmente
COORD_ACT_REJECTED_TOTAL	INTEGER	coord_act_rejected_total - Elemento de supervisor de total de actividades de coordinador rechazadas
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - Elemento de supervisor de máximo de actividades simultáneas
ADDRESS	VARCHAR(255)	dirección - dirección IP desde la que se ha iniciado la conexión
APPL_ID	VARCHAR(128)	appl_id - ID de aplicación
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos
COORD_MEMBER	SMALLINT	coord_member - Elemento de supervisor de miembro coordinador

## Función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS - Devolver estadísticas de subclases de servicio

La función WLM\_GET\_SERVICE\_SUBCLASS\_STATS devuelve estadísticas básicas para una o varias subclases de servicio.

Consulte la Tabla 74 en la página 425 para ver una lista completa de la información que se puede devolver.

### Sintaxis

```
►►—WLM_GET_SERVICE_SUBCLASS_STATS—(—nombre_superclase_servicio—, —————→  
►—nombre_subclase_servicio—, —miembro—)—————→◄◄
```

El esquema es SYSPROC.

### Parámetros de la función de tabla

#### *nombre\_superclase\_servicio*

Argumento de entrada de tipo VARCHAR(128) que especifica el nombre de una superclase de servicio de la base de datos actualmente conectada. Si el argumento es nulo o una serie vacía, se recuperan datos correspondientes a todas las superclases de la base de datos.

#### *nombre\_subclase\_servicio*

Argumento de entrada de tipo VARCHAR(128) que especifica el nombre de una subclase de servicio de la base de datos actualmente conectada. Si el argumento es nulo o una serie vacía, se recuperan datos correspondientes a todas las subclases de la base de datos.

#### *miembro*

Argumento de entrada de tipo INTEGER que especifica un número de miembro válido en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro actual o -2 para todos los miembros de base de datos. Si se especifica el valor nulo, se establece -1 de forma implícita.

### Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

### Privilegio PUBLIC por omisión

Ninguna

### Ejemplos

*Ejemplo 1:* Puesto que cada actividad se tiene que correlacionar con una clase de servicio de DB2 antes de que se ejecute, se puede supervisar el estado global del

sistema utilizando las funciones de tabla de estadísticas de clase de servicio y consultando todas las clases de servicio de todos los miembros. En el ejemplo siguiente, se pasa un valor nulo para *service\_superclass\_name* y *service\_subclass\_name* para devolver estadísticas para todas las clases de servicio, y se especifica el valor -2 para *member* para devolver estadísticas para todos los miembros:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       CAST(COORD_ACT_LIFETIME_AVG / 1000 AS DECIMAL(9,3))
       AS AVGLIFETIME,
       CAST(COORD_ACT_LIFETIME_STDDEV / 1000 AS DECIMAL(9,3))
       AS STDDEVLIFETIME,
       SUBSTR(CAST(LAST_RESET AS VARCHAR(30)),1,16) AS LAST_RESET
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB
```

La sentencia devuelve estadísticas de clase de servicio como duración media de la actividad y desviación estándar en segundos, tal como se muestra en el siguiente ejemplo de salida:

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	...
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	...
...	AVGLIFETIME	STDDEVLIFETIME	LAST_RESET
...	691.242	34.322	2006-07-24-11.44
...	644.740	22.124	2006-07-24-11.44
...	612.431	43.347	2006-07-24-11.44
...	593.451	28.329	2006-07-24-11.44

*Ejemplo 2:* La misma función de tabla también puede dar el valor más alto correspondiente a simultaneidad promedia de actividades de coordinación que se ejecutan en la clase de servicio de cada miembro:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       CONCURRENT_ACT_TOP AS ACTTOP,
       CONCURRENT_WLO_TOP AS CONNTOP
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB
```

Un ejemplo de la salida es el siguiente:

SUPERCLASS_NAME	SUBCLASS_NAME	MEMB	ACTTOP	CONNTOP
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	10	7
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	0	0

Mediante la comprobación de los tiempos promedios de ejecución y los números de actividades en la salida de esta función de tabla, obtendrá una buena vista a nivel alto de la carga de cada miembro para una determinada base de datos. Cualquier variación significativa de los indicadores de nivel alto que devuelve esta función de tabla puede indicar un cambio en la carga del sistema.

*Ejemplo 3:* Si una actividad utiliza umbrales con acciones REMAP ACTIVITY TO, la actividad puede dedicar tiempo a más de una clase de servicio durante su vida útil. Se puede determinar cuántas actividades han pasado a través de un conjunto de clases de servicio consultando las columnas ACTIVITIES\_MAPPED\_IN y ACTIVITIES\_MAPPED\_OUT, como se muestra en el ejemplo siguiente:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       ACTIVITIES_MAPPED_IN AS MAPPED_IN,
       ACTIVITIES_MAPPED_OUT AS MAPPED_OUT
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS(CAST(NULL AS VARCHAR(128)),
      CAST(NULL AS VARCHAR(128)), -2)) AS SCSTATS
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME
```

Un ejemplo de la salida es el siguiente:

SUPERCLASS_NAME	SUBCLASS_NAME	MAPPED_IN	MAPPED_OUT
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SUBCLASS1	0	7
SUPERCLASS1	SUBCLASS2	7	0

## Notas de uso

Algunas estadísticas sólo se devolverán si se establecen los parámetros COLLECT AGGREGATE ACTIVITY DATA y COLLECT AGGREGATE REQUEST DATA para la subclase de servicio correspondiente en un valor que no sea "NONE".

La función de tabla WLM\_GET\_SERVICE\_SUBCLASS\_STATS devuelve una fila de datos por subclase de servicio y por miembro. La función no realiza ninguna agregación de datos entre clases de servicio (de una partición) ni entre particiones (para una o más clases de servicio). Sin embargo, puede utilizar consultas de SQL para agregar datos.

Los parámetros tienen el efecto de sumarse (operación AND). Es decir, si se especifican parámetros de entrada conflictivos, como un nombre de superclase llamado SUPA y una subclase llamada SUBB de modo que SUBB no es una subclase de SUPA, no se devolverá ninguna fila.

## Información devuelta

Tabla 74. Información devuelta para WLM\_GET\_SERVICE\_SUBCLASS\_STATS

Nombre de columna	Tipo de datos	Descripción
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Elemento de supervisor de nombre de superclase de servicio
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Elemento de supervisor de nombre de subclase de servicio
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
LAST_RESET	TIMESTAMP	last_reset - elemento de supervisor de indicación de fecha y hora de último restablecimiento
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - Elemento de supervisor de total de actividades de coordinador completadas

Tabla 74. Información devuelta para WLM\_GET\_SERVICE\_SUBCLASS\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - Elemento de supervisor de total de actividades de coordinador finalizadas anormalmente
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - Elemento de supervisor de total de actividades de coordinador rechazadas
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - Elemento de supervisor de máximo de actividades simultáneas
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - Elemento de supervisor de vida útil de actividades de coordinador más elevada
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - Elemento de supervisor de promedio de vida útil de actividades de coordinador
COORD_ACT_LIFETIME_STDDEV	DOUBLE	<p>Desviación estándar de duración para actividades de coordinación a nivel de anidamiento 0 asociadas a esta subclase de servicio desde el último restablecimiento. Si el parámetro COLLECT AGGREGATE ACTIVITY DATA de la clase de servicio está establecido en NONE, el valor de la columna es nulo. Las unidades son milisegundos.</p> <p>Esta desviación estándar se calcula a partir del histograma de duración de las actividades de coordinación y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Se devuelve el valor -1 si alguno de los valores está dentro del último intervalo de histograma.</p> <p>El valor de COORD_ACT_LIFETIME_STDDEV de una subclase de servicio no se ve afectado por las actividades que pasan a través de la subclase de servicio pero que están correlacionadas con una subclase distinta antes de completarse.</p>
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - Elemento de supervisor de promedio de tiempo de ejecución de actividades de coordinador

Tabla 74. Información devuelta para WLM\_GET\_SERVICE\_SUBCLASS\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	<p>Desviación estándar de tiempos de ejecución para actividades de coordinación a nivel de anidamiento 0 asociadas a esta subclase de servicio desde el último restablecimiento. Las unidades son milisegundos.</p> <p>Esta desviación estándar se calcula a partir del histograma de tiempo de ejecución de las actividades de coordinación y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Se devuelve el valor -1 si alguno de los valores está dentro del último intervalo de histograma.</p> <p>La desviación estándar del tiempo de ejecución de una subclase de servicio no se ve afectada por las actividades que pasan a través de la subclase pero que están correlacionadas con una subclase distinta antes de completarse.</p>
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	<p>coord_act_queue_time_avg - Elemento de supervisor de promedio de tiempo en cola de actividades de coordinador</p>
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	<p>Desviación estándar del tiempo en cola para actividades de coordinación a nivel de anidamiento 0 asociadas a esta subclase de servicio desde el último restablecimiento. Si el parámetro COLLECT AGGREGATE ACTIVITY DATA de la clase de servicio está establecido en NONE, el valor de la columna es nulo. Las unidades son milisegundos.</p> <p>Esta desviación estándar se calcula a partir del histograma de tiempo en cola de las actividades de coordinación y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Se devuelve el valor -1 si alguno de los valores está dentro del último intervalo de histograma.</p> <p>La desviación estándar del tiempo en cola sólo se cuenta para establecer el total de la subclase de servicio en la que se ha puesto en cola la actividad.</p>
NUM_REQUESTS_ACTIVE	BIGINT	<p>Número de peticiones que se están ejecutando en la subclase de servicio en el momento de ejecutar esta función de tabla.</p>

Tabla 74. Información devuelta para WLM\_GET\_SERVICE\_SUBCLASS\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
NUM_REQUESTS_TOTAL	BIGINT	<p>Número de peticiones que se han acabado de ejecutar en esta subclase de servicio desde el último restablecimiento. Este estado acabado se aplica a cualquier petición, independientemente de su pertenencia a una actividad. Si el parámetro COLLECT AGGREGATE ACTIVITY DATA de la clase de servicio está establecido en NONE, el valor de la columna es nulo.</p> <p>El valor NUM_REQUESTS_TOTAL de una subclase de servicio no se ve afectado por peticiones que pasan a través de la subclase de servicio pero que no se completan en la misma.</p>
REQUEST_EXEC_TIME_AVG	DOUBLE	request_exec_time_avg - Elemento de supervisor de promedio de tiempo de ejecución de peticiones
REQUEST_EXEC_TIME_STDDEV	DOUBLE	<p>Desviación estándar de tiempos de ejecución para peticiones asociadas a esta subclase de servicio desde el último restablecimiento. Las unidades son milisegundos. Si el parámetro COLLECT AGGREGATE REQUEST DATA en la clase de servicio está establecido en NONE, el valor de esta columna es NULL.</p> <p>Esta desviación estándar se calcula a partir del histograma de tiempo de ejecución de peticiones y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Se devuelve el valor -1 si alguno de los valores está dentro del último intervalo de histograma.</p> <p>La desviación estándar del tiempo de ejecución de una subclase de servicio no se ve afectada por peticiones que pasan a través de la subclase pero que no se completan en la misma.</p>

Tabla 74. Información devuelta para WLM\_GET\_SERVICE\_SUBCLASS\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
REQUEST_EXEC_TIME_TOTAL	BIGINT	<p>Suma de los tiempos de ejecución para peticiones asociadas a esta subclase de servicio desde el último restablecimiento. Las unidades son milisegundos. Si el parámetro COLLECT AGGREGATE REQUEST DATA en la clase de servicio está establecido en NONE, el valor de esta columna es NULL.</p> <p>Este total se calcula a partir del histograma de tiempos de ejecución de peticiones y puede resultar impreciso si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Se devuelve el valor -1 si alguno de los valores está dentro del último intervalo de histograma.</p> <p>El tiempo total de ejecución de una subclase de servicio no se ve afectada por peticiones que pasan a través de la subclase pero que no se completan en la misma.</p>
ACT_REMAPPED_IN	BIGINT	Número de actividades que se han correlacionado con esta subclase de servicio por una acción REMAP ACTIVITY de umbral desde el último restablecimiento.
ACT_REMAPPED_OUT	BIGINT	Número de actividades que se han correlacionado fuera de esta subclase de servicio por una acción REMAP ACTIVITY de umbral desde el último restablecimiento.
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - Elemento de supervisor de máximo de apariciones de carga de trabajo simultáneas
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - Elemento de supervisor de tiempo total de UOW superior
UOW_THROUGHPUT	DOUBLE	<p>uow_throughput - Elemento de supervisor de rendimiento de unidad de trabajo</p> <p>Rendimiento de la unidad de trabajo desde el último restablecimiento de las estadísticas.</p>
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - Elemento de supervisor de promedio de vida útil unidad de trabajo
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - Elemento de supervisor de número total de unidades de trabajo completadas
TOTAL_CPU_TIME	BIGINT	total_cpu_time - Elemento de supervisor de tiempo total de CPU
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - Elemento de supervisor de total de tiempo en cola de ejecución del asignador

Tabla 74. Información devuelta para WLM\_GET\_SERVICE\_SUBCLASS\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
ACT_THROUGHPUT	DOUBLE	act_throughput - Elemento de supervisor de rendimiento de actividad
CPU_UTILIZATION	DOUBLE	cpu_utilization - Elemento de supervisor de utilización de CPU
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - Elemento de supervisor de total de actividades de coordinador externo satisfactorias
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - Elemento de supervisor de total de actividades de coordinador externo anómalas
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - Elemento de supervisor de total de actividades de coordinador externo rechazadas
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

## WLM\_GET\_SERVICE\_SUPERCLASS\_STATS - Devolver estadísticas de superclases de servicio

La función WLM\_GET\_SERVICE\_SUPERCLASS\_STATS devuelve estadísticas básicas para una o varias superclases de servicio.

### Sintaxis

```
►► WLM_GET_SERVICE_SUPERCLASS_STATS (—nombre_superclase_servicio—, —miembro—)
```

El esquema es SYSPROC.

### Parámetros de la función de tabla

#### *nombre\_superclase\_servicio*

Argumento de entrada de tipo VARCHAR(128) que especifica el nombre de una superclase de servicio de la base de datos actualmente conectada. Si el argumento es nulo o una serie vacía, se recuperan datos correspondientes a todas las superclases de la base de datos.

#### *miembro*

Argumento de entrada de tipo INTEGER que especifica un número de miembro válido en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros de base de datos. Si se especifica el valor nulo, se establece -1 de forma implícita.

### Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM

- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplo

La siguiente consulta muestra las estadísticas básicas correspondientes a todas las superclases de servicio del sistema de todos los miembros de la base de datos:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME, 1, 26) SERVICE_SUPERCLASS_NAME,
       MEMBER,
       LAST_RESET,
       CONCURRENT_CONNECTION_TOP CONCURRENT_CONN_TOP
FROM TABLE(WLM_GET_SERVICE_SUPERCLASS_STATS(' ', -2)) as SCSTATS
```

Un ejemplo de la salida es el siguiente:

```
SERVICE_SUPERCLASS_NAME  MEMBER ...
-----
SYSDEFAULTSYSTEMCLASS    0 ...
SYSDEFAULTMAINTENANCECLASS 0 ...
SYSDEFAULTUSERCLASS      0 ...
... LAST_RESET            CONCURRENT_CONN_TOP
-----
... 2006-09-05-09.38.44.396788 0
... 2006-09-05-09.38.44.396795 0
... 2006-09-05-09.38.44.396796 1
```

## Nota sobre uso

La función de tabla WLM\_GET\_SERVICE\_SUPERCLASS\_STATS devuelve una fila de datos por superclase de servicio y por miembro. La función no realiza ninguna agregación de datos entre superclases de servicio (de un miembro) ni entre miembros (para una o más superclases de servicio). Sin embargo, puede utilizar consultas de SQL para agregar datos, como se muestra en el ejemplo anterior.

## Información devuelta

Tabla 75. Información devuelta para WLM\_GET\_SERVICE\_SUPERCLASS\_STATS

Nombre de columna	Tipo de datos	Descripción
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Elemento de supervisor de nombre de superclase de servicio
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
LAST_RESET	TIMESTAMP	last_reset - elemento de supervisor de indicación de fecha y hora de último restablecimiento
CONCURRENT_CONNECTION_TOP	INTEGER	concurrent_connection_top - Elemento de supervisor de máximo de conexiones simultáneas
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

## WLM\_GET\_WORK\_ACTION\_SET\_STATS - Devolver estadísticas de conjunto de acciones de trabajo

La función WLM\_GET\_WORK\_ACTION\_SET\_STATS devuelve las estadísticas correspondientes a un conjunto de acciones de trabajo.

### Sintaxis

```
►►—WLM_GET_WORK_ACTION_SET_STATS—(—nombre_conjunto_acciones_trabajo—, —————►  
►—miembro—)—————►◀
```

El esquema es SYSPROC.

### Parámetros de la función de tabla

*nombre\_conjunto\_acciones\_trabajo*

Argumento de entrada de tipo VARCHAR (128) que especifica el conjunto de acciones de trabajo para el que hay que devolver estadísticas. Si el argumento es nulo o una serie vacía, se devuelven estadísticas correspondientes a todos los conjuntos de acción de trabajo.

*miembro*

Argumento de entrada de tipo INTEGER que especifica un número de miembro válido en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros de base de datos. Si se especifica el valor nulo, se establece -1 de forma implícita.

### Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

### Privilegio PUBLIC por omisión

Ninguna

### Ejemplo

Supongamos que hay tres clases de trabajo: ReadClass, WriteClass y LoadClass. Hay una acción de trabajo asociada a ReadClass y una asociada a LoadClass, pero no hay ninguna acción de trabajo asociada a WriteClass. En el miembro 0, el número de actividades que está actualmente en ejecución o en cola es el siguiente:

- Clase ReadClass: ocho
- Clase WriteClass: cuatro
- Clase LoadClass: dos
- Sin asignar: tres

```

SELECT SUBSTR(WORK_ACTION_SET_NAME,1,18) AS WORK_ACTION_SET_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(WORK_CLASS_NAME,1,15) AS WORK_CLASS_NAME,
       LAST_RESET,
       SUBSTR(CHAR(ACT_TOTAL),1,14) AS ACT_TOTAL
FROM TABLE(WLM_GET_WORK_ACTION_SET_STATS
            (CAST(NULL AS VARCHAR(128)), -2)) AS WASSTATS
ORDER BY WORK_ACTION_SET_NAME, WORK_CLASS_NAME, MEMB

```

Un ejemplo de la salida es el siguiente. Puesto que no hay ninguna acción de trabajo asociada a la clase de trabajo WriteClass, las cuatro actividades a las que se aplican se cuentan en la clase artificial indicada con un asterisco (\*) en la salida. Las tres actividades que no estaban asignadas a ninguna clase de trabajo también se incluyen en la clase artificial.

WORK_ACTION_SET_NAME	MEMB	WORK_CLASS_NAME	LAST_RESET	ACT_TOTAL
AdminActionSet	0	ReadClass	2005-11-25-18.52.49.343000	8
AdminActionSet	1	ReadClass	2005-11-25-18.52.50.478000	0
AdminActionSet	0	LoadClass	2005-11-25-18.52.49.343000	2
AdminActionSet	1	LoadClass	2005-11-25-18.52.50.478000	0
AdminActionSet	0	*	2005-11-25-18.52.49.343000	7
AdminActionSet	1	*	2005-11-25-18.52.50.478000	0

## Información devuelta

Tabla 76. Información devuelta para WLM\_GET\_WORK\_ACTION\_SET\_STATS

Nombre de columna	Tipo de datos	Descripción
WORK_ACTION_SET_NAME	VARCHAR(128)	work_action_set_name - Elemento de supervisor de nombre de conjunto de acciones de trabajo
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
LAST_RESET	TIMESTAMP	last_reset - elemento de supervisor de indicación de fecha y hora de último restablecimiento
WORK_CLASS_NAME	VARCHAR(128)	work_class_name - Elemento de supervisor de clase de trabajo
ACT_TOTAL	BIGINT	act_total - Elemento de supervisor de total de actividades
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

## WLM\_GET\_WORKLOAD\_OCCURRENCE - Devolución de una lista de actividades

La función WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES devuelve la lista de todas las actividades enviadas por una aplicación especificada en el miembro especificado y que todavía no se han completado.

Consulte la Tabla 77 en la página 435 para ver una lista completa de la información que se puede devolver.

### Sintaxis

```

►—WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES—(—descriptor_contexto_aplicación—, —————►
►—miembro—)———►

```

El esquema es SYSPROC.

## Parámetros de la función de tabla

### *descriptor\_contexto\_aplicación*

Argumento de entrada de tipo BIGINT que especifica un descriptor de contexto de aplicación para el que se debe devolver una lista de actividades. Si el argumento es nulo, se recuperan datos correspondientes a todas las aplicaciones de la base de datos.

### *miembro*

Argumento de entrada de tipo INTEGER que especifica un número de miembro válido en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros. Si se especifica el valor nulo, se establece -1 de forma implícita.

## Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## Privilegio PUBLIC por omisión

Ninguna

## Ejemplos

### Actividades actualmente en ejecución con un descriptor de contexto de aplicación conocido

Después de identificar el descriptor de contexto de aplicación, se pueden buscar todas las actividades que se ejecutan actualmente en dicha aplicación. Por ejemplo, supongamos que un administrador desea tener una lista de las actividades de una aplicación cuyo descriptor de contexto de aplicación, determinado mediante el mandato **LIST APPLICATIONS**, es 1. El administrador ejecuta la consulta siguiente:

```
SELECT SUBSTR(CHAR(COORD_MEMBER),1,5) AS COORD,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       SUBSTR(CHAR(UOW_ID),1,5) AS UOWID,
       SUBSTR(CHAR(ACTIVITY_ID),1,5) AS ACTID,
       SUBSTR(CHAR(PARENT_UOW_ID),1,8) AS PARUOWID,
       SUBSTR(CHAR(PARENT_ACTIVITY_ID),1,8) AS PARACTID,
       ACTIVITY TYPE AS ACTTYPE,
       SUBSTR(CHAR(NESTING_LEVEL),1,7) AS NESTING
FROM TABLE(WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES(1, -2)) AS WLOACTS
ORDER BY MEMB, UOWID, ACTID
```

Un ejemplo de resultado de la consulta es el siguiente:

COORD	MEMB	UOWID	ACTID	PARUOWID	PARACTID	ACTTYPE	NESTING
0	0	2	3	-	-	CALL	0
0	0	2	5	2	3	READ_DML	1

### Actividades actualmente en ejecución en el sistema

La consulta siguiente une la salida de WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES con la salida de

MON\_GET\_PKG\_CACHE\_STMT en EXECUTABLE\_ID para proporcionar el texto de la sentencia para todas las actividades SQL que se están ejecutando actualmente en el sistema:

```
SELECT t.application_handle,
       t.uow_id,
       t.activity_id,
       varchar(p.stmt_text, 256) as stmt_text
FROM table(wlm_get_workload_occurrence_ACTIVITIES(NULL, -1)) as t,
     table(mon_get_pkg_cache_stmt(NULL, NULL, NULL, -1)) as p
WHERE t.executable_id = p.executable_id
```

Un ejemplo de la salida es el siguiente:

```
APPLICATION_HANDLE  UOW_ID    ACTIVITY_ID  STMT_TEXT
-----
1          1          1    SELECT * FROM SYSCAT.TABLES
47         1          36    INSERT INTO T1 VALUES(123)
```

## Información devuelta

Tabla 77. Información devuelta por WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES

Nombre de columna	Tipo de datos	Descripción
APPLICATION_HANDLE	BIGINT	application_handle - Elemento de supervisor de descriptor de contexto de aplicación
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - Elemento de supervisor de número de partición coordinadora
LOCAL_START_TIME	TIMESTAMP	local_start_time - Elemento de supervisor de hora de inicio local
UOW_ID	INTEGER	uow_id - Elemento de supervisor de ID de unidad de trabajo
ACTIVITY_ID	INTEGER	activity_id - Elemento de supervisor de ID de actividad
PARENT_UOW_ID	INTEGER	parent_uow_id - Elemento de supervisor de ID de unidad de trabajo padre
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - Elemento de supervisor de ID de actividad padre
ACTIVITY_STATE	VARCHAR(32)	activity_state - Elemento de supervisor de estado de actividad

Tabla 77. Información devuelta por *WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES* (continuación)

Nombre de columna	Tipo de datos	Descripción
ACTIVITY_STATE (continuación)	VARCHAR(32)	<p>Estado de la actividad. Los valores posibles son los siguientes:</p> <p><i>QUEUED</i>                      Un umbral de colas de gestión de carga de trabajo pone en cola la actividad. En un entorno de base de datos particionada, este estado podría significar que el agente coordinador ha realizado un RPC en el miembro de catálogo para obtener certificados de umbral y aún no ha recibido una respuesta. Este estado puede indicar que un umbral de colas de gestión de carga de trabajo ha colocado en cola la actividad o, si no ha transcurrido mucho tiempo, puede indicar que la actividad está en proceso de obtener sus tiquets. Para obtener una imagen más precisa de si la actividad se está colocando realmente en cola, determine qué agente está trabajando en la actividad y busque si el valor <i>EVENT_OBJECT</i> del objeto del miembro del catálogo tiene el valor <i>WLM_QUEUE</i>.</p> <p><i>TERMINATING</i>                      Ha finalizado la ejecución de la actividad y se está eliminando del sistema.</p>
ACTIVITY_TYPE	VARCHAR(32)	<p>Tipo de actividad. Los valores posibles son los siguientes:</p> <ul style="list-style-type: none"> <li>• CALL</li> <li>• DDL</li> <li>• LOAD</li> <li>• OTHER</li> <li>• READ_DML</li> <li>• WRITE_DML</li> </ul>
NESTING_LEVEL	INTEGER	nesting_level - Elemento de supervisor de nivel de anidamiento
INVOCATION_ID	INTEGER	invocation_id - Elemento de supervisor de ID de invocación
ROUTINE_ID	INTEGER	routine_id - Elemento de supervisor de ID de rutina
UTILITY_ID	INTEGER	utility_id - Elemento de supervisor de ID de programa de utilidad
SERVICE_CLASS_ID	INTEGER	service_class_id - Elemento de supervisor de ID de clase de servicio

Tabla 77. Información devuelta por WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES (continuación)

Nombre de columna	Tipo de datos	Descripción
DATABASE_WORK_ACTION_SET_ID	INTEGER	Uno de los valores siguientes: <ul style="list-style-type: none"> <li>• Si esta actividad se ha clasificado en una clase de trabajo de ámbito de base de datos, el valor es el ID del conjunto de la clase de trabajo del que esta clase de trabajo es miembro.</li> <li>• Si esta actividad no se ha clasificado en una clase de trabajo de ámbito de base de datos, el valor es nulo.</li> </ul>
DATABASE_WORK_CLASS_ID	INTEGER	Uno de los valores siguientes: <ul style="list-style-type: none"> <li>• Si esta actividad se ha clasificado en una clase de trabajo de ámbito de base de datos, el valor es el ID del conjunto de la clase de trabajo.</li> <li>• Si esta actividad no se ha clasificado en una clase de trabajo de ámbito de base de datos, el valor es nulo.</li> </ul>
SERVICE_CLASS_WORK_ACTION_SET_ID	INTEGER	Uno de los valores siguientes: <ul style="list-style-type: none"> <li>• Si esta actividad se ha clasificado en una clase de trabajo de ámbito de clase de servicio, el valor es el ID del conjunto de la acción de trabajo asociado al conjunto de la clase de trabajo al que pertenece esta clase de trabajo.</li> <li>• Si esta actividad no se ha clasificado en una clase de trabajo de ámbito de clase de servicio, el valor es nulo.</li> </ul>
SERVICE_CLASS_WORK_CLASS_ID	INTEGER	Uno de los valores siguientes: <ul style="list-style-type: none"> <li>• Si esta actividad se ha clasificado en una clase de trabajo de ámbito de clase de servicio, el valor es el ID de la clase de trabajo asociado a esta actividad.</li> <li>• Si esta actividad no se ha clasificado en una clase de trabajo de ámbito de clase de servicio, el valor es nulo.</li> </ul>
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - Elemento de supervisor de ID ejecutable
TOTAL_CPU_TIME	BIGINT	total_cpu_time - Tiempo total de CPU
ROWS_READ	BIGINT	rows_read - Filas leídas
ROWS_RETURNED	BIGINT	rows_returned - Filas devueltas
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - Estimación del coste de la consulta
DIRECT_READS	BIGINT	direct_reads - Lecturas directas de la base de datos
DIRECT_WRITES	BIGINT	direct_writes - Grabaciones directas en la base de datos
ENTRY_TIME	TIMESTAMP	entry_time - Hora de entrada
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

Tabla 77. Información devuelta por WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES (continuación)

Nombre de columna	Tipo de datos	Descripción
COORD_MEMBER	SMALLINT	coord_member - Elemento de supervisor de miembro coordinador

## Función de tabla WLM\_GET\_WORKLOAD\_STATS - Devolver estadísticas de carga de trabajo

La función WLM\_GET\_WORKLOAD\_STATS devuelve una fila de estadísticas de carga de trabajo correspondientes a cada combinación de nombre de carga de trabajo y número de miembro de base de datos.

Consulte la Tabla 78 en la página 439 para ver una lista completa de la información que se puede devolver.

### Sintaxis

```
►► WLM_GET_WORKLOAD_STATS (—nombre_carga_trabajo—, —miembro—) ◀◀
```

El esquema es SYSPROC.

### Parámetros de la función de tabla

#### *nombre\_cargatrabajo*

Argumento de entrada de tipo VARCHAR(128) que especifica una carga de trabajo para la que se tienen que devolver estadísticas. Si el argumento es NULL o una serie vacía, se devuelven estadísticas correspondientes a todas las cargas de trabajo.

#### *miembro*

Argumento de entrada de tipo INTEGER que especifica el número de miembro en la misma instancia que la base de datos conectada actualmente. Especifique -1 para el miembro de base de datos actual o -2 para todos los miembros. Si se especifica un valor nulo, se establece -1 de forma implícita.

### Autorización

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

### Privilegio PUBLIC por omisión

Ninguna

### Ejemplo

La consulta siguiente muestra estadísticas para las cargas de trabajo:

```

SELECT SUBSTR(WORKLOAD_NAME,1,18) AS WL_DEF_NAME,
       SUBSTR(CHAR(MEMBER),1,4) AS MEMB,
       COORD_ACT_LIFETIME_TOP,
       COORD_ACT_LIFETIME_AVG,
       COORD_ACT_LIFETIME_STDDEV
FROM TABLE(WLM_GET_WORKLOAD_STATS(CAST(NULL AS VARCHAR(128)), -2)) AS WLSTATS
ORDER BY WL_DEF_NAME, MEMB

```

Un ejemplo de resultado de la consulta es el siguiente:

```

WL_DEF_NAME      MEMB COORD_ACT_LIFETIME_TOP ...
-----
SYSDEFAULTADMWOKL 0          -1 ...
SYSDEFAULTUSERWOKL 0          -1 ...
WL1                0           2 ...
... COORD_ACT_LIFETIME_AVG  COORD_ACT_LIFETIME_STDDEV
... -----
... -1.000000000000000E+000  -1.000000000000000E+000
... -1.000000000000000E+000  -1.000000000000000E+000
... +2.560000000000000E+000   +6.00000000000001E-002

```

### Nota sobre uso

La función no realiza ninguna agregación de datos entre cargas de trabajo, miembros o clases de servicio. Sin embargo, puede utilizar consultas de SQL para agregar datos.

### Información devuelta

Tabla 78. Información devuelta por WLM\_GET\_WORKLOAD\_STATS

Nombre de columna	Tipo de datos	Descripción
WORKLOAD_NAME	VARCHAR(128)	workload_name - Elemento de supervisor de nombre de carga de trabajo
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Elemento de supervisor de número de partición de base de datos
LAST_RESET	TIMESTAMP	last_reset - elemento de supervisor de indicación de fecha y hora de último restablecimiento
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - Elemento de supervisor de máximo de apariciones de carga de trabajo simultáneas
CONCURRENT_WLO_ACT_TOP	INTEGER	concurrent_act_top - Elemento de supervisor de máximo de actividades de carga de trabajo simultáneas
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - Elemento de supervisor de total de actividades de coordinador completadas
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - Elemento de supervisor de total de actividades de coordinador finalizadas anormalmente
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - Elemento de supervisor de total de actividades de coordinador rechazadas
WLO_COMPLETED_TOTAL	BIGINT	wlo_completed_total - Elemento de supervisor de total de apariciones completadas de la carga de trabajo
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - Elemento de supervisor de vida útil de actividades de coordinador más elevada

Tabla 78. Información devuelta por WLM\_GET\_WORKLOAD\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - Elemento de supervisor de promedio de vida útil de actividades de coordinador
COORD_ACT_LIFETIME_STDDEV	DOUBLE	Desviación estándar de duración para actividades de coordinación completadas o abortadas a nivel de anidamiento 0 asociadas a esta carga de trabajo. Las unidades son milisegundos. Si el parámetro COLLECT AGGREGATE ACTIVITY DATA de la carga de trabajo está establecido en NONE, el valor de la columna es nulo. Esta desviación estándar se calcula a partir del histograma de duración de las actividades de coordinación y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Si alguno de los valores está dentro del último intervalo de histograma, se devuelve el valor -1.
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - Elemento de supervisor de promedio de tiempo de ejecución de actividades de coordinador
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	Desviación estándar de los tiempos de ejecución para actividades de coordinación completadas o abortadas a nivel de anidamiento 0 asociadas a esta carga de trabajo. Las unidades son milisegundos. Esta desviación estándar se calcula a partir del histograma de tiempo de ejecución de las actividades de coordinación y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Si alguno de los valores está dentro del último intervalo de histograma, se devuelve el valor -1. Si el parámetro COLLECT AGGREGATE ACTIVITY DATA de la carga de trabajo está establecido en NONE, el valor de la columna es nulo.
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	coord_act_queue_time_avg - Elemento de supervisor de promedio de tiempo en cola de actividades de coordinador
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	Desviación estándar del tiempo en cola para actividades de coordinación completadas o abortadas a nivel de anidamiento 0 asociadas a esta carga de trabajo. Las unidades son milisegundos. Si el parámetro COLLECT AGGREGATE ACTIVITY DATA de la carga de trabajo está establecido en NONE, el valor de la columna es nulo. Esta desviación estándar se calcula a partir del histograma de tiempo en cola de las actividades de coordinación y puede resultar imprecisa si el histograma no se ha dimensionado correctamente para dar cabida a los datos. Si alguno de los valores está dentro del último intervalo de histograma, se devuelve el valor -1.
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - Elemento de supervisor de tiempo total de UOW superior
UOW_THROUGHPUT	DOUBLE	uow_throughput - Elemento de supervisor de rendimiento de unidad de trabajo

Tabla 78. Información devuelta por WLM\_GET\_WORKLOAD\_STATS (continuación)

Nombre de columna	Tipo de datos	Descripción
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - Elemento de supervisor de promedio de vida útil unidad de trabajo
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - Elemento de supervisor de número total de unidades de trabajo completadas
TOTAL_CPU_TIME	BIGINT	total_cpu_time - Elemento de supervisor de tiempo total de CPU
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - Elemento de supervisor de total de tiempo en cola de ejecución del asignador
ACT_THROUGHPUT	DOUBLE	act_throughput - Elemento de supervisor de rendimiento de actividad
CPU_UTILIZATION	DOUBLE	cpu_utilization - Elemento de supervisor de utilización de CPU
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - Elemento de supervisor de total de actividades de coordinador externo satisfactorias
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - Elemento de supervisor de total de actividades de coordinador externo anómalas
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - Elemento de supervisor de total de actividades de coordinador externo rechazadas
MEMBER	SMALLINT	member - Elemento de supervisor de miembro de base de datos

## Procedimiento WLM\_SET\_CLIENT\_INFO - Establecer información de cliente

El procedimiento WLM\_SET\_CLIENT\_INFO establece información de cliente asociada con la conexión actual en el servidor DB2.

Mediante la utilización de este procedimiento, puede establecer el ID de usuario del cliente, el nombre de aplicación, el nombre de estación de trabajo, la información de contabilidad o la información de carga de trabajo en el servidor DB2. La invocación de este procedimiento cambia los valores almacenados para esta conexión de los campos de información de cliente de supervisor de procesador de transacciones (TP) pertinentes y los valores de registro especial.

Los campos de información de cliente se utilizan en el servidor DB2 para determinar la identidad de la aplicación o del usuario que utiliza actualmente la conexión. Los campos de información de cliente para una conexión se tienen en cuenta durante la evaluación de carga de trabajo de DB2 y también aparecen en cualquier instantánea de aplicación o registros de auditoría de DB2 que se genere para esta conexión.

A diferencia de la API sqleseti, este procedimiento no establece información de cliente en el cliente sino que establece los atributos de cliente correspondientes en el servidor DB2. Por consiguiente, no se puede utilizar la API sqleqry para consultar la información de cliente que se establece en el servidor DB2 utilizando este procedimiento. Si una aplicación utiliza la API sqleseti para cambiar la información del cliente, los nuevos valores cambiarán el valor en el servidor DB2.

Si se utiliza la API sqleseti para cambiar el ID de usuario o el nombre de la aplicación sin cambiar la información de contabilidad, también se restaurará la información de contabilidad en el servidor DB2 en el valor que tenga la información de contabilidad en el cliente.

Los valores de datos proporcionados con el procedimiento se convierten en la página de códigos de base de datos apropiada antes de almacenarse en los campos de Supervisor de TP relacionados o los registros especiales. Los valores de datos que sobrepasan el tamaño máximo soportado después de la conversión a la página de códigos de la base de datos se truncarán antes de almacenarse en el servidor. Los campos de Supervisor de TP y los registros especiales devolverán los valores truncados cuando se consulten esos valores almacenados.

El procedimiento WLM\_SET\_CLIENT\_INFO no está bajo el control de la transacción y los cambios en la información de cliente efectuados por el procedimiento son independientes de la acción de confirmar o retrotraer unidades de trabajo. Sin embargo, debido a que la reevaluación de la carga de trabajo se produce al principio de la siguiente unidad de trabajo para cada aplicación, debe emitir una sentencia COMMIT o bien ROLLBACK para que los cambios en la información de cliente sean efectivos.

## Sintaxis

```
►►—WLM_SET_CLIENT_INFO—(—id_usuario_cliente—,—nombre_estación_trabajo_cliente—,—  
►—nombre_apl_cliente—,—serie_cont_cliente—,—carga_trabajo_cliente—)————►►
```

El esquema es SYSPROC.

## Parámetros del procedimiento

### *id\_usuario\_cliente*

Argumento de entrada de tipo VARCHAR(255) que especifica el ID de usuario para el cliente. Si se especifica NULL, el valor se queda como está. Si se especifica una serie vacía, que es el valor por omisión, el ID de usuario del cliente se restablece en el valor por omisión, que es un espacio en blanco.

### *nombre\_estación\_trabajo\_cliente*

Argumento de entrada de tipo VARCHAR(255) que especifica el nombre de estación de trabajo para el cliente. Si se especifica NULL, el valor se queda como está. Si se especifica una serie vacía, que es el valor por omisión, el nombre de estación de trabajo del cliente se restablece en el valor por omisión, que es un espacio en blanco.

### *nombre\_apl\_cliente*

Argumento de entrada de tipo VARCHAR(255) que especifica el nombre de aplicación del cliente. Si se especifica NULL, el valor se queda como está. Si se especifica una serie vacía, que es el valor por omisión, el nombre de aplicación del cliente se restablece en el valor por omisión, que es un espacio en blanco.

### *serie\_cont\_cliente*

Argumento de entrada de tipo VARCHAR(200) que especifica la serie de contabilidad para el cliente. Si se especifica NULL, el valor se queda como está. Si se especifica una serie vacía, que es el valor por omisión, la serie de contabilidad del cliente se restablece en el valor por omisión, que es un espacio en blanco.

### *carga\_trabajo\_cliente*

Argumento de entrada de tipo VARCHAR(255) que especifica la modalidad de asignación de carga de trabajo para el cliente. Si se especifica NULL, el valor se queda como está. Los valores son los siguientes:

#### **SYSDEFAULTADMWORKLOAD**

Especifica que la conexión de base de datos se asignará a SYSDEFAULTADMWORKLOAD, lo que permitirá que los usuarios con autorización ACCESSCTRL, DATAACCESS, DBADM, SECADM o WLMADM ignoren la evaluación de carga de trabajo normal.

#### **AUTOMATIC**

Especifica que la base de datos se asignará a una carga de trabajo escogida mediante la evaluación de la carga de trabajo que el servidor efectúa automáticamente.

**Nota:** El argumento *carga\_trabajo\_cliente* es sensible a las mayúsculas y minúsculas.

## **Autorización**

Para ejecutar la rutina se requiere una de las autorizaciones siguientes:

- Privilegio EXECUTE para la rutina
- Autorización DATAACCESS
- Autorización DBADM
- Autorización SQLADM
- Autorización WLMADM

## **Privilegio PUBLIC por omisión**

Ninguna

## **Ejemplos**

La siguiente invocación de procedimiento establece el ID de usuario, el nombre de estación de trabajo, el nombre de aplicación, la serie de contabilidad y la modalidad de asignación de carga de trabajo para el cliente:

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user', 'machine.torolab.ibm.com',  
    'auditor', 'Departamento contabilidad', 'AUTOMATIC')
```

La siguiente invocación de procedimiento establece el ID de usuario en db2user2 para cliente sin establecer los demás atributos de cliente:

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user2', NULL, NULL, NULL, NULL)
```

La siguiente invocación de procedimiento restablece el ID de usuario para el cliente en un espacio en blanco sin modificar los valores de los demás atributos de cliente:

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('', NULL, NULL, NULL, NULL)
```

## **Notas de uso**

Si la entrada que especifica para cualquier parámetro de procedimiento excede la longitud de campo especificada, el campo de entrada se trunca y el procedimiento se ejecuta utilizando las entradas truncadas.

Los campos de entrada que contienen comillas simples no están soportados y provocan un error.

---

## Elementos de supervisor de gestión de carga de trabajo

Los elementos de supervisor siguientes proporcionan información sobre actividades, violaciones de umbral y estadísticas sobre gestión de carga de trabajo.

### Elemento de supervisor `act_cpu_time_top` – Tiempo de CPU de actividad máximo

Marca de límite superior del tiempo de procesador empleado por las actividades en todos los niveles de anidamiento en una clase de servicio, carga de trabajo o clase de trabajo. Este valor se expresa en microsegundos.

El elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la clase de servicio o la carga de trabajo en la que se ejecuta la actividad se ha establecido en NONE. Las actividades contribuyen a alcanzar esta marca de límite superior sólo cuando se habilitan las métricas de petición. Si la recopilación de métricas de actividad no está habilitada, se devuelve un valor 0.

Para las clases de servicio, cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, sólo se actualiza la marca de límite superior de la subclase de servicio donde se lleva a cabo una actividad, siempre que se alcance una nueva marca de límite superior. Ello no afecta a las marcas de límite superior `act_cpu_time_top` de las subclases de servicio con las que está correlacionada una actividad pero en las que no se lleva a cabo.

*Tabla 79. Información de supervisión de sucesos*

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	Siempre se recopilan
Estadísticas	event_wcstats	Siempre se recopilan
Estadísticas	event_wlstats	Siempre se recopilan

### Uso

Utilice este elemento para determinar la cantidad más alta de tiempo de procesador empleado por una actividad en un miembro para una clase de servicio, una carga de trabajo o una clase de trabajo durante el intervalo de tiempo de recopilación.

### Elemento de supervisor `act_exec_time` - Tiempo de ejecución de la actividad

Tiempo empleado en la ejecución en este miembro, en microsegundos. En el caso de los cursores, el tiempo de ejecución es el tiempo combinado para la apertura, las captaciones y el cierre. El tiempo durante el que el cursor está inactivo no cuenta como tiempo de ejecución. En el caso de las rutinas, el tiempo de ejecución es el que transcurre entre el inicio y el final de la invocación de la rutina. Las vidas útiles de los cursores que deja abiertos una rutina (para devolver un conjunto de resultados) después de que finalice la rutina no cuentan como tiempo de ejecución de la rutina. Para el resto de actividades, el tiempo de ejecución es la diferencia entre la hora de inicio y la hora de finalización. En ningún caso, el tiempo de ejecución incluye el tiempo de inicialización o de colocación en cola.

Tabla 80. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Este elemento se puede utilizar solo para saber el tiempo que DB2 ha dedicado a la ejecución de la actividad en cada miembro. Este elemento también se puede utilizar junto con los elementos de supervisor **time\_started** y **time\_completed** en el miembro coordinador para calcular el tiempo de inactividad de las actividades de cursor. Puede utilizar la fórmula siguiente:

Tiempo de inactividad de cursor = (time\_completed - time\_started) - act\_exec\_time

## Elemento de supervisor act\_remapped\_in – Actividades incluidas en una nueva correlación

Recuento del número de actividades que se deben incluir en la nueva correlación con esta subclase de servicio desde el último restablecimiento.

Tabla 81. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-

## Uso

Utilice este recuento para determinar si la inclusión de actividades en la nueva correlación con la subclase de servicio tiene lugar de la forma deseada.

## Elemento de supervisor act\_remapped\_out – Actividades excluidas en una nueva correlación

Recuento del número de actividades que se deben excluir de la nueva correlación con esta subclase de servicio desde el último restablecimiento.

Tabla 82. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-

## Uso

Utilice este recuento para determinar si la exclusión de actividades en la nueva correlación con la subclase de servicio tiene lugar de la forma deseada.

## Elemento de supervisor act\_rows\_read\_top – Lectura máxima de filas de actividad

Marca de límite superior para el número de filas leídas por actividades en todos los niveles de anidamiento en una clase de servicio, carga de trabajo o clase de trabajo.

El elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la clase de servicio o la carga de trabajo en la que se ejecuta la

actividad se ha establecido en NONE. Las actividades contribuyen a alcanzar esta marca de límite superior sólo cuando se habilitan las métricas de petición. Si la recopilación de métricas de actividad no está habilitada, se devuelve un valor 0.

Para las clases de servicio, cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, sólo se actualiza la marca de límite superior `act_rows_read_top` de la subclase de servicio donde se lleva a cabo una actividad, siempre que se alcance una nueva marca de límite superior. Ello no afecta a las marcas de límite `act_rows_read_top` de las subclases de servicio con las que está correlacionada una actividad pero que no completa.

*Tabla 83. Información de supervisión de sucesos*

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	Siempre se recopilan
Estadísticas	event_wcstats	Siempre se recopilan
Estadísticas	event_wlstats	Siempre se recopilan

## Uso

Utilice este elemento para determinar el número más alto de filas leídas por una actividad en un miembro para una clase de servicio, una carga de trabajo o una clase de trabajo durante el intervalo de tiempo de recopilación.

## Elemento de supervisor `act_throughput` - Rendimiento de las actividades

Ritmo al que se llevan a cabo las actividades de coordinador en cualquier nivel de anidamiento. Se mide en actividades de coordinador por segundo.

*Tabla 84. Información de supervisión de funciones de tabla*

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

*Tabla 85. Información de supervisión de sucesos*

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento <code>details_xml</code> )	Siempre se recopilan
Estadísticas	event_wlstats (notificadas en el documento <code>details_xml</code> )	Siempre se recopilan

## Uso

Cuando lo devuelve la función `WLM_GET_SERVICE_SUBCLASS_STATS` o la función `WLM_GET_WORKLOAD_STATS`, este elemento de supervisor representa el rendimiento de las actividades desde el último restablecimiento de las estadísticas.

Cuando lo devuelve la función `MON_SAMPLE_SERVICE_CLASS_METRICS` o la función `MON_SAMPLE_WORKLOAD_METRICS`, este elemento de supervisor representa el rendimiento de las actividades desde que se ha ejecutado la función.

## Elemento de supervisor `act_total` - Total de actividades

Número total de actividades en cualquier nivel de anidamiento que tenían acciones de trabajo correspondientes a la clase de trabajo especificada que se les ha aplicado desde el último restablecimiento.

Tabla 86. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
<code>WLM_GET_WORK_ACTION_SET_STATS</code> - Devolución de estadísticas de conjunto de acciones de trabajo	ACTIVITY METRICS BASE

Tabla 87. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	<code>event_wcstats</code>	-

## Uso

Cada vez que una actividad tiene una o más acciones de trabajo asociadas a una clase de trabajo que se le ha aplicado, se actualiza un contador para la clase de trabajo. Este contador se expone mediante el elemento de supervisor `act_total`. El contador se puede utilizar para evaluar la eficacia del conjunto de acciones de trabajo (por ejemplo, la cantidad de actividades que tienen acciones aplicadas). También se puede utilizar para conocer los distintos tipos de actividades en el sistema.

## Elemento de supervisor `activate_timestamp` - Indicación de la fecha y la hora de activación

Hora a la que se ha activado un supervisor de sucesos.

Tabla 88. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividad	<code>event_activity</code>	-
Actividad	<code>event_activitystmt</code>	-
Actividad	<code>event_activityvals</code>	-
Violaciones de umbral	<code>event_thresholdviolations</code>	-

## Uso

Utilice este elemento para correlacionar la información devuelta por los tipos de sucesos mencionados anteriormente.

## Elemento de supervisor `activity_collected` - Actividad recopilada

Este elemento indica si se deben recopilar los registros de supervisor de sucesos de actividad de un umbral que se ha violado.

Tabla 89. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-

## Uso

Utilice este elemento para determinar si se debe esperar que se grabe un suceso de la actividad que ha violado el umbral en el supervisor de sucesos de actividad.

Cuando una actividad finaliza o se aborta y el supervisor de sucesos de actividad está activo en ese momento, si el valor de ese elemento de supervisor es 'Y', se recopilará la actividad que ha violado este umbral. Si el valor de este elemento de supervisor es 'N', no se recopilará.

## Elemento de supervisor `activity_id` - ID de actividad

Contador que identifica de forma exclusiva una actividad de una aplicación dentro de una unidad de trabajo determinada.

Tabla 90. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 91. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Bloqueo	-	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan
Actividades	event_activitystmt	Siempre se recopilan
Actividades	event_activityvals	Siempre se recopilan
Actividades	event_activitymetrics	ACTIVITY METRICS BASE

Tabla 91. Información de supervisión de sucesos (continuación)

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de historial de actividad para analizar el comportamiento de una actividad.

Para identificar de forma exclusiva una actividad fuera de su unidad de trabajo, utilice la combinación de **activity\_id** y **uow\_id** más uno de los elementos de supervisor siguientes: **appl\_id** o **agent\_id**.

## Elemento de supervisor **activity\_secondary\_id** - ID secundario de actividad

El valor de este elemento aumenta cada vez que se graba un registro de actividad para la misma actividad. Por ejemplo, si se graba una actividad como resultado de haber llamado al procedimiento **WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS** y una segunda vez cuando finaliza la actividad, el elemento tendría un valor 0 para el primer registro y 1 para el segundo registro.

Tabla 92. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	-
Actividades	event_activitystmt	-
Actividades	event_activityvals	-
Actividades	event_activitymetrics	ACTIVITY METRICS BASE

## Uso

Utilice este elemento con los elementos de supervisor **activity\_id**, **uow\_id** y **appl\_id** para identificar de forma exclusiva los registros de actividad cuando se ha grabado información sobre la misma actividad en el supervisor de sucesos de actividades varias veces.

Por ejemplo, la información sobre una actividad se enviaría al supervisor de sucesos de actividades dos veces en el caso siguiente:

- El procedimiento almacenado **WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS** se ha utilizado para capturar información sobre la actividad mientras se estaba ejecutando.
- Se ha recopilado información sobre la actividad cuando la actividad se ha completado, porque se ha especificado la cláusula **COLLECT ACTIVITY DATA** en la clase de servicio con la que está asociada la actividad.

## Elemento de supervisor activity\_type - Tipo de actividad

Tipo de la actividad.

Tabla 93. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

Tabla 94. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

### Uso

Los valores posibles son:

- LOAD
- READ\_DML
- WRITE\_DML
- DDL
- CALL
- OTHER

Se devuelve el valor OTHER para las sentencias SET que no ejecutan SQL (por ejemplo, SET registro especial o SET EVENT MONITOR STATE) y la sentencia LOCK TABLE.

## Elemento de supervisor agg\_temp\_tablespace\_top - Espacio de tablas temporal agregado máximo

Marca de límite superior en KB de uso de espacio de tablas temporal agregado de las actividades de DML en todos los niveles de anidamiento en una clase de servicio. El agregado se calcula sumando el uso de espacio de tablas temporal en todas las actividades de la subclase de servicio y esta marca de límite superior representa el valor más alto que alcanza este agregado desde el último restablecimiento. Este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. Debe haberse definido y habilitado un umbral AGGSQLTEMPSPACE para una subclase de servicio, como mínimo, en la misma superclase que la subclase a la que pertenece este registro; en caso contrario, se devuelve un valor 0.

Tabla 95. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	Siempre se recopilan

### Uso

Utilice este elemento para determinar el uso de espacio de tablas temporal del sistema de las actividades de DML agregado más alto que se ha alcanzado en un

miembro para una subclase de servicio en el intervalo de tiempo de recopilación.

## Elemento de supervisor **arm\_correlator** - Correlacionador de medición de respuestas de aplicación

Identificador de una transacción en el estándar ARM (medición de respuestas de aplicación).

Tabla 96. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	-

### Uso

Este elemento se puede utilizar para enlazar una actividad recopilada por el supervisor de sucesos de actividades con las aplicaciones asociadas a la actividad, si tales aplicaciones también admiten el estándar ARM (medición de respuestas de aplicación).

## Elemento de supervisor **bin\_id** - Identificador de intervalo de histograma

Identificador de un intervalo de histograma. El **bin\_id** es exclusivo dentro de un histograma.

Tabla 97. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_histogrambin	-

### Uso

Utilice este elemento para distinguir intervalos dentro del mismo histograma.

## Elemento de supervisor **bottom** - Fin de intervalo de histograma

Extremo final exclusivo del rango de un intervalo de histograma. El valor de este elemento de supervisor también es el principio inclusivo del rango del intervalo de histograma anterior, si lo hay.

Tabla 98. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_histogrambin	-

### Uso

Utilice este elemento con el elemento **top** correspondiente para determinar el rango de un intervalo dentro de un histograma.

## Elemento de supervisor concurrent\_act\_top - Actividad simultánea máxima

Marca de límite superior para las actividades simultáneas (en cualquier nivel de anidamiento) en una subclase de servicio desde el último restablecimiento.

**Nota:** Este elemento supervisa la ejecución simultánea más alta de todas las actividades, incluidas las actividades que no participan en el umbral CONCURRENTDBCOORDACTIVITIES. Por ejemplo, aunque las sentencias CALL no cuentan en términos de la simultaneidad que aplica el umbral CONCURRENTDBCOORDACTIVITIES, se incluyen en la medición de la marca de límite superior de la actividad simultánea.

Tabla 99. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD _OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE

Tabla 100. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-

### Uso

Utilice este elemento para conocer la simultaneidad de actividades más alta (incluidas las actividades anidadas) a la que se llega en un miembro para una subclase de servicio en el intervalo de tiempo de recopilación.

## Elemento de supervisor concurrent\_connection\_top - Máximo de conexiones simultáneas

Marca de límite superior de conexiones de coordinador simultáneas en esta clase de servicio desde el último restablecimiento. Este campo tiene el mismo valor en cada subclase de la misma superclase.

Tabla 101. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_SUPERCLASS_STATS - Devolución de estadísticas de superclases de servicio	ACTIVITY METRICS BASE

Tabla 102. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-

## Uso

Este elemento puede ser útil para determinar dónde situar los umbrales de simultaneidad de conexiones al mostrar dónde se encuentra la marca de límite superior actual. También es útil para verificar que dicho umbral se ha configurado correctamente y está haciendo su trabajo.

## Elemento de supervisor `concurrent_wlo_act_top` - Actividad de WLO simultánea máxima

Marca de límite superior para las actividades simultáneas (en el nivel de anidamiento) de cualquier instancia de esta carga de trabajo desde el último restablecimiento.

Tabla 103. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 104. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wlstats	-

## Uso

Utilice este elemento para conocer el número más alto de actividades simultáneas que se ha alcanzado en un miembro para cualquier instancia de esta carga de trabajo en el intervalo de tiempo de la recopilación.

## Elemento de supervisor `concurrent_wlo_top` - Máximo de instancias de carga de trabajo simultáneas

Marca de límite superior para las instancias simultáneas de una carga de trabajo desde el último restablecimiento.

Tabla 105. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 106. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wlstats	-
Estadísticas	event_scstats	-

## Uso

Utilice este elemento para conocer la simultaneidad más alta de instancias de carga de trabajo que se ha alcanzado en un miembro para una carga de trabajo en el intervalo de tiempo de la recopilación.

## Elemento de supervisor concurrentdbcoordactivities\_db\_threshold\_id - ID de umbral de base de datos de actividades simultáneas de coordinador de base de datos

ID del umbral de la base de datos CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

Tabla 107. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para saber qué umbral de base de datos CONCURRENTDBCOORDACTIVITIES, si lo hubiera, se ha aplicado a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_db\_threshold\_queued - Umbral de base de datos de actividades simultáneas de coordinador de base de datos en cola

Este elemento de supervisor devuelve 'Yes' para indicar que el umbral de la base de datos CONCURRENTDBCOORDACTIVITIES ha puesto en cola la actividad. 'No' indica que la actividad no se ha colocado en cola.

Tabla 108. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para saber si el umbral de la base de datos CONCURRENTDBCOORDACTIVITIES aplicado a la actividad la ha puesto en cola.

### **Elemento de supervisor concurrentdbcoordactivities\_db\_threshold\_value - Valor de umbral de base de datos de actividades simultáneas de coordinador de base de datos**

Este elemento de supervisor devuelve el límite superior del umbral de la base de datos CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

*Tabla 109. Información de supervisión de funciones de tabla*

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para conocer el valor del umbral de la base de datos CONCURRENTDBCOORDACTIVITIES aplicado a la actividad, si lo hay.

### **Elemento de supervisor concurrentdbcoordactivities\_db\_threshold\_violated - Umbral violado de base de datos de actividades simultáneas de coordinador de base de datos**

Este elemento de supervisor devuelve 'Yes' para indicar que la actividad ha violado el umbral de la base de datos CONCURRENTDBCOORDACTIVITIES. 'No' indica que la actividad aún no ha violado el umbral.

*Tabla 110. Información de supervisión de funciones de tabla*

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para determinar si la actividad ha violado el umbral de la base de datos CONCURRENTDBCOORDACTIVITIES que se aplicaba a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_subclass\_threshold\_id - ID de umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos

Este elemento de supervisor devuelve el ID de la subclase de servicio CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

Tabla 111. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

### Uso

Utilice este elemento para saber qué umbral de subclase de servicio CONCURRENTDBCOORDACTIVITIES, si lo hubiera, se ha aplicado a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_subclass\_threshold\_queued - Umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos en cola

Este elemento de supervisor devuelve 'Yes' para indicar que el umbral de la subclase de servicio CONCURRENTDBCOORDACTIVITIES ha puesto en cola la actividad. 'No' indica que la actividad no se ha colocado en cola.

Tabla 112. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

### Uso

Utilice este elemento para saber si el umbral de la subclase de servicio CONCURRENTDBCOORDACTIVITIES aplicado a la actividad la ha puesto en cola.

## Elemento de supervisor concurrentdbcoordactivities\_subclass\_threshold\_value - Valor de umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos

Este elemento de supervisor devuelve el límite superior del umbral de subclase de servicio CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

Tabla 113. Información de supervisión de funciones de tabla

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla WLM_GET_ACTIVITY_DETAILS - Devolver información detallada sobre una actividad específica	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para entender el valor del umbral de subclase de servicio CONCURRENTDBCOORDACTIVITIES aplicado a la actividad, si lo hay.

## Elemento de supervisor concurrentdbcoordactivities\_subclass\_threshold\_violated - Umbral de subclase de servicio de actividades simultáneas de coordinador de base de datos violado

Este elemento de supervisor devuelve 'Yes' para indicar que la actividad ha violado el umbral de la subclase de servicio CONCURRENTDBCOORDACTIVITIES. 'No' indica que la actividad aún no ha violado el umbral.

Tabla 114. Información de supervisión de funciones de tabla

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para determinar si la actividad ha violado el umbral de la subclase de servicio CONCURRENTDBCOORDACTIVITIES que se aplicaba a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_superclass\_threshold\_id - ID de umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos

ID del umbral CONCURRENTDBCOORDACTIVITIES\_SUPERCLASS que se ha aplicado a la actividad.

Tabla 115. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para saber qué umbral de superclase de servicio CONCURRENTDBCOORDACTIVITIES, si lo hubiera, se ha aplicado a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_superclass\_threshold\_queued - Umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos en cola

Este elemento de supervisor devuelve 'Yes' para indicar que el umbral de la superclase de servicio CONCURRENTDBCOORDACTIVITIES ha puesto en cola la actividad. 'No' indica que la actividad no se ha colocado en cola.

Tabla 116. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para saber si el umbral de la superclase de servicio CONCURRENTDBCOORDACTIVITIES aplicado a la actividad la ha puesto en cola.

## Elemento de supervisor concurrentdbcoordactivities\_superclass\_threshold\_value - Valor de umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos

Límite superior del umbral de superclase de servicio CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

Tabla 117. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para entender el valor de umbral de superclase de servicio CONCURRENTDBCOORDACTIVITIES aplicado a la actividad, si lo hay.

### Elemento de supervisor concurrentdbcoordactivities\_superclass\_threshold\_violated - Umbral de superclase de servicio de actividades simultáneas de coordinador de base de datos violado

Este elemento de supervisor devuelve 'Yes' para indicar que la actividad ha violado el umbral de la superclase de servicio CONCURRENTDBCOORDACTIVITIES. 'No' indica que la actividad aún no ha violado el umbral.

Tabla 118. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para determinar si la actividad ha violado el umbral de la superclase de servicio CONCURRENTDBCOORDACTIVITIES que se aplicaba a la actividad.

### Elemento de supervisor concurrentdbcoordactivities\_wl\_was\_threshold\_id - ID de umbral de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos

Identificador del umbral de conjunto de acciones de trabajo de carga de trabajo de CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

Tabla 119. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

## Uso

Utilice este elemento para saber qué umbral de conjunto de acciones de trabajo de carga de trabajo de CONCURRENTDBCOORDACTIVITIES, si lo hubiera, se ha aplicado a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_wl\_was\_threshold\_queued - Umbral en cola de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos

Este elemento de supervisor devuelve 'Yes' para indicar que el umbral de conjunto de acciones de trabajo de carga de trabajo de CONCURRENTDBCOORDACTIVITIES ha puesto en cola a la actividad. 'No' indica que la actividad no se ha colocado en cola.

Tabla 120. Información de supervisión de funciones de tabla

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

### Uso

Utilice este elemento para saber si el umbral de conjunto de acciones de trabajo de carga de trabajo CONCURRENTDBCOORDACTIVITIES aplicado a la actividad la ha puesto en cola.

## Elemento de supervisor concurrentdbcoordactivities\_wl\_was\_threshold\_value - Valor de umbral de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos

Límite superior del umbral de conjunto de acciones de trabajo de carga de trabajo de CONCURRENTDBCOORDACTIVITIES que se ha aplicado a la actividad.

Tabla 121. Información de supervisión de funciones de tabla

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

### Uso

Utilice este elemento para entender el valor del umbral de conjunto de acciones de trabajo de carga de trabajo de CONCURRENTDBCOORDACTIVITIES aplicado a la actividad.

## Elemento de supervisor concurrentdbcoordactivities\_wl\_was\_threshold\_violated - Umbral violado de conjunto de acciones de trabajo de carga de trabajo de actividades simultáneas de coordinador de base de datos

Este elemento de supervisor devuelve 'Yes' para indicar que la actividad ha violado el umbral de conjunto de acciones de trabajo de carga de trabajo de CONCURRENTDBCOORDACTIVITIES. 'No' indica que la actividad todavía no ha violado el umbral.

Tabla 122. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

### Uso

Utilice este elemento para determinar si la actividad ha violado el umbral de conjunto de acciones de trabajo de carga de trabajo CONCURRENTDBCOORDACTIVITIES que se aplicaba a la actividad.

## Elemento de supervisor coord\_act\_aborted\_total - Total de actividades de coordinador abortadas

Número total de actividades de coordinador en cualquier nivel de anidamiento que se han llevado a cabo con errores desde el último restablecimiento. En el caso de las clases de servicio, el valor se actualiza cuando se lleva a cabo la actividad. En el caso de las cargas de trabajo, el valor se actualiza con cada instancia de carga de trabajo al final de su unidad de trabajo.

En el caso de las clases de servicio, si vuelve a correlacionar una actividad con una subclase diferente mediante una acción REMAP ACTIVITY antes de que se aborte, esta actividad contará únicamente para el total de la subclase en la que se aborta.

Tabla 123. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 124. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice este elemento para saber si las actividades del sistema se están llevando a cabo satisfactoriamente. Las actividades pueden abortarse a causa de una cancelación, de errores o de umbrales reactivos.

## Elemento de supervisor coord\_act\_completed\_total - Total de actividades de coordinador completadas

Número total de actividades de coordinador en cualquier nivel de anidamiento que se han completado desde el último restablecimiento. En el caso de las clases de servicio, el valor se actualiza cuando se lleva a cabo la actividad. En el caso de las cargas de trabajo, el valor se actualiza con cada instancia de carga de trabajo al final de su unidad de trabajo.

En el caso de las clases de servicio, si vuelve a correlacionar una actividad con una subclase diferente mediante una acción REMAP ACTIVITY antes de que se complete, esta actividad contará únicamente para el total de la subclase en la que se complete.

Tabla 125. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD _OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 126. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wlstats	-
Estadísticas	event_scstats	-

## Uso

Este elemento se puede utilizar para determinar el rendimiento de las actividades en el sistema o como ayuda para calcular el promedio de vida útil de las actividades entre varios miembros.

## Elemento de supervisor coord\_act\_est\_cost\_avg - Promedio de coste estimado de actividad de coordinador

Promedio aritmético de los costes estimados de las actividades de DML de coordinador en el nivel de anidamiento 0 asociado con esta subclase de servicio o clase de trabajo desde el último restablecimiento. Si se ha desbordado el promedio del que se hace un seguimiento interno, se devuelve el valor -2. En el caso de las subclases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE o BASE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA EXTENDED para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE o BASE. Las unidades son los timerons.

En el caso de las clases de servicio, el coste estimado de una actividad sólo cuenta para la subclase de servicio en la que la actividad entra en el sistema. Cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, el promedio coord\_act\_est\_cost\_avg de la subclase de servicio con la que se vuelve a correlacionar una actividad no se ve afectado por ello.

Tabla 127. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	Siempre se recopilan
Estadísticas	event_wcstats	Siempre se recopilan
Estadísticas	event_wlstats	Siempre se recopilan

### Uso

Utilice esta estadística para determinar el promedio aritmético de los costes estimados de las actividades de DML de coordinador en el nivel de anidamiento 0 que están asociados a esta subclase de servicio, carga de trabajo o clase de trabajo completadas o abortadas desde el último restablecimiento de las estadísticas.

Este promedio también se puede utilizar para determinar si la plantilla de histograma utilizada para el histograma de coste estimado de la actividad es adecuada. Calcule el promedio de coste estimado de actividad a partir del histograma de coste estimado de la actividad. Compare el promedio calculado con este elemento de supervisor. Si el promedio calculado se desvía del promedio real que indica este elemento de supervisor, considere la posibilidad de modificar la plantilla del histograma de coste estimado de la actividad utilizando un conjunto de valores de intervalo que sea más apropiado para los datos en cuestión.

## Elemento de supervisor coord\_act\_exec\_time\_avg - Promedio de tiempo de ejecución de las actividades de coordinador

Promedio aritmético de los tiempos de ejecución de las actividades de coordinador en el nivel de anidamiento 0 asociado con esta subclase de servicio o clase de trabajo desde el último restablecimiento. Si se ha desbordado el promedio del que se hace un seguimiento interno, se devuelve el valor -2. En el caso de las subclases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se

ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE. Las unidades son milisegundos.

En el caso de clases de servicio, cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, el promedio coord\_act\_exec\_time\_avg de subclases de servicio con las que está correlacionada una actividad pero en las que no se lleva a cabo no se ve afectado por ello.

*Tabla 128. Información de supervisión de funciones de tabla*

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	COLLECT AGGREGATE ACTIVITY DATA
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	COLLECT AGGREGATE ACTIVITY DATA

*Tabla 129. Información de supervisión de sucesos*

<b>Tipo de suceso</b>	<b>Agrupación de datos lógicos</b>	<b>Conmutador de supervisor</b>
Estadísticas	event_scstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice esta estadística para determinar el promedio aritmético del tiempo de ejecución de las actividades de coordinador asociadas con una subclase de servicio, una carga de trabajo o una clase de trabajo que se ha completado o abortado.

Este promedio también se puede utilizar para determinar si la plantilla de histograma utilizada para el histograma de tiempo de ejecución de las actividades es adecuada. Calcule el promedio de tiempo de ejecución de actividades a partir del histograma de tiempo de ejecución de actividades. Compare el promedio calculado con este elemento de supervisor. Si el promedio calculado se desvía del promedio real que indica este elemento de supervisor, considere la posibilidad de modificar la plantilla del histograma de tiempo de ejecución de actividades utilizando un conjunto de valores de intervalo que sea más apropiado para los datos en cuestión.

## **Elemento de supervisor coord\_act\_interarrival\_time\_avg - Promedio de tiempo de llegada de las actividades de coordinador**

Promedio aritmético del tiempo transcurrido entre las llegadas de actividades de coordinador en el nivel de anidamiento 0 asociado con esta subclase de servicio o clase de trabajo desde el último restablecimiento. Si se ha desbordado el promedio del que se hace un seguimiento interno, se devuelve el valor -2. En el caso de las subclases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT

AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE o BASE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA EXTENDED para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE o BASE. Las unidades son milisegundos.

Para las clases de servicio, el promedio de tiempo entre llegadas se calcula para las subclases por medio de las cuales las actividades entran en el sistema. Cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, el promedio coord\_act\_interarrival\_time\_avg de la subclase de servicio con la que se vuelve a correlacionar una actividad no se ve afectado por ello.

Tabla 130. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice esta estadística para determinar el promedio aritmético entre llegadas de actividades de coordinador en el nivel de anidamiento 0 asociado a esta subclase de servicio, carga de trabajo o clase de trabajo.

El tiempo entre llegadas se puede utilizar para determinar el ritmo de llegadas, que es lo contrario al tiempo entre llegadas. Este promedio también se puede utilizar para determinar si la plantilla de histograma utilizada para el histograma de tiempo entre llegadas de actividades es adecuada. Calcule el promedio de tiempo entre llegadas de actividades a partir del histograma de tiempo entre llegadas de actividades. Compare el promedio calculado con este elemento de supervisor. Si el promedio calculado se desvía del promedio real que indica este elemento de supervisor, considere la posibilidad de modificar la plantilla del histograma de tiempo entre llegadas de actividades utilizando un conjunto de valores de intervalo que sea más apropiado para los datos en cuestión.

## Elemento de supervisor coord\_act\_lifetime\_avg - Promedio de vida útil de las actividades de coordinador

Promedio aritmético de la vida útil de las actividades de coordinador en el nivel de anidamiento 0 asociado con esta subclase de servicio, carga de trabajo o clase de trabajo desde el último restablecimiento. Si se ha desbordado el promedio del que se hace un seguimiento interno, se devuelve el valor -2. En el caso de las subclases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE. Las unidades son milisegundos.

En el caso de clases de servicio, cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, sólo se ve afectado el promedio coord\_act\_lifetime\_avg de la clase de servicio donde se lleva a cabo la actividad.

*Tabla 131. Información de supervisión de funciones de tabla*

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	COLLECT AGGREGATE ACTIVITY DATA
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	COLLECT AGGREGATE ACTIVITY DATA

*Tabla 132. Información de supervisión de sucesos*

<b>Tipo de suceso</b>	<b>Agrupación de datos lógicos</b>	<b>Conmutador de supervisor</b>
Estadísticas	event_scstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice esta estadística para determinar el promedio aritmético de la vida útil de las actividades de coordinador asociadas con una subclase de servicio, una carga de trabajo o una clase de trabajo que se ha completado o abortado.

Esta estadística también se puede utilizar para determinar si la plantilla de histograma utilizada para el histograma de vida útil de las actividades es adecuada. Calcule el promedio de vida útil de las actividades a partir del histograma de vida de útil de las actividades. Compare el promedio calculado con este elemento de supervisor. Si el promedio calculado se desvía del promedio real que indica este elemento de supervisor, considere la posibilidad de modificar la plantilla del histograma de vida útil de las actividades utilizando un conjunto de valores de intervalo que sea más apropiado para los datos en cuestión.

## Elemento de supervisor coord\_act\_lifetime\_top - Vida útil máxima de las actividades de coordinador

Marca de límite superior para la vida útil de las actividades de coordinador, contada en todos los niveles de anidamiento. Las unidades son milisegundos. En el caso de las clases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE.

Para utilizar de forma efectiva esta estadística con las clases de servicio cuando también vuelva a correlacionar actividades entre subclases de servicio con una

acción REMAP ACTIVITY, debe agregar la marca de límite superior coord\_act\_lifetime\_top de cualquier subclase de servicio concreta con las otras subclases afectadas por los mismos umbrales de nueva correlación. Esto se debe a que una actividad se llevará a cabo después de que se haya vuelto a correlacionar con una subclase de servicio distinta mediante un umbral de nueva correlación y el tiempo que la actividad emplea en otras subclases de servicio antes de que se vuelvan a correlacionar sólo cuenta para la clase de servicio en la que se lleva a cabo.

Tabla 133. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	COLLECT AGGREGATE ACTIVITY DATA
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	COLLECT AGGREGATE ACTIVITY DATA

Tabla 134. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wcstats	-
Estadísticas	event_scstats	-
Estadísticas	event_wlstats	-

## Uso

Este elemento se puede utilizar para ayudar a determinar si los umbrales sobre la vida útil de las actividades son efectivos y también puede ayudar a determinar cómo configurar esos umbrales.

## Elemento de supervisor coord\_act\_queue\_time\_avg - Promedio de tiempo en cola de las actividades de coordinador

Promedio aritmético del tiempo en cola de las actividades de coordinador en el nivel de anidamiento 0 asociado con esta subclase de servicio o clase de trabajo desde el último restablecimiento. Si se ha desbordado el promedio del que se hace un seguimiento interno, se devuelve el valor -2. En el caso de las subclases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE. Las unidades son milisegundos.

En el caso de las clases de servicio, el tiempo en cola sólo cuenta para la subclase de servicio en la que se ha llevado a cabo o se ha abortado la actividad. Cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción

REMAP ACTIVITY, el promedio coord\_act\_queue\_time\_avg de subclases de servicio con las que está correlacionada una actividad pero en las que no se lleva a cabo no se ve afectado por ello.

*Tabla 135. Información de supervisión de funciones de tabla*

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	COLLECT AGGREGATE ACTIVITY DATA
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	COLLECT AGGREGATE ACTIVITY DATA

*Tabla 136. Información de supervisión de sucesos*

<b>Tipo de suceso</b>	<b>Agrupación de datos lógicos</b>	<b>Conmutador de supervisor</b>
Estadísticas	event_scstats	Siempre se recopilan
Estadísticas	event_wcstats	Siempre se recopilan
Estadísticas	event_wlstats	Siempre se recopilan

## Uso

Utilice esta estadística para determinar el promedio aritmético del tiempo en cola de las actividades de coordinador asociadas con una subclase de servicio, una carga de trabajo o una clase de trabajo que se ha completado o abortado.

Esta estadística también se puede utilizar para determinar si la plantilla de histograma utilizada para el histograma de tiempo en cola de las actividades es adecuada. Calcule el promedio de tiempo en cola de las actividades a partir del histograma de tiempo en cola de las actividades. Compare el promedio calculado con este elemento de supervisor. Si el promedio calculado se desvía del promedio real que indica este elemento de supervisor, considere la posibilidad de modificar la plantilla del histograma de tiempo en cola de las actividades utilizando un conjunto de valores de intervalo que sea más apropiado para los datos en cuestión.

## Elemento de supervisor coord\_act\_rejected\_total - Total de actividades de coordinador rechazadas

Número total de actividades de coordinador en cualquier nivel de anidamiento que se han rechazado desde el último restablecimiento. Este contador se actualiza cuando un umbral predictivo o una acción de trabajo de ejecución preventiva ha impedido que se ejecute una actividad. En el caso de las clases de servicio, el valor se actualiza cuando se lleva a cabo la actividad. En el caso de las cargas de trabajo, el valor se actualiza con cada instancia de carga de trabajo al final de su unidad de trabajo.

Tabla 137. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	COLLECT AGGREGATE ACTIVITY DATA
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 138. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wlstats	-

## Uso

Este elemento se puede utilizar para ayudar a determinar si los umbrales predictivos y las acciones de trabajo que impiden la ejecución son efectivos y si son demasiado restrictivos.

## Elemento de supervisor coord\_partition\_num - Número de partición coordinadora

Partición coordinadora de la unidad de trabajo o la actividad. En un sistema con varias particiones, la partición coordinadora es la partición donde la aplicación se ha conectado a la base de datos.

Tabla 139. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 140. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan
Violaciones de umbral	event_thresholdviolations	Siempre se recopilan

## Uso

Este elemento permite identificar la partición coordinadora para actividades o unidades de trabajo que tengan registros en particiones que no sean la coordinadora.

## Elemento de supervisor `cost_estimate_top` - Coste estimado máximo

Marca de límite superior del coste estimado de las actividades de DML en todos los niveles de anidamiento de una subclase de servicio o una clase de trabajo. En el caso de las subclases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo.

En el caso de las clases de servicio, el coste estimado de las actividades de DML sólo cuenta para la subclase de servicio en la que la actividad entra en el sistema. Cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, el `cost_estimate_top` de la subclase de servicio con la que se vuelve a correlacionar una actividad no se ve afectado por ello.

Tabla 141. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice este elemento para determinar el coste estimado de las actividades de DML más alto alcanzado en un miembro para una clase de servicio, una carga de trabajo o una clase de trabajo en el intervalo de tiempo de recopilación.

## Elemento de supervisor `cpu_limit` - Límite de CPU de asignador de WLM

Límite de CPU de asignador de WLM configurado para la clase de servicio.

Tabla 142. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	ACTIVITY METRICS BASE

## Elemento de supervisor `cpu_share_type` - Tipo de cuota de CPU de asignador de WLM

Tipo de cuotas de CPU de asignador de WLM configuradas para la clase de servicio. Los valores posibles son soft (flexible) y hard (fija).

Tabla 143. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	ACTIVITY METRICS BASE

## Elemento de supervisor `cpu_shares` - Cuotas de CPU de asignador de WLM

Número de cuotas de CPU de asignador de WLM configuradas para la clase de servicio.

Tabla 144. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	ACTIVITY METRICS BASE

## Elemento de supervisor `cpu_utilization` - Utilización de CPU

Tiempo de CPU total consumido por la clase de servicio o la carga de trabajo en una partición lógica concreta dividido por la cantidad de tiempo de CPU disponible en el sistema principal o la LPAR durante un periodo de tiempo determinado.

Tabla 145. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	REQUEST METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	REQUEST METRICS BASE

Tabla 146. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics	REQUEST METRICS BASE

## Uso

Cuando lo devuelve la función `WLM_GET_WORKLOAD_STATS` o la función `WLM_GET_SERVICE_SUBCLASS_STATS`, este elemento de supervisor representa la utilización de CPU desde el último restablecimiento de las estadísticas.

Cuando lo devuelve la función `MON_SAMPLE_SERVICE_CLASS_METRICS` o la función `MON_SAMPLE_WORKLOAD_METRICS`, este elemento de supervisor representa la utilización de CPU desde que se ha ejecutado la función.

## Elemento de supervisor `cpu_velocity` - Velocidad de CPU

Medida de la cantidad de contención para los recursos de CPU, medidos en una escala de 0 a 1, donde los números más bajos indican una mayor contención.

La velocidad de CPU se calcula midiendo la cantidad de tiempo que el trabajo en una clase de servicio tiene acceso a la CPU dividida por el tiempo total dedicado a acceder a la CPU o a esperar a acceder a la CPU. Ofrece una medida de la eficacia de ejecución del trabajo con respecto a la eficacia con la que podría ejecutarse si dicho trabajo nunca hubiera tenido que esperar a la CPU. La fórmula es la siguiente:

$$\text{cpu\_velocity} = \text{tiempo\_cpu\_total} / (\text{tiempo\_cpu\_total} + \text{tiempo\_cola\_ejecución\_asig\_total})$$

El parámetro de configuración de gestor de bases de datos `wlm_dispatcher` debe establecerse en `ON` para que se recopile `cpu_velocity`.

Tabla 147. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
<code>MON_SAMPLE_SERVICE_CLASS_METRICS</code> - Métricas de clase de servicio de ejemplo	REQUEST METRICS BASE
<code>MON_SAMPLE_WORKLOAD_METRICS</code> - Métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE

## Uso

El asignador es eficaz priorizando una clase de servicio o una carga de trabajo cuando esa clase de servicio o carga de trabajo exige más recursos de CPU en un momento determinado que la que se puede proporcionar. En tales casos, el trabajo que se ejecuta en la clase de servicio o en la carga de trabajo tiene que ponerse en cola durante un tiempo determinado para acceder a los recursos de CPU. Cuando esto sucede, el asignador puede ofrecer más recursos de CPU a dicha clase de servicio o carga de trabajo reduciendo la cantidad de recursos de CPU que ofrece a otros. Una velocidad de CPU alta indica que el asignador puede tener poco impacto en la mejora de los tiempos de respuesta o el rendimiento de esta clase de servicio en este nivel actual de demanda de CPU porque ya se ha respondido a ella. Una velocidad de CPU baja indica que el asignador puede tener un efecto considerable en la mejora de los tiempos de respuesta o el rendimiento de esta clase de servicio o carga de trabajo en este nivel actual de demanda de CPU.

Utilice este elemento para determinar si el trabajo que se ejecuta en una clase de servicio o en una carga de trabajo está dedicando una proporción relativamente elevada de su tiempo en la cola para utilizar los recursos de CPU. Si la velocidad de CPU de una clase de servicio es baja y desea aumentarla, puede ajustar el control de asignador de WLM de los recursos de CPU aumentando el número de

cuotas de CPU o aumentando el límite de CPU asignado a la clase de servicio que muestre exhibe una velocidad de CPU baja.

## Elemento de supervisor `db_work_action_set_id` - ID de conjunto de acciones de trabajo de base de datos

Si esta actividad se ha clasificado en una clase de trabajo de ámbito de base de datos, este elemento de supervisor muestra el ID del conjunto de acciones de trabajo asociado al conjunto de clases de trabajo al que pertenece la clase de trabajo. De lo contrario, este elemento de supervisor muestra el valor 0.

Tabla 148. Información de supervisión de funciones de tabla

Función de tabla	Mandato y nivel de recopilación de elementos de supervisor
WLM_GET_ACTIVITY_DETAILS_COMPLETE (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad	ACTIVITY METRICS BASE

Tabla 149. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

### Uso

Este elemento se puede utilizar con el elemento `db_work_class_id` para identificar exclusivamente la clase de trabajo de base de datos de la actividad, si existe una.

## Elemento de supervisor `db_work_class_id` - ID de clase de trabajo de base de datos

Si esta actividad se ha clasificado en una clase de trabajo de ámbito de base de datos, este elemento de supervisor muestra el ID de la clase de trabajo. De lo contrario, este elemento de supervisor muestra el valor 0.

Tabla 150. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_ACTIVITY_DETAILS_COMPLETE - Obtención de todos los detalles sobre la actividad (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE

Tabla 151. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

### Uso

Este elemento se puede utilizar con el elemento `db_work_action_set_id` para identificar de forma exclusiva la clase de trabajo de base de datos de la actividad, si la hay.

## Elemento de supervisor `destination_service_class_id` – de ID de clase de servicio de destino

ID de la subclase de servicio con la que una actividad se ha vuelto a correlacionar cuando se ha generado el registro de violación de umbral al que pertenece este elemento. Este elemento tiene un valor cero para cualquier acción de umbral que no sea REMAP ACTIVITY.

Tabla 152. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-

### Uso

Utilice este elemento para rastrear la vía de acceso de una actividad a través de las clases de servicio con las que se ha vuelto a correlacionar. Este elemento también se puede utilizar para calcular agregados de la cantidad de actividades que se han correlacionado con una subclase de servicio determinada.

## Elemento de supervisor `estimated_cpu_entitlement` - Titularidad de CPU estimada

Porcentaje de CPU total en un sistema principal o una LPAR que una subclase de servicio está configurada para consumir según sus cuotas de CPU, suponiendo que no consume ni más ni menos de lo que está configurada para consumir. La determinación de las clases de servicio que participan en su cálculo se basa en la utilización de CPU real medida durante el periodo de muestreo frente al valor de configuración de gestor de bases de datos WLM\_DISP\_MIN\_UTIL. El impacto del límite de CPU en una clase de servicio, en las clases de servicio con las que compite o en una clase de servicio padre (si tiene) no se tienen en cuenta en el cálculo.

Tabla 153. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	ACTIVITY METRICS BASE

## Elemento de supervisor `histogram_type` - Tipo de histograma

Tipo del histograma, en formato de serie de caracteres.

Existen siete tipos de histograma.

### **CoordActQueueTime**

Histograma del tiempo en cola de las actividades no anidadas (por ejemplo, en una cola de umbral), medido en el miembro coordinador.

### **CoordActExecTime**

Histograma del tiempo de ejecución de las actividades no anidadas en el miembro coordinador. El tiempo de ejecución no incluye el tiempo de inicialización o de colocación en cola. En el caso de los cursores, el tiempo de ejecución sólo incluye el tiempo dedicado a las peticiones de apertura, captación y cierre. Cuando se vuelve a correlacionar una actividad entre

subclases de servicio, el histograma de tiempo de ejecución se actualiza sólo para la subclase de servicio en la que la actividad lleva a cabo la ejecución.

#### **CoordActLifetime**

Histograma del tiempo transcurrido desde que el gestor de bases de datos identifica una actividad no anidada hasta que la actividad finaliza su ejecución, medido en el miembro coordinador. Cuando se vuelven a correlacionar actividades entre subclases de servicio, el histograma de vida útil sólo se actualiza para la subclase de servicio en la que la actividad finaliza su ejecución.

#### **CoordActInterArrivalTime**

Histograma del intervalo de tiempo entre la llegada de actividades de coordinador no anidadas. El promedio de tiempo entre llegadas se calcula para las subclases por medio de las cuales las actividades entran en el sistema. Cuando se vuelven a correlacionar actividades entre subclases de servicio, el histograma de tiempo entre llegadas de la subclase de servicio con la que se vuelve a correlacionar una actividad no se ve afectado por ello.

#### **CoordActEstCost**

Histograma del coste estimado de las actividades de DML no anidadas. El coste estimado de una actividad sólo cuenta para la subclase de servicio en la que la actividad entra en el sistema.

#### **ReqExecTime**

Histograma de tiempos de ejecución de peticiones, que incluye las peticiones en el miembro coordinador y las subpeticiones tanto en miembros coordinadores como no coordinadores (por ejemplo, las peticiones RPC o las peticiones de subagente SMP). Las peticiones incluidas pueden estar asociadas o no a una actividad: en este histograma se incluyen tanto las peticiones PREPARE como OPEN, por ejemplo, pero mientras que las peticiones OPEN se asocian siempre a una actividad de cursor, las peticiones PREPARE no forman parte de una actividad. El histograma de tiempo de ejecución de una subclase de servicio implicada en una nueva correlación cuenta la parte del tiempo de ejecución la petición parcial ha dedicado a la subclase de servicio.

#### **UowLifetime**

Histograma del tiempo transcurrido, medido en milisegundos, desde el momento en que el gestor de bases de datos identifica una unidad de trabajo hasta que la unidad de trabajo finaliza su ejecución (se confirma o se retrotrae).

*Tabla 154. Información de supervisión de sucesos*

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_histogramabin	Siempre se recopilan

## **Uso**

Utilice este elemento para identificar el tipo de histograma. El mismo registro de estadísticas puede poseer varios histogramas, pero únicamente uno de cada tipo.

## Elemento de supervisor last\_wlm\_reset - Hora del último restablecimiento

Este elemento, con formato de indicación de fecha y hora local, muestra la hora a la que se ha creado el último registro de suceso de estadísticas de este tipo.

Tabla 155. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wlstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_qstats	-

### Uso

Utilice los elementos de supervisor **wlm\_last\_reset** y **statistics\_timestamp** para determinar un periodo de tiempo durante el que se han recopilado las estadísticas de un registro de estadísticas de supervisor de sucesos. El intervalo de recopilación empieza a la hora indica por **wlm\_last\_reset** y acaba a la hora indicada por **statistics\_timestamp**.

## Elemento de supervisor num\_remaps - Número de nuevas correlaciones

Recuento del número de veces que esta actividad se ha vuelto a correlacionar. Si **num\_remaps** es superior a cero, el **service\_class\_id** de este registro de actividad es el ID de la última clase de servicio con la que se ha correlacionado la actividad.

Tabla 156. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

### Uso

Utilice esta información para verificar si la actividad se ha vuelto a correlacionar el número esperado de veces.

## Elemento de supervisor num\_threshold\_violations - Número de violaciones de umbral

Número de violaciones de umbral que han tenido lugar en esta base de datos desde que se activó por última vez.

Este elemento de supervisor es un alias del elemento de supervisor “Elemento de supervisor thresh\_violations - Número de violaciones de umbral” en la página 493, que devuelven algunas funciones de tabla de supervisión (MON\_\*).

Tabla 157. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Base de datos	dbase	Básico

Este contador se puede restablecer para la supervisión de instantáneas.

Tabla 158. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Base de datos	event_db	Siempre se recopilan

## Uso

Este elemento puede utilizarse para ayudar a determinar si los umbrales son efectivos o no para esta aplicación concreta o si las violaciones de umbral son excesivas.

## Elemento de supervisor number\_in\_bin - Número en intervalo

Este elemento lleva la cuenta del número de actividades o de peticiones que forman parte del intervalo de histograma.

Tabla 159. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_histogrambin	-

## Uso

Utilice este elemento para representar la altura de un intervalo en el histograma.

## Elemento de supervisor parent\_activity\_id - ID de la actividad padre

ID exclusivo de la actividad padre de la actividad dentro de la unidad de trabajo de la actividad padre. Si no existe ninguna actividad padre, el valor de este elemento de supervisor es 0.

Tabla 160. Información de supervisión de funciones de tabla

Función de tabla	Mandato y nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 161. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento junto con el elemento **parent\_uow\_id** y el elemento **appl\_id** para identificar de forma exclusiva la actividad padre de la actividad que se describe en este registro de actividad.

## Elemento de supervisor parent\_uow\_id - ID de unidad de trabajo padre

Identificador exclusivo de unidad de trabajo dentro de un descriptor de contexto de aplicación. ID de la unidad de trabajo en la que se origina la actividad padre de la actividad. Si no existe ninguna actividad padre, el valor es 0.

Tabla 162. Información de supervisión de funciones de tabla

Función de tabla	Mandato y nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 163. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento junto con el elemento **parent\_activity\_id** y el elemento **appl\_id** para identificar de forma exclusiva la actividad padre de la actividad que se describe en este registro de actividad.

## Elemento de supervisor prep\_time - Tiempo de preparación

Tiempo en milisegundos necesario para preparar una sentencia de SQL (si la actividad es una sentencia de SQL; si no, el valor es 0).

Tabla 164. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE

Tabla 164. Información de supervisión de funciones de tabla (continuación)

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE

Tabla 165. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan
Antememoria de paquetes	-	COLLECT BASE DATA

## Uso

El elemento de supervisor prep\_time indica la cantidad de tiempo dedicado a la preparación de la sentencia de SQL, si esta actividad era una sentencia de SQL, cuando la sentencia se ha incorporado por primera vez en la antememoria de paquetes de DB2. Este tiempo de preparación no forma parte de la vida útil de la actividad ni representa el tiempo empleado durante una invocación específica de la sentencia si la sentencia ya se ha almacenado en la antememoria de paquetes antes de dicha invocación.

## Elemento de supervisor queue\_assignments\_total - Asignaciones totales en cola

Número de veces que se ha asignado una conexión o una actividad a este umbral desde el último restablecimiento.

Tabla 166. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 167. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	-

## Uso

Este elemento se puede utilizar para determinar el número de veces que se ha puesto una conexión o una actividad en esta cola concreta durante un periodo de tiempo determinado por el intervalo de recopilación de estadísticas. Esto ayuda a determinar la eficacia de los umbrales de colocación en cola.

## Elemento de supervisor `queue_size_top` - Tamaño de cola máximo

Tamaño de cola más alto que se ha alcanzado desde el último restablecimiento.

Tabla 168. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 169. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	-

### Uso

Utilice este elemento para medir la eficacia de los umbrales de colocación en cola y detectar cuándo es excesiva la colocación en cola.

## Elemento de supervisor `queue_time_total` - Tiempo total en cola

Suma del tiempo transcurrido en la cola para todas las conexiones o actividades colocadas en dicha cola desde la última restauración. Las unidades son milisegundos.

Tabla 170. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 171. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	Siempre se recopilan

Este elemento se utiliza para medir la eficacia de los umbrales de colocación en cola y para detectar cuándo es excesiva la colocación en cola.

### Notas de uso

`queue_time_total` no se restablece al final de un intervalo de recopilación de estadísticas. Si `queue_time_total` se utiliza en varios intervalos, puede ser superior al valor obtenido de multiplicar `wlm_collect_int` i `queue_size_top`.

## Elemento de supervisor request\_exec\_time\_avg - Promedio de tiempo de ejecución de peticiones

Promedio aritmético de los tiempos de ejecución de las peticiones asociadas a esta subclase de servicio desde el último restablecimiento. Si se ha desbordado el promedio del que se hace un seguimiento interno, se devuelve el valor -2. Este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE REQUEST DATA para la subclase de servicio se ha establecido en NONE. Las unidades son milisegundos.

Cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, el promedio request\_exec\_time\_avg cuenta la petición parcial de cada subclase implicada en la nueva correlación.

Tabla 172. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	COLLECT AGGREGATE REQUEST DATA

Tabla 173. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-

### Uso

Utilice esta estadística para conocer rápidamente el promedio de tiempo que se dedica a procesar cada petición en un miembro de esta subclase de servicio.

Este promedio también se puede utilizar para determinar si la plantilla de histograma utilizada para el histograma de tiempo de ejecución de peticiones es adecuada. Calcule el promedio de tiempo de ejecución de peticiones a partir del histograma de tiempo de ejecución de peticiones. Compare el promedio calculado con este elemento de supervisor. Si el promedio calculado se desvía del promedio real que indica este elemento de supervisor, considere la posibilidad de modificar la plantilla del histograma de tiempo de ejecución de peticiones utilizando un conjunto de valores de intervalo que sea más apropiado para los datos en cuestión.

## Elemento de supervisor routine\_id - ID de rutina

Identificador de rutina exclusivo. Este elemento de supervisor devuelve 0 si la actividad no forma parte de una rutina.

Tabla 174. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT	ACTIVITY METRICS BASE

Tabla 174. Información de supervisión de funciones de tabla (continuación)

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 175. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activitystmt	Siempre se recopilan
Unidad de trabajo	uow_package_list	Siempre se recopilan
Antememoria de paquetes	pkgcache_metrics	ACTIVITY METRICS BASE

## Uso

El valor de este elemento coincide con un valor de la columna ROUTINEID de la vista SYSCAT.ROUTINES. Cuando la actividad forma parte de una rutina SQL PL que se declara en otra rutina SQL PL, el valor de este elemento es el ROUTINEID de la rutina externa.

## Elemento de supervisor rows\_fetched - Filas captadas

Número de filas leídas en la tabla.

Este elemento de supervisor es un alias del elemento de supervisor **rows\_read**.

**Nota:** Este elemento de supervisor sólo notifica los valores del miembro para el que se registra esta información. En entornos de base de datos de varios miembros, es posible que estos valores no reflejen los totales correctos de toda la actividad.

Tabla 176. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Sentencia

## Uso

Consulte el elemento de supervisor **rows\_read** para obtener información detallada.

## Elemento de supervisor rows\_modified - Filas modificadas

Número de filas insertadas, actualizadas o suprimidas.

Este elemento de supervisor es un alias del elemento de supervisor **rows\_written**.

Tabla 177. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_FORMAT_XML_METRICS_BY_ROW - Obtención de la salida basada en filas formateada para todas las métricas	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_CONNECTION - Obtención de métricas de conexión	REQUEST METRICS BASE
Función de tabla MON_GET_CONNECTION_DETAILS - Obtención de métricas de conexión detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	REQUEST METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE

Tabla 178. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Actividades	event_activitymetrics	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	REQUEST METRICS BASE
Actividades	event_activity	Sentencia
Antememoria de paquetes	Notificadas en el documento activity_metrics.	ACTIVITY METRICS BASE

## Uso

Consulte el elemento de supervisor **rows\_written** para obtener información detallada.

## Elemento de supervisor rows\_returned - Filas devueltas

Número de filas que se han seleccionado y devuelto a la aplicación. Este elemento tiene el valor 0 para los registros de actividad parciales (por ejemplo, si se ha recopilado una actividad mientras aún se estaba ejecutando o cuando no se ha podido grabar un registro de actividad completo en el supervisor de sucesos a causa de limitaciones de memoria).

Este elemento de supervisor es un alias del elemento de supervisor **fetch\_count**.

Tabla 179. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_FORMAT_XML_METRICS_BY_ROW - Obtención de la salida basada en filas formateada para todas las métricas	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_CONNECTION - Obtención de métricas de conexión	REQUEST METRICS BASE
Función de tabla MON_GET_CONNECTION_DETAILS - Obtención de métricas de conexión detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE

Tabla 179. Información de supervisión de funciones de tabla (continuación)

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	REQUEST METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE _ACTIVITIES - Devolución de una lista de actividades	REQUEST METRICS BASE

Tabla 180. Información de supervisión de sucesos

<b>Tipo de suceso</b>	<b>Agrupación de datos lógicos</b>	<b>Conmutador de supervisor</b>
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Actividades	event_activitymetrics	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	REQUEST METRICS BASE
Actividades	event_activity	Siempre se recopilan
Antememoria de paquetes	Notificadas en el documento activity_metrics.	ACTIVITY METRICS BASE

## Uso

Este elemento se puede utilizar para ayudar a determinar los umbrales de las filas devueltas a la aplicación o se puede utilizar para verificar que dicho umbral esté bien configurado y realice su trabajo.

## Elemento de supervisor rows\_returned\_top - Máximo de filas reales devueltas

Marca de límite superior de las filas reales devueltas de actividades de DML en todos los niveles de anidamiento de una clase de servicio o una clase de trabajo. En el caso de las clases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE.

En el caso de clases de servicio, cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, sólo se actualiza la marca de límite superior rows\_returned\_top de la subclase de servicio donde se realiza una actividad. Las marcas de límite superiores de las subclases de servicio con las que se correlaciona una actividad pero en las que no se lleva a cabo dicha actividad no se ven afectadas por ello.

Tabla 181. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice este elemento para conocer la cantidad más alta de filas reales de actividades de DML devueltas que se ha alcanzado en un miembro para una clase de servicio, una carga de trabajo o una clase de trabajo en el intervalo de tiempo de recopilación.

## Elemento de supervisor sc\_work\_action\_set\_id - ID de conjunto de acciones de trabajo de clase de servicio

Si esta actividad se ha clasificado en una clase de trabajo de ámbito de clase de servicio, este elemento de supervisor muestra el ID del conjunto de acciones de trabajo asociado al conjunto de clases de trabajo al que pertenece la clase de trabajo. De lo contrario, este elemento de supervisor muestra el valor 0.

Tabla 182. Información de supervisión de funciones de tabla

Función de tabla	Mandato y nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_ACTIVITY_DETAILS_COMPLETE - Obtención de todos los detalles sobre la actividad (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE

Tabla 183. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Este elemento se puede utilizar con el elemento **sc\_work\_class\_id** para identificar de forma exclusiva la clase de trabajo de la clase de servicio de la actividad, si la hay.

## Elemento de supervisor **sc\_work\_class\_id** - ID de clase de trabajo de clase de servicio

Si esta actividad se ha clasificado en una clase de trabajo de ámbito de clase de servicio, este elemento de supervisor muestra el ID de la clase de trabajo asignada a esta actividad. De lo contrario, este elemento de supervisor muestra el valor 0.

Tabla 184. Información de supervisión de funciones de tabla

Función de tabla	Mandato y nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_ACTIVITY_DETAILS_COMPLETE - Obtención de todos los detalles sobre la actividad (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE

Tabla 185. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Este elemento se puede utilizar con el elemento **sc\_work\_action\_set\_id** para identificar de forma exclusiva la clase de trabajo de la clase de servicio de la actividad, si la hay.

## Elemento de supervisor **section\_env** - Entorno de sección

Blob que contiene la sección de una sentencia de SQL. Es el contenido real de la sección, es decir, la forma ejecutable del plan de consulta.

Tabla 186. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activitystmt	Siempre se recopilan
Antememoria de paquetes	-	COLLECT DETAILED DATA

## Uso

Utilice este elemento con la los procedimientos de explicación de sección para explicar la sentencia y ver el plan de acceso para la sentencia.

## Elemento de supervisor service\_class\_id - ID de clase de servicio

ID exclusivo de la subclase de servicio. En el caso de una unidad de trabajo, este ID representa el ID de subclase de servicio de la carga de trabajo asociada a la conexión que emite la unidad de trabajo.

Tabla 187. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas detalladas de subclase de servicio	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo	ACTIVITY METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de ejemplo de clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 188. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Bloqueo	-	Siempre se recopilan
Unidad de trabajo	-	Siempre se recopilan
Estadísticas	event_histogrambin	Siempre se recopilan
Estadísticas	event_scstats	Siempre se recopilan

### Uso

El valor de este elemento coincide con un valor de la columna SERVICECLASSID de la vista SYSCAT.SERVICECLASSES. Utilice este elemento para buscar el nombre de la subclase de servicio o enlazar con información sobre una subclase de servicio de distintas fuentes. Por ejemplo, una las estadísticas de clase de servicio con los registros de intervalo de histograma.

El valor de este elemento es 0 cuando se cumplen las condiciones siguientes:

- El elemento se notifica en un grupo de datos lógicos event\_histogrambin.
- Se recopilan datos de histograma para un objeto que no es una clase de servicio.

## Elemento de supervisor `service_subclass_name` - Nombre de subclase de servicio

Nombre de una subclase de servicio.

Tabla 189. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de ejemplo de clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE

Tabla 190. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Bloqueo	-	Siempre se recopilan
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan
Estadísticas	event_scstats	Siempre se recopilan

Tabla 190. Información de supervisión de sucesos (continuación)

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de actividad para analizar el comportamiento de una actividad o con otros elementos de estadísticas para analizar una clase de servicio o una cola de umbral.

## Elemento de supervisor `service_superclass_name` - Nombre de superclase de servicio

Nombre de una superclase de servicio.

Tabla 191. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de ejemplo de clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	ACTIVITY METRICS BASE

Tabla 191. Información de supervisión de funciones de tabla (continuación)

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_SUPERCLASS_STATS - Devolución de estadísticas de superclases de servicio	ACTIVITY METRICS BASE

Tabla 192. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan
Estadísticas	event_scstats	Siempre se recopilan
Estadísticas	event_qstats	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de actividad para analizar el comportamiento de una actividad o con otros elementos de estadísticas para analizar una clase de servicio o una cola de umbral.

## Elemento de supervisor `source_service_class_id` - ID de clase de servicio

ID de la subclase de servicio desde la que una actividad se ha vuelto a correlacionar cuando se ha generado el registro de violación de umbral al que pertenece este elemento. Este elemento tiene un valor cero cuando la acción de umbral es una acción distinta de REMAP ACTIVITY.

Tabla 193. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-

## Uso

Utilice este elemento para rastrear la vía de acceso de una actividad a través de las clases de servicio con las que se ha vuelto a correlacionar. También se puede utilizar para calcular agregados de la cantidad de actividades que se han correlacionado desde subclase de servicio determinada.

## Elemento de supervisor `statistics_timestamp` - Indicación de la fecha y la hora de las estadísticas

Hora a la que se ha generado este registro de estadísticas.

Tabla 194. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wlstats	-

Tabla 194. Información de supervisión de sucesos (continuación)

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wcstats	-
Estadísticas	event_qstats	-
Estadísticas	event_histogrambin	-

## Uso

Utilice este elemento para determinar cuándo se ha generado este registro de estadísticas.

Utilice este elemento, junto con el elemento **last\_wlm\_reset**, para identificar el intervalo de tiempo sobre el que se han generado las estadísticas de este registro de estadísticas.

Este elemento de supervisor también se puede utilizar para agrupar todos los registros de estadísticas que se hayan generado para el mismo intervalo de recopilación.

## Elemento de supervisor stmt\_invocation\_id - Identificador de invocación de sentencia

Identificador que distingue una invocación de una rutina de otras en el mismo nivel de anidamiento dentro de una unidad de trabajo. Es exclusivo dentro de una unidad de trabajo para un nivel de anidamiento específico.

Tabla 195. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

Tabla 196. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activitystmt	-
Bloqueo	-	-
Puntos muertos con valores de historial de detalles <sup>1</sup>	event_stmt_history	-
Puntos muertos con historial de detalles <sup>1</sup>	event_stmt_history	-
Unidad de trabajo	Notificada en la lista de paquetes.	-

- 1 Esta opción está en desuso. No se recomienda su uso y podría eliminarse en un próximo release. Utilice la sentencia CREATE EVENT MONITOR FOR LOCKING para supervisar los sucesos relacionados con el bloqueo, como los tiempos de espera excedidos de bloqueo, las esperas de bloqueo y los puntos muertos.

## Uso

Puede utilizar este elemento para identificar de forma exclusiva la invocación en la que una sentencia de SQL concreta se ha ejecutado. También puede utilizar este elemento junto con otras entradas de historial de sentencias para ver la secuencia de sentencias de SQL que han ocasionado el punto muerto.

## Elemento de supervisor temp\_tablespace\_top - Espacio de tablas temporal máximo

Marca de límite superior en KB para el uso de espacios de tablas temporales de actividades de DML en todos los niveles de anidamiento en una clase de servicio o en una clase de trabajo. En el caso de las clases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE. En el caso de las clases de trabajo, este elemento de supervisor devuelve -1 si no se ha especificado ninguna acción de trabajo COLLECT AGGREGATE ACTIVITY DATA para la clase de trabajo. En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE.

En el caso de clases de servicio, cuando se vuelven a correlacionar actividades entre subclases de servicio con una acción REMAP ACTIVITY, sólo cambia la marca de límite superior temp\_tablespace\_top de la subclase de servicio donde se realiza una actividad. Las marcas de límite superiores de las subclases de servicio con las que se correlaciona una actividad pero en las que no se lleva a cabo dicha actividad no se ven afectadas por ello.

Tabla 197. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats	-
Estadísticas	event_wcstats	-
Estadísticas	event_wlstats	-

## Uso

Utilice este elemento para determinar el uso de espacio de tablas temporal del sistema de actividades de DML más alto alcanzado en un miembro para una clase de servicio, una carga de trabajo o una clase de trabajo en el intervalo de tiempo de recopilación.

Sólo actualizan este elemento las actividades que tienen aplicado un umbral de espacio de tablas temporal. Si no se aplica ningún umbral de espacio de tablas temporal a una actividad, se devuelve un valor 0.

## Elemento de supervisor thresh\_violations - Número de violaciones de umbral

Número de veces que se ha violado un umbral.

Este elemento de supervisor es un alias del elemento de supervisor "Elemento de supervisor num\_threshold\_violations - Número de violaciones de umbral" en la página 476, que devuelven las rutinas de supervisión de instantáneas y el supervisor de sucesos de base de datos.

Tabla 198. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_FORMAT_XML_METRICS_BY_ROW - Obtención de la salida basada en filas formateada para todas las métricas	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_CONNECTION - Obtención de métricas de conexión	REQUEST METRICS BASE
Función de tabla MON_GET_CONNECTION_DETAILS - Obtención de métricas de conexión detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	REQUEST METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE

Tabla 199. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Actividades	event_activitymetrics	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Antememoria de paquetes	Notificadas en el documento activity_metrics.	ACTIVITY METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	REQUEST METRICS BASE

## Uso

Utilice este elemento para determinar con rapidez si se ha violado algún umbral de WLM. Si se ha violado algún umbral, puede utilizar el supervisor de sucesos de violaciones de umbral (si se ha creado y está activo) para obtener detalles sobre las violaciones de umbral.

Por ejemplo, para obtener información detallada sobre el umbral que se ha violado.

## Elemento de supervisor threshold\_action - Acción de umbral

Acción del umbral al que se aplica este registro de violación de umbral. Los valores posibles son: Detener (Stop), continuar (Continue) y volver a correlacionar (Remap).

Tabla 200. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-

## Uso

Utilice este elemento para determinar si la actividad que ha violado el umbral se ha detenido cuando se ha producido la violación, ha podido seguir ejecutándose o se ha vuelto a correlacionar con otra subclase de servicio. Si la actividad se ha detenido, la aplicación que ha enviado la actividad habrá recibido un error SQL4712N. Si la actividad se ha correlacionado con otra subclase de servicio, los agentes que trabajen para la actividad en el miembro se desplazarán a la subclase de servicio de destino del umbral.

## Elemento de supervisor threshold\_domain - Dominio de umbral

Dominio del umbral responsable de esta cola.

Los valores posibles son:

- Base de datos
- Conjunto de acciones de trabajo
- Superclase de servicio

- Subclase de servicio
- Carga de trabajo

Tabla 201. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 202. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	-

## Uso

Este elemento se puede utilizar para distinguir las estadísticas de cola de los umbrales que tienen el mismo predicado, pero distintos dominios.

## Elemento de supervisor **threshold\_maxvalue** - Valor máximo de umbral

En el caso de umbrales que no sean de colocación en cola, este elemento de supervisor representa el valor que se ha superado y ha ocasionado esta violación de umbral. En el caso de umbrales de colocación en cola, este elemento de supervisor representa el nivel de simultaneidad que ha generado la colocación en cola. El nivel de simultaneidad que ha ocasionado la violación del umbral de colocación en cola es la suma de los elementos de supervisor **threshold\_maxvalue** y **threshold\_queuesize**.

Tabla 203. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	Siempre se recopilan

## Uso

En el caso de umbrales de actividad, este elemento proporciona un registro histórico del valor máximo del umbral en el momento en el que se violó. Esto es útil cuando el valor máximo del umbral ha cambiado desde que se produjo la violación y el valor anterior ya no está disponible en la vista SYSCAT.THRESHOLDS. En el caso de los umbrales DATATAGINSC IN y DATATAGINSC NOT IN, este elemento contiene el valor del código de datos que ha violado el umbral.

## Elemento de supervisor threshold\_name - Nombre del umbral

Nombre exclusivo del umbral responsable de esta cola.

Tabla 204. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 205. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	-

### Uso

Utilice este elemento para identificar de forma exclusiva el umbral de colocación en cola cuyas estadísticas representa este registro.

## Elemento de supervisor threshold\_predicate - Predicado de umbral

Identifica el tipo de umbral que se ha violado o sobre el que se han recopilado estadísticas.

Tabla 206. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 207. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	Siempre se recopilan
Estadísticas	event_qstats	Siempre se recopilan

### Uso

Utilice este elemento de supervisor junto con otras estadísticas o elementos de supervisor de violación de umbral para analizar una violación de umbral.

Los valores válidos de este elemento de supervisor cuando se notifica en el grupo lógico event\_thresholdviolations son:

- AggSQLTempSpace
- SQLTempSpace
- SQLRowsReturned
- ActivityTotalTime
- EstimatedSQLCost

TotalMemberConnections  
 ConnectionIdleTime  
 ConcurrentWorkloadOccurrences  
 ConcurrentWorkloadActivities  
 ConcurrentDBCoordActivities  
 TotalSCMemberConnections  
 SQLRowsRead  
 SQLRowsReadInSC  
 CPUTime  
 CPUTimeInSC  
 UowTotalTime  
 DataTagInSC  
 DataTagNotInSC

Los valores válidos de este elemento de supervisor cuando se notifica en el grupo lógico event\_qstats son:

TotalMemberConnections  
 ConcurrentDBCoordActivities  
 TotalSCMemberConnections

## Elemento de supervisor threshold\_queuesize - Tamaño de cola de umbral

Tamaño de la cola de un umbral de colocación en cola. Si se intenta sobrepasar este tamaño, se produce una violación de umbral. En el caso de un umbral que no sea de colocación en cola, este valor es 0.

Tabla 208. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-

### Uso

Utilice este elemento para determinar el número de actividades o conexiones en la cola para este umbral en el momento en el que se ha violado el umbral.

## Elemento de supervisor thresholdid - ID de umbral

Identifica el umbral al que se aplica un registro de violación de umbral o para el que se han recopilado estadísticas de cola.

Tabla 209. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE

Tabla 210. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-
Estadísticas	event_qstats	-

## Uso

Utilice este elemento de supervisor junto con otros elementos de supervisor de historial de actividades para analizar una cola de umbral o para analizar la actividad que ha violado un umbral.

## Elemento de supervisor time\_completed - Hora de finalización

Hora a la que la actividad descrita por este registro de actividad ha finalizado su ejecución. Este elemento es una indicación de fecha y hora local.

Tabla 211. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de historial de actividad para analizar el comportamiento de una actividad.

Este campo tiene el valor "0000-00-00-00.00.00.000000" cuando no se ha grabado un registro de actividad completo en un supervisor de sucesos de tabla debido a limitaciones de memoria. Si la actividad se capturó mientras estaba en proceso, este campo representa la hora a la que se recopiló la actividad.

## Elemento de supervisor time\_created - Hora de creación

Hora a la que un usuario ha sometido la actividad que describe este registro de actividad. Este elemento es una indicación de fecha y hora local.

Tabla 212. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	-

## Uso

Utilice este elemento junto con otros elementos de historial de actividad para analizar el comportamiento de una actividad.

## Elemento de supervisor time\_of\_violation - Hora de violación

Hora a la que se ha producido la violación de umbral que describe este registro de violación de umbral. Este elemento es una indicación de fecha y hora local.

Tabla 213. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Violaciones de umbral	event_thresholdviolations	-

## Uso

Utilice este elemento junto con otros elementos de supervisor de violaciones de umbral para analizar una violación de umbral.

## Elemento de supervisor `time_started` - Hora de inicio

Hora a la que la actividad descrita por este registro de actividad ha empezado su ejecución. Este elemento es una indicación de fecha y hora local.

Tabla 214. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de historial de actividad para analizar el comportamiento de una actividad.

Si se rechaza la actividad, el valor del elemento de supervisor `act_exec_time` es 0. En este caso, el valor del elemento de supervisor `time_started` equivale al valor del elemento de supervisor `time_completed`.

## Elemento de supervisor `top` - Principio de intervalo de histograma

Extremo inicial inclusivo del rango de un intervalo de histograma. El valor de este elemento de supervisor también es el extremo inferior exclusivo del rango del siguiente intervalo de histograma.

Tabla 215. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_histogrambin	-

## Uso

Utilice este elemento con el elemento `bottom` correspondiente para determinar el rango de un intervalo dentro de un histograma.

## Elemento de supervisor `total_disp_run_queue_time` - Tiempo total en cola de ejecución de asignador

Tiempo total que las peticiones que se han ejecutado en esta clase de servicio han estado a la espera de acceder a la CPU. Este valor está expresado en microsegundos.

Tabla 216. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_FORMAT_XML_METRICS_BY_ROW - Obtención de la salida basada en filas formateada para todas las métricas	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.

Tabla 216. Información de supervisión de funciones de tabla (continuación)

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_CONNECTION - Obtención de métricas de conexión	REQUEST METRICS BASE
Función de tabla MON_GET_CONNECTION_DETAILS - Obtención de métricas de conexión detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	REQUEST METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	REQUEST METRICS BASE

Tabla 216. Información de supervisión de funciones de tabla (continuación)

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	REQUEST METRICS BASE

Tabla 217. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Antememoria de paquetes	Notificadas en el documento activity_metrics	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics	REQUEST METRICS BASE

## Uso

Junto con **total\_cpu\_time**, el elemento de supervisor **total\_disp\_run\_queue\_time** se puede utilizar para calcular una medida de la cantidad de contención para el recurso de CPU, en una escala del 0 al 1, donde los números más bajos indican una mayor contención para el recurso de CPU. Esta medida, denominada velocidad de CPU, se calcula midiendo la cantidad de tiempo durante el que el trabajo en una clase de servicio tiene acceso a la CPU dividido entre el tiempo total dedicado a acceder a la CPU o a esperar acceder a la CPU. Ofrece una medida de la eficacia de ejecución del trabajo con respecto a la eficacia con la que podría ejecutarse si dicho trabajo nunca hubiera tenido que esperar a la CPU. La fórmula es la siguiente:

$$\text{Velocidad de CPU} = \text{tiempo\_cpu\_total} / (\text{tiempo\_cpu\_total} + \text{tiempo\_cola\_ejecución\_asig\_total})$$

Cuando lo devuelve la función WLM\_GET\_SERVICE\_SUBCLASS\_STATS o la función WLM\_GET\_WORKLOAD\_STATS, este elemento de supervisor representa el tiempo total de espera en cola de ejecución de asignador desde el último restablecimiento de las estadísticas.

Cuando lo devuelve la función MON\_SAMPLE\_SERVICE\_CLASS\_METRICS o la función MON\_SAMPLE\_WORKLOAD\_METRICS, este elemento de supervisor representa el tiempo total de espera en cola de ejecución de asignador desde que se ha ejecutado la función.

## Elemento de supervisor uow\_completed\_total - Total de unidades de trabajo completadas

Número total de unidades de trabajo que se ha completado, ya sea confirmándolas o retrotrayéndolas.

Tabla 218. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	REQUEST METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	REQUEST METRICS BASE

Tabla 219. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento details_xml)	Siempre se recopilan
Estadísticas	event_wlstats (notificadas en el documento details_xml)	Siempre se recopilan

## Uso

Cuando lo devuelve la función WLM\_GET\_SERVICE\_SUBCLASS\_STATS o la función WLM\_GET\_WORKLOAD\_STATS, este elemento de supervisor representa el total de unidades de trabajo completadas desde el último restablecimiento de las estadísticas.

Cuando lo devuelve la función MON\_SAMPLE\_SERVICE\_CLASS\_METRICS o la función MON\_SAMPLE\_WORKLOAD\_METRICS, este elemento de supervisor representa el total de unidades de trabajo completadas desde que se ha ejecutado la función.

## uow\_comp\_status - Estado de finalización de la unidad de trabajo

Estado de la unidad de trabajo y cómo se ha detenido.

### Identificador del elemento

uow\_comp\_status

### Tipo de elemento

información

Tabla 220. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl	Unidad de trabajo
Aplicación DCS	dcs_appl	Básico

Tabla 221. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Transacciones	event_xact	Siempre se recopilan

**Uso** Puede utilizar este elemento para determinar si la unidad de trabajo ha finalizado a causa de un punto muerto o de una terminación anómala. Puede que:

- Se haya confirmado debido a una sentencia de confirmación.
- Se haya retrotraído debido a una sentencia de retrotracción.
- Se haya retrotraído debido a un punto muerto.
- Se haya retrotraído debido a una terminación anómala.
- Se haya confirmado debido a una terminación normal de la aplicación.
- Sea desconocido como resultado de un mandato FLUSH EVENT MONITOR para el que hubiera unidades de trabajo en curso.

**Nota:** Los usuarios de API deben hacer referencia al archivo de cabecera (*sqlmon.h*) que contienen las definiciones de las constantes de supervisor del sistema de bases de datos.

## uow\_elapsed\_time - Tiempo transcurrido de la unidad más reciente de trabajo

Tiempo de ejecución transcurrido de la unidad de trabajo completada más recientemente.

### Identificador del elemento

uow\_elapsed\_time

### Tipo de elemento

hora

Tabla 222. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl	Unidad de trabajo, Indicación de fecha y hora
Aplicación DCS	dcs_appl	Unidad de trabajo, Indicación de fecha y hora

## Uso

Utilice este elemento como indicador del tiempo que tardan las unidades de trabajo en completarse.

Este elemento consta de dos subelementos que indican el tiempo dedicado en segundos y microsegundos (la millonésima parte de un segundo). Los nombres de los subelementos pueden derivarse añadiendo "\_s" y "\_ms" al nombre de este elemento de supervisor. Para recuperar el tiempo total dedicado de este elemento de supervisor, deben sumarse los valores de los dos subelementos. Por ejemplo, si el valor del subelemento "\_s" es 3 y el valor del subelemento "\_ms" es 20, el tiempo total dedicado al elemento de supervisor es 3,00002 segundos.

## Elemento de supervisor uow\_id - ID de unidad de trabajo

Identificador de la unidad de trabajo. El ID de la unidad de trabajo es exclusivo dentro de un descriptor de contexto de aplicación.

Tabla 223. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Devolución de una lista de actividades	ACTIVITY METRICS BASE

Tabla 224. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Bloqueo	-	Siempre se recopilan
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan
Actividades	event_activitystmt	Siempre se recopilan
Actividades	event_activityvals	Siempre se recopilan
Actividades	event_activitymetrics	ACTIVITY METRICS BASE
Violaciones de umbral	event_thresholdviolations	Siempre se recopilan
Historial de cambios	ddlstmtexec txncompletion	Siempre se recopilan

### Uso

Utilice este elemento junto con otros elementos de historial de actividad para analizar el comportamiento de una actividad.

También puede utilizar este elemento con los elementos de supervisor **activity\_id** y **appl\_id** para identificar de forma exclusiva una actividad.

## Elemento de supervisor uow\_lifetime\_avg - Promedio de vida útil de la unidad de trabajo

Promedio de vida útil de una unidad de trabajo. Se mide en milisegundos.

Tabla 225. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	REQUEST METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	REQUEST METRICS BASE

Tabla 226. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento details_xml)	Siempre se recopilan
Estadísticas	event_wlstats (notificadas en el documento details_xml)	Siempre se recopilan

### Uso

Cuando lo devuelve la función WLM\_GET\_SERVICE\_SUBCLASS\_STATS o la función WLM\_GET\_WORKLOAD\_STATS, este elemento de supervisor representa el promedio de vida útil de la unidad de trabajo desde el último restablecimiento de las estadísticas.

Cuando lo devuelve la función MON\_SAMPLE\_SERVICE\_CLASS\_METRICS o la función MON\_SAMPLE\_WORKLOAD\_METRICS, este elemento de supervisor representa el promedio de vida útil de la unidad de trabajo desde que se ha ejecutado la función.

## Elemento de supervisor uow\_lock\_wait\_time - Tiempo total que la unidad de trabajo ha esperado en bloqueos

Tiempo total transcurrido que esta unidad de trabajo has ha estado a la espera de bloqueos. El valor se proporciona en milisegundos.

**Identificador del elemento**  
uow\_lock\_wait\_time

**Tipo de elemento**  
contador

Tabla 227. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl	Unidad de trabajo

**Uso** Este elemento puede ayudar a determinar la gravedad del problema de contención de recursos.

## Elemento de supervisor uow\_log\_space\_used - Espacio utilizado de anotaciones cronológicas de unidad de trabajo

Cantidad de espacio de anotaciones cronológicas (en bytes) utilizado en la unidad de trabajo actual de la aplicación supervisada.

Tabla 228. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE

Tabla 229. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl	Unidad de trabajo

Tabla 230. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Transacciones	event_xact	Siempre se recopilan
Unidad de trabajo	-	Siempre se recopilan

### Uso

Puede utilizar este elemento para conocer los requisitos de anotación cronológica en el nivel de unidad de trabajo.

## Elemento de supervisor uow\_start\_time - Indicación de fecha y hora de inicio de la unidad de trabajo

Fecha y hora a las que la unidad de trabajo ha exigido por primera vez recursos de base de datos.

Tabla 231. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE

Tabla 231. Información de supervisión de funciones de tabla (continuación)

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE

Tabla 232. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl	Unidad de trabajo, Indicación de fecha y hora
Aplicación DCS	dcs_appl	Unidad de trabajo, Indicación de fecha y hora

Tabla 233. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Unidad de trabajo	-	-
Transacción	event_xact	-

## Uso

Esta necesidad de recursos se genera en la primera ejecución de sentencias de SQL de dicha unidad de trabajo:

- En el caso de la primera unidad de trabajo, es la hora de la primera petición de base de datos (ejecución de sentencias de SQL) tras lo indicado por **conn\_complete\_time**.
- En el caso de las unidades de trabajo posteriores, es la hora de la primera petición de base de datos (ejecución de sentencias de SQL) tras la anterior operación COMMIT o ROLLBACK.

**Nota:** En la publicación *Consulta de SQL* se definen los límites de una unidad de trabajo como puntos COMMIT o ROLLBACK.

El supervisor del sistema de bases de datos excluye el tiempo transcurrido entre la operación COMMIT/ROLLBACK y la siguiente sentencia de SQL a partir de su definición de una unidad de trabajo. Este método de medición refleja el tiempo que dedica el gestor de bases de datos a procesar peticiones de base de datos, que es independiente del tiempo dedicado a la lógica de aplicación antes de la primera sentencia de SQL de esa unidad de trabajo. El tiempo transcurrido de la unidad de trabajo incluye el tiempo dedicado a la ejecución de la lógica de aplicación entre las sentencias de SQL dentro de la unidad de trabajo.

Puede utilizar este elemento con el elemento de supervisor **uow\_stop\_time** para calcular el tiempo total transcurrido de la unidad de trabajo y con el elemento de supervisor **prev\_uow\_stop\_time** para calcular el tiempo dedicado en la aplicación entre unidades de trabajo.

Puede utilizar los elementos de supervisor **uow\_stop\_time** y **prev\_uow\_stop\_time** para calcular el tiempo transcurrido para la definición que se indica en la

publicación *Consulta de SQL* de una unidad de trabajo.

## uow\_status - Estado de la unidad de trabajo

Estado de la unidad de trabajo.

### Identificador del elemento

uow\_status

### Tipo de elemento

información

Tabla 234. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Transacciones	event_xact	Siempre se recopilan

**Uso** Puede utilizar este elemento para determinar el estado de una unidad de trabajo. Los usuarios de la API deben consultar el archivo de cabecera `sqlmon.h` que contiene las definiciones de las constantes del supervisor de sistema de bases de datos.

## Elemento de supervisor uow\_stop\_time - Indicación de fecha y hora de detención de unidad de trabajo

Fecha y hora a las que se ha completado la unidad de trabajo más reciente, lo cual tiene lugar cuando se confirman o retrotraen cambios en la base de datos.

Tabla 235. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl	Unidad de trabajo, Indicación de fecha y hora
Aplicación DCS	dcs_appl	Unidad de trabajo, Indicación de fecha y hora

Tabla 236. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Unidad de trabajo	-	-

### Uso

Utilice este elemento con el elemento de supervisor **prev\_uow\_stop\_time** para calcular el total de tiempo transcurrido entre los puntos COMMIT/ROLLBACK, y con el elemento de supervisor **uow\_start\_time** para calcular el tiempo transcurrido de la unidad de trabajo más reciente.

El contenido de la indicación de fecha y hora se establecerá del modo siguiente:

- Cuando la aplicación haya completado una unidad de trabajo y no haya iniciado una nueva (tal como se haya definido en el elemento de supervisor **uow\_start\_time**), este elemento notifica una indicación de fecha y hora válida distinta a cero.
- Cuando la aplicación está ejecutando una unidad de trabajo, este elemento notifica ceros.
- Cuando la aplicación se conecta primero a la base de datos, este elemento se establece en el valor del elemento de supervisor **conn\_complete\_time**.

Cuando se inicia una unidad de trabajo nueva, el contenido de este elemento se desplaza al elemento de supervisor `prev_uow_stop_time`.

## Elemento de supervisor `uow_throughput` - Rendimiento de las unidades de trabajo

Porcentaje de finalización de unidades de trabajo medido en unidades de trabajo por segundo.

Tabla 237. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_SAMPLE_SERVICE_CLASS_METRICS - Obtención de métricas de clase de servicio	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de métricas de carga de trabajo de ejemplo	REQUEST METRICS BASE
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	REQUEST METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	REQUEST METRICS BASE

Tabla 238. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_scstats (notificadas en el documento details_xml)	Siempre se recopilan
Estadísticas	event_wlstats (notificadas en el documento details_xml)	Siempre se recopilan

### Uso

Cuando lo devuelve la función `WLM_GET_SERVICE_SUBCLASS_STATS` o la función `WLM_GET_WORKLOAD_STATS`, este elemento de supervisor representa el rendimiento de las unidades de trabajo desde el último restablecimiento de las estadísticas.

Cuando lo devuelve la función `MON_SAMPLE_SERVICE_CLASS_METRICS` o la función `MON_SAMPLE_WORKLOAD_METRICS`, este elemento de supervisor representa el rendimiento de las unidades de trabajo desde que se ha ejecutado la función.

## Elemento de supervisor `uow_total_time_top` - Tiempo total de UOW máximo

Marca de límite superior para la unidad de tiempo útil de trabajo, en milisegundos.

Tabla 239. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_SERVICE_SUBCLASS_STATS - Devolución de estadísticas de subclases de servicio	REQUEST METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	REQUEST METRICS BASE

Tabla 240. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wlstats	Siempre se recopilan
Estadísticas	event_scstats	Siempre se recopilan

## Uso

Este elemento se puede utilizar para ayudar a determinar si el umbral UOWTOTALTIME es eficaz y puede ayudar a determinar también cómo configurar dicho umbral.

En el caso de las clases de servicio, este elemento de supervisor devuelve -1 cuando COLLECT AGGREGATE ACTIVITY DATA para la subclase de servicio se establece en NONE.

En el caso de cargas de trabajo, este elemento de supervisor devuelve -1 si COLLECT AGGREGATE ACTIVITY DATA para la carga de trabajo se establece en NONE.

En el caso de una clase de servicio, las mediciones realizadas para esta marca de límite superior se calculan para la clase de servicio asignada por la carga de trabajo. Ninguna correlación realizada por un conjunto de acciones de trabajo para cambiar la clase de servicio de una actividad afecta a esta marca de límite superior.

## Elemento de supervisor wl\_work\_action\_set\_id - Identificador de conjunto de acciones de trabajo de carga de trabajo

Si esta actividad se ha clasificado en una clase de trabajo de ámbito de carga de trabajo, este elemento de supervisor muestra el ID del conjunto de acciones de trabajo asociado al conjunto de clases de trabajo al que pertenece la clase de trabajo. De lo contrario, este elemento de supervisor muestra el valor 0.

Tabla 241. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

Tabla 242. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento de supervisor, junto con el elemento de supervisor **wl\_work\_class\_id**, para identificar de forma exclusiva la clase de trabajo de carga de trabajo de la actividad, si la hay.

## Elemento de supervisor **wl\_work\_class\_id** - Identificador de clase de trabajo de carga de trabajo

Si esta actividad se ha clasificado en una clase de trabajo de ámbito de carga de trabajo, este elemento de supervisor muestra el identificador de la clase de trabajo. De lo contrario, este elemento de supervisor muestra el valor 0.

Tabla 243. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

Tabla 244. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento de supervisor, junto con el elemento de supervisor **wl\_work\_action\_set\_id**, para identificar de forma exclusiva la clase de trabajo de carga de trabajo de la actividad, si la hay.

## Elemento de supervisor **wlm\_queue\_assignments\_total** - Asignaciones totales en cola de gestor de carga de trabajo

Número de veces que un umbral de WLM ha puesto en cola actividades.

Tabla 245. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_FORMAT_XML_METRICS_BY_ROW - Obtención de la salida basada en filas formateada para todas las métricas	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE

Tabla 245. Información de supervisión de funciones de tabla (continuación)

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_CONNECTION - Obtención de métricas de conexión	REQUEST METRICS BASE
Función de tabla MON_GET_CONNECTION_DETAILS - Obtención de métricas de conexión detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	REQUEST METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE

Tabla 246. Información de supervisión de sucesos

<b>Tipo de suceso</b>	<b>Agrupación de datos lógicos</b>	<b>Conmutador de supervisor</b>
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Actividades	event_activitymetrics	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE

Tabla 246. Información de supervisión de sucesos (continuación)

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Antememoria de paquetes	Notificadas en el documento activity_metrics.	ACTIVITY METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	REQUEST METRICS BASE

## Elemento de supervisor wlm\_queue\_time\_total - Tiempo total del gestor de carga de trabajo en cola

Tiempo en espera de un umbral de colocación en cola de WLM. Este valor se expresa en milisegundos.

Tabla 247. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
MON_FORMAT_XML_METRICS_BY_ROW - Obtención de la salida basada en filas formateada para todas las métricas	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
MON_FORMAT_XML_TIMES_BY_ROW - Obtención de los tiempos de espera y de proceso de la jerarquía combinada basada en filas formateada	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
MON_FORMAT_XML_WAIT_TIMES_BY_ROW - Obtención de la salida basada en filas formateada para los tiempos de espera	no aplicables; notifica los elementos contenidos en el documento XML proporcionado como entrada a la función de formato.
Función de tabla MON_GET_ACTIVITY_DETAILS - Obtención de todos los detalles sobre la actividad (notificados en el documento XML DETAILS)	ACTIVITY METRICS BASE
Función de tabla MON_GET_CONNECTION - Obtención de métricas de conexión	REQUEST METRICS BASE
Función de tabla MON_GET_CONNECTION_DETAILS - Obtención de métricas de conexión detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT - Obtención de métricas de actividades de sentencia de SQL en la antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_PKG_CACHE_STMT_DETAILS - Obtención de métricas detalladas para entradas de antememoria de paquetes	ACTIVITY METRICS BASE
Función de tabla MON_GET_SERVICE_SUBCLASS - Obtención de métricas de subclase de servicio	REQUEST METRICS BASE

Tabla 247. Información de supervisión de funciones de tabla (continuación)

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla MON_GET_SERVICE_SUBCLASS_DETAILS - Obtención de métricas de subclase de servicio detalladas (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	REQUEST METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo (notificadas en el documento DETAILS XML)	REQUEST METRICS BASE

Tabla 248. Información de supervisión de sucesos

<b>Tipo de suceso</b>	<b>Agrupación de datos lógicos</b>	<b>Conmutador de supervisor</b>
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Actividades	event_activitymetrics	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Antememoria de paquetes	Notificadas en el documento activity_metrics.	ACTIVITY METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	REQUEST METRICS BASE

## Elemento de supervisor wlo\_completed\_total - Instancias de carga de trabajo completadas totales

Número de instancias de carga de trabajo que se deben completar desde el último restablecimiento.

Tabla 249. Información de supervisión de funciones de tabla

<b>Función de tabla</b>	<b>Nivel de recopilación de elementos de supervisor</b>
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 250. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wlstats	-

## Uso

Utilice este elemento para determinar la cantidad de instancias de una carga de trabajo determinada que aportan trabajo al sistema.

## Elemento de supervisor work\_action\_set\_id - ID de conjunto de acciones de trabajo

ID del conjunto de acciones de trabajo al que se aplica este registro de estadísticas.

Tabla 251. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_histogrambin	Siempre se recopilan
Estadísticas	event_wcstats	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de historial de actividad para analizar el comportamiento de una actividad o con otros elementos de estadísticas para analizar una clase de trabajo.

El valor de este elemento es 0 cuando se cumplen las condiciones siguientes:

- El elemento se notifica en un grupo de datos lógicos event\_histogrambin.
- Se recopilan datos de histograma para un objeto que no es una clase de trabajo.

## Elemento de supervisor work\_action\_set\_name - Nombre de conjunto de acciones de trabajo

Nombre del conjunto de acciones de trabajo con el que están asociadas las estadísticas mostradas como parte de este suceso.

Tabla 252. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE
WLM_GET_WORK_ACTION_SET_STATS - Devolución de estadísticas de conjunto de acciones de trabajo	ACTIVITY METRICS BASE

Tabla 253. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	-
Estadísticas	event_wcstats	-

## Uso

Utilice este elemento junto con el elemento **work\_class\_name** para identificar de manera exclusiva la clase de trabajo cuyas estadísticas se muestran en este registro o para identificar de manera exclusiva la clase de trabajo que es el dominio de la cola de umbral cuyas estadísticas se muestran en este registro.

## Elemento de supervisor **work\_class\_id** - ID de clase de trabajo

Identificador de la clase de trabajo a la que se aplica este registro de estadísticas.

Tabla 254. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wcstats	Siempre se recopilan
Estadísticas	event_histogrambin	Siempre se recopilan

## Uso

Utilice este elemento junto con otros elementos de estadísticas para analizar una clase de trabajo.

El valor de este elemento es 0 cuando se cumplen las condiciones siguientes:

- El elemento se notifica en un grupo de datos lógicos event\_histogrambin.
- Se recopilan datos de histograma para un objeto que no es una clase de trabajo.

## Elemento de supervisor **work\_class\_name** - Nombre de clase de trabajo

Nombre de la clase de trabajo con la que están asociadas las estadísticas mostradas como parte de este suceso.

Tabla 255. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE
WLM_GET_WORK_ACTION_SET_STATS - Devolución de estadísticas de conjunto de acciones de trabajo	ACTIVITY METRICS BASE

Tabla 256. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_qstats	-
Estadísticas	event_wcstats	-

## Uso

Utilice este elemento junto con el elemento **work\_action\_set\_name** para identificar de manera exclusiva la clase de trabajo cuyas estadísticas se muestran en este registro o para identificar de manera exclusiva la clase de trabajo que es el dominio de la cola de umbral cuyas estadísticas se muestran en este registro.

## Elemento de supervisor workload\_id - ID de la carga de trabajo

Entero que identifica de forma exclusiva una carga de trabajo.

Tabla 257. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo	ACTIVITY METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de ejemplo	ACTIVITY METRICS BASE

Tabla 258. Información de supervisión de instantáneas

Nivel de instantánea	Agrupación de datos lógicos	Conmutador de supervisor
Aplicación	appl_info	Básico

Tabla 259. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Bloqueo	-	Siempre se recopilan
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	Siempre se recopilan
Estadísticas	event_wlstats	Siempre se recopilan
Estadísticas	event_histogrambin	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan

### Uso

Utilice este ID para identificar de forma exclusiva la carga de trabajo a la que pertenece este registro de estadísticas de actividad, aplicación, intervalo de histograma o carga de trabajo.

El valor de este elemento es 0 cuando se cumplen las condiciones siguientes:

- El elemento se notifica en un grupo de datos lógicos event\_histogrambin.
- Se recopilan datos de histograma para un objeto que no es una carga de trabajo.

## Elemento de supervisor workload\_name - Nombre de carga de trabajo

Nombre de la carga de trabajo.

Tabla 260. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_FORMAT_LOCK_NAME - Formato del nombre de bloqueo interno y devolución de detalles	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_WORKLOAD - Obtención de métricas de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_WORKLOAD_DETAILS - Obtención de métricas detalladas de carga de trabajo	ACTIVITY METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Obtención de ejemplo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_QUEUE_STATS - Devolución de estadísticas de cola de umbral	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_WORKLOAD_STATS - Devolución de las estadísticas de carga de trabajo	ACTIVITY METRICS BASE

Tabla 261. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Bloqueo	-	Siempre se recopilan
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity (notificadas en el documento details_xml)	ACTIVITY METRICS BASE
Estadísticas	event_scstats (notificadas en el documento details_xml)	REQUEST METRICS BASE

Tabla 261. Información de supervisión de sucesos (continuación)

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Estadísticas	event_wlstats (notificadas en el documento details_xml)	REQUEST METRICS BASE
Unidad de trabajo	Notificadas en el documento system_metrics.	Siempre se recopilan
Estadísticas	event_wlstats	Siempre se recopilan

## Uso

En las funciones de tabla de supervisor de sucesos estadísticos y de carga de trabajo, el nombre de la carga de trabajo identifica la carga de trabajo para la que se recopilan y se notifican estadísticas o métricas. En las funciones de tabla de supervisor de sucesos de trabajo y de unidad de trabajo, el nombre de la carga de trabajo identifica la carga de trabajo a la que está asociada la unidad de trabajo.

Utilice el nombre de la carga de trabajo para identificar conjuntos de unidades de trabajo o conjuntos de información que se aplican a una carga de trabajo concreta de interés.

## Elemento de supervisor workload\_occurrence\_id - Identificador de instancia de carga de trabajo

ID de la instancia de carga de trabajo a la que pertenece esta actividad.

Tabla 262. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_AGENTS - Lista de agentes que se ejecutan en una clase de servicio	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE

Tabla 263. Información de supervisión de sucesos

Tipo de suceso	Agrupación de datos lógicos	Conmutador de supervisor
Unidad de trabajo	-	Siempre se recopilan
Actividades	event_activity	Siempre se recopilan

## Uso

Utilice este elemento para identificar la instancia de carga de trabajo que ha sometido la actividad.

## Elemento de supervisor workload\_occurrence\_state - Estado de la instancia de carga de trabajo

Estado de la instancia de carga de trabajo.

Tabla 264. Información de supervisión de funciones de tabla

Función de tabla	Nivel de recopilación de elementos de supervisor
Función de tabla MON_GET_UNIT_OF_WORK - Obtención de métricas de unidad de trabajo	ACTIVITY METRICS BASE
Función de tabla MON_GET_UNIT_OF_WORK_DETAILS - Obtención de métricas detalladas de unidad de trabajo (notificadas en el documento DETAILS XML)	ACTIVITY METRICS BASE
Función de tabla WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Lista de instancias de carga de trabajo	ACTIVITY METRICS BASE

## Uso

Los valores posibles incluyen:

### DECOUPLED

La instancia de carga de trabajo no tiene asignado un agente coordinador (caso de concentrador).

### DISCONNECTPEND

La instancia de carga de trabajo se está desconectando de la base de datos.

### FORCED

Se ha forzado la instancia de carga de trabajo.

### INTERRUPTED

Se ha interrumpido la instancia de carga de trabajo.

### QUEUED

Un umbral de colocación en cola de gestión de carga de trabajo ha puesto en cola un agente coordinador de instancias de carga de trabajo. En un entorno de base de datos particionada, este estado puede indicar que el agente coordinador ha convertido un RPC en otro miembro a fin de obtener tíquets de umbral y aún no ha recibido una respuesta.

### TRANSIENT

La instancia de carga de trabajo aún no se ha correlacionado con una superclase de servicio.

### UOWEXEC

La instancia de carga de trabajo está procesando una petición.

### UOWWAIT

La instancia de carga de trabajo está a la espera de una petición del cliente.

---

## Mandatos

### SET WORKLOAD

Especifica la carga de trabajo a la que se va asignar la base de datos. Este mandato puede emitirse antes de conectarse con una base de datos o puede utilizarse para reasignar la conexión actual después de haberse establecido la conexión. Si se ha establecido la conexión, la reasignación de la carga de trabajo se realizará al principio de la siguiente unidad de trabajo.

#### Autorización

Ninguna, pero consulte las notas de uso

#### Conexión necesaria

Ninguna

#### Sintaxis del mandato

```
➤—SET WORKLOAD TO—┐—AUTOMATIC—┐—  
                          └—SYSDEFAULTADMWORKLOAD—┘—▶
```

#### Parámetros del mandato

##### AUTOMATIC

Especifica que la base de datos se asignará a una carga de trabajo escogida mediante la evaluación de la carga de trabajo que el servidor efectúa automáticamente.

##### SYSDEFAULTADMWORKLOAD

Especifica que la conexión de base de datos se asignará a SYSDEFAULTADMWORKLOAD, permitiendo que los usuarios con autorización *accessctrl*, *dataaccess*, *wladm*, *secadm* o *dbadm* puedan eludir la evaluación de carga de trabajo normal.

#### Ejemplos

Para asignar la conexión a **SYSDEFAULTADMWORKLOAD**:

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

Para restablecer la asignación de carga de trabajo, de forma que se utilice la carga de trabajo que se ha escogido mediante la evaluación de carga de trabajo realizada por el servidor:

```
SET WORKLOAD TO AUTOMATIC
```

#### Notas de uso

Si el ID de autorización de sesión de la conexión de base de datos no dispone de autorización *accessctrl*, *dataaccess*, *wladm*, *secadm* o *dbadm*, la conexión no se puede asignar a **SYSDEFAULTADMWORKLOAD** y se devolverá un error SQL0552N. Si se emite el mandato **SET WORKLOAD TO SYSDEFAULTADMWORKLOAD** antes de conectarse con una base de datos, se devolverá el error SQL0552N después de haberse establecido la conexión de base de datos, al principio de la primera unidad de trabajo. Si el mandato se emite cuando se ha establecido la conexión de base de datos, se

devolverá el error SQL0552N al principio de la siguiente unidad de trabajo, cuando se supone que tendrá lugar la reasignación de carga de trabajo.

---

## Parámetros de configuración

### **wlm\_collect\_int - Intervalo de recopilación de gestión de carga de trabajo**

Este parámetro especifica un intervalo de recopilación y restablecimiento en minutos, para estadísticas de gestión de carga de trabajo (WLM).

Cada  $x$  minutos, (donde  $x$  es el valor del parámetro **wlm\_collect\_int**) se recopilan todas las estadísticas de gestión de carga de trabajo y se envían a cualquier supervisor de sucesos estadísticos activo; a continuación, las estadísticas se restablecen. Si existe un supervisor de sucesos estadísticos activo, en función de cómo se haya creado, las estadísticas se grabarán en un archivo, una conexión o una tabla. Si no existe, las estadísticas sólo se restablecen y no se recopilan.

Las recopilaciones tienen lugar en los intervalos de tiempo especificados partiendo como referencia desde un Domingo a las 00:00:00. Cuando el miembro del catálogo se activa, la siguiente recopilación tiene lugar al principio del siguiente intervalo programado con respecto a esa hora fija. El intervalo programado no es relativo a la hora de activación del miembro del catálogo. Si un miembro no está activo en el momento de la recopilación, no se recopila ninguna estadística para ese miembro. Por ejemplo, si el valor del intervalo se ha establecido en 60 y el miembro del catálogo se ha activado el domingo a las 9:24 de la mañana, la ejecución de las recopilaciones quedará programada para cada hora a partir de la siguiente hora en punto. Por lo tanto, la siguiente recopilación se producirá a las 10:00 de la mañana. Si el miembro no está activo a las 10:00 de la mañana, no se recopilará ninguna estadística para ese miembro.

El proceso de recopilación y restablecimiento se inicia desde el miembro del catálogo. El parámetro **wlm\_collect\_int** debe especificarse en el miembro del catálogo. No se utiliza en otros miembros.

#### **Tipo de configuración**

Base de datos

#### **Tipo de parámetro**

- Configurable en línea

#### **Valor por omisión [rango]**

0 [0 (no se realiza ninguna recopilación), 5 - 32 767]

Las estadísticas de gestión de carga de trabajo recopiladas por un supervisor de sucesos estadísticos se pueden utilizar para supervisar el comportamiento del sistema a largo y a corto plazo. Se puede utilizar un intervalo breve para obtener el comportamiento del sistema tanto a largo como a corto plazo porque los resultados se pueden fusionar para obtener el comportamiento a largo plazo. Sin embargo, si se tienen que fusionar manualmente los resultados de los distintos intervalos, el análisis se complica. Si no es necesario, un intervalo breve aumenta de forma innecesaria el tiempo de proceso. Reduzca, pues, el intervalo para capturar el comportamiento a un plazo más corto y aumente el intervalo para reducir el tiempo de proceso, cuando sólo sea suficiente el comportamiento a largo plazo.

El intervalo tiene que personalizarse según la base de datos, no para cada petición SQL, invocación de mandato o aplicación. No se debe tener en cuenta ningún otro parámetro de configuración.

**Nota:** Todas las funciones de tabla de estadísticas de WLM devuelven estadísticas que se han acumulado desde la última vez que se restablecieron las estadísticas. Las estadísticas se restablecerán con regularidad en el intervalo especificado por este parámetro de configuración.

## **wlm\_dispatcher - Asignador de gestión de carga de trabajo**

Este parámetro habilita (YES) o inhabilita (NO), el asignador de gestión de carga de trabajo de DB2. Por omisión, un asignador de WLM habilitado permite establecer los límites de CPU.

### **Tipo de configuración**

Gestor de bases de datos

### **Se aplica a**

- Servidor de bases de datos con clientes locales y remotos
- Servidor de bases de datos con clientes locales
- Servidor de bases de datos particionadas con clientes locales y remotos

### **Tipo de parámetro**

Configurable en línea

### **Clase de propagación**

Inmediata

### **Valor por omisión [rango]**

NO [NO; YES]

Al actualizar el gestor de bases de datos de DB2, el valor del parámetro de configuración de gestor de bases de datos **wlm\_dispatcher** se establece en NO.

El asignador de WLM proporciona características de planificación de CPU en el nivel de clase de servicio en el gestor de bases de datos DB2 mediante la asignación basada en cuotas de asignación de recursos de CPU, o límites de CPU, o ambos.

Con el asignador de WLM habilitado, todo el trabajo que se ejecute en las clases de servicio de mantenimiento queda bajo el control del asignador. Cuando está habilitado, el asignador aplica los valores de límite de CPU como caso por omisión. Para utilizar la asignación de recursos de CPU basada en cuotas, el parámetro de configuración de gestor de bases de datos **wlm\_disp\_cpu\_shares** debe estar habilitado.

Cuando el parámetro de configuración **wlm\_dispatcher** se establece en YES, se aplican las condiciones siguientes:

- Si una clase de servicio tiene una prioridad de agente establecida en un valor distinto del valor por omisión, se graba un mensaje de aviso en las anotaciones cronológicas db2diag y en el registro de notificaciones de administración en el momento de la activación de la base de datos.
- Cualquier intento de crear o modificar una clase de servicio para establecer la prioridad de agente en un valor que no sea el valor por omisión hace que se devuelva un aviso a la aplicación que ha emitido la sentencia para crear o modificar la clase de servicio.

## wlm\_disp\_concur - Simultaneidad de hebras de asignador de gestor de carga de trabajo

Este parámetro especifica cómo el asignador del gestor de carga de trabajo (WLM) de DB2 establece el nivel de simultaneidad de hebras. También puede establecer manualmente el nivel de simultaneidad de hebras en un valor fijo.

### Tipo de configuración

Gestor de bases de datos

### Se aplica a

- Servidor de bases de datos con clientes locales y remotos
- Servidor de bases de datos con clientes locales
- Servidor de bases de datos particionadas con clientes locales y remotos

### Tipo de parámetro

Configurable en línea

### Clase de propagación

Inmediata

### Valor por omisión [rango]

COMPUTED [COMPUTED; *valor\_establecido\_manualmente*]

Al actualizar el gestor de bases de datos DB2, el valor del parámetro de configuración de gestor de bases de datos **wlm\_disp\_concur** es COMPUTED.

### COMPUTED

El gestor de bases de datos DB2 calcula un nivel de simultaneidad de hebras fijo según el valor de 4 veces el número de CPU lógicas disponibles para el gestor de bases de datos DB2.

### *valor\_establecido\_manualmente*

Puede establecer manualmente el nivel de simultaneidad de hebras en un valor fijo (1 - 32767). El valor óptimo depende del hardware específico utilizado y del nivel del sistema operativo; por lo general, de 2 a 4 veces el número de CPU lógicas del sistema principal o la LPAR.

### Unidad de medida

Número de hebras simultáneas

El valor de este parámetro de configuración de gestor de bases de datos controla el número de hebras que el asignador de WLM permite asignar a las colas de ejecución del sistema operativo en paralelo. El valor se establece como múltiple bajo del número de CPU lógicas disponibles para el gestor de bases de datos DB2. Por lo general, puede establecer el valor en 4 veces el número de CPU lógicas disponibles para tener en cuenta las posibles latencias de planificación que se generan cuando las hebras entran y salen del estado activo. Un valor óptimo es aquel lo suficientemente alto para garantizar que haya un número adecuado de hebras para que el gestor de bases de datos DB2 utilice por completo las CPU en el sistema principal o la LPAR, pero no más alto. Este valor óptimo garantiza la máxima eficacia y proporciona al asignador del WLM de DB2 un control máximo sobre la asignación de CPU.

## wlm\_disp\_cpu\_shares - Cuotas de CPU de asignador de gestor de carga de trabajo

Este parámetro habilita (YES) o inhabilita (NO) el control de las cuotas de CPU por parte del asignador del gestor de carga de trabajo de DB2. Por omisión, un asignador de WLM habilitado sólo controla límites de CPU.

### Tipo de configuración

Gestor de bases de datos

### Se aplica a

- Servidor de bases de datos con clientes locales y remotos
- Servidor de bases de datos con clientes locales
- Servidor de bases de datos particionadas con clientes locales y remotos

### Tipo de parámetro

Configurable en línea

### Clase de propagación

Inmediata

### Valor por omisión [rango]

NO [NO; YES]

Al actualizar el gestor de bases de datos DB2, el valor del parámetro de configuración de gestor de bases de datos **wlm\_disp\_cpu\_shares** es NO.

Si el valor del parámetro de configuración de gestor de bases de datos **wlm\_dispatcher** se establece en YES y el valor del parámetro de configuración de gestor de bases de datos **wlm\_disp\_cpu\_shares** se establece en NO, el asignador de WLM sólo puede aplicar límites de CPU a la gestión de clases de servicio.

Si el valor del parámetro de configuración de gestor de bases de datos **wlm\_dispatcher** se establece en YES y el valor del parámetro de configuración de gestor de bases de datos **wlm\_disp\_cpu\_shares** se establece en YES, el asignador de WLM puede aplicar límites de CPU y cuotas de CPU a la gestión de clases de servicio. Por omisión, a todas las clases de servicio se les asignan 1000 cuotas de CPU fijas para garantizar una división equitativa de los recursos de CPU.

Tabla 265. Resumen de los valores del parámetro de configuración de gestor de bases de datos para la gestión de clases de servicio por parte del asignador de WLM de DB2

Gestión de clases de servicio	Valor de wlm_dispatcher	Valor de wlm_disp_cpu_shares
Ninguna	NO	NO
Límites de CPU	YES	NO
Límites de CPU + cuotas de CPU	YES	YES

## wlm\_disp\_min\_util - Utilización de CPU mínima del asignador del gestor de carga de trabajo

Este parámetro especifica la cantidad mínima de utilización de CPU necesaria para que una clase de servicio se incluya en el compartimiento de recursos de CPU gestionado por el WLM de DB2.

### Tipo de configuración

Gestor de bases de datos

**Se aplica a**

- Servidor de bases de datos con clientes locales y remotos
- Servidor de bases de datos con clientes locales
- Servidor de bases de datos de varios miembros con clientes locales y remotos

**Tipo de parámetro**

Configurable en línea

**Clase de propagación**

Inmediata

**Valor por omisión [rango]**

5 [de 0 a 100]

Al actualizar el gestor de bases de datos DB2, el valor del parámetro de configuración de gestor de bases de datos **wlm\_disp\_min\_util** es 5.

**Unidad de medida**Porcen-  
taje

Para ilustrar el uso de este parámetro de configuración de gestor de bases de datos con un ejemplo, supongamos que hay tres clases de servicio (A, B y C) y que cada una tiene 1000 cuotas de recursos de CPU. En este ejemplo, se obtiene el mismo resultado si las cuotas de clase de servicio son cuotas de CPU fijas o cuotas de CPU flexibles. Las clases de servicio A y B tienen respectivamente una utilización de CPU superior o igual al valor de 8% establecido para el parámetro de configuración **wlm\_disp\_min\_util**. La clase de servicio C tiene una utilización de CPU del 3%, que es inferior al valor del 8% establecido para el parámetro de configuración **wlm\_disp\_min\_util**. En los cálculos de cuotas de CPU, se considera que la clase de servicio C no está ejecutando ningún trabajo. Por lo tanto, sólo las clases de servicio A y B comparten de forma equitativa los recursos de CPU y cada una recibe una cuota del 50%. Cuando la clase de servicio C empieza a ejecutar trabajo hasta tal punto que la utilización de la CPU es superior o igual al valor del 8% establecido para el parámetro de configuración **wlm\_disp\_min\_util**, en este punto, ahora se considera que las clases de servicio A, B y C comparten de forma equitativa los recursos de CPU y cada una recibe una cuota del 33,3%.

En entornos de bases de datos con varios miembros, se compara la suma de las utilizaciones de CPU de todos los miembros de un sistema principal o de una LPAR con el parámetro de configuración **wlm\_disp\_min\_util** a fin de determinar si el sistema principal o la LPAR se incluye en el compartimiento de recursos de CPU gestionado por el WLM.

---

**Vistas de catálogo****SYSCAT.HISTOGRAMTEMPLATEBINS**

Cada fila representa un intervalo de plantilla de histograma.

Tabla 266. Vista de catálogo SYSCAT.HISTOGRAMTEMPLATEBINS

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
TEMPLATENAME	VARCHAR (128)	S	Nombre de la plantilla de histograma.
TEMPLATEID	INTEGER		Identificador de la plantilla de histograma.

Tabla 266. Vista de catálogo SYSCAT.HISTOGRAMTEMPLATEBINS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
BINID	INTEGER		Identificador del binario de plantilla de histograma.
BINUPPERVALUE	BIGINT		El valor superior de un único binario de la plantilla de histograma.

## SYSCAT.HISTOGRAMTEMPLATES

Cada fila representa una plantilla de histograma.

Tabla 267. Vista de catálogo SYSCAT.HISTOGRAMTEMPLATES

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
TEMPLATEID	INTEGER		Identificador de la plantilla de histograma.
TEMPLATENAME	VARCHAR (128)		Nombre de la plantilla de histograma.
CREATE_TIME	TIMESTAMP		Hora a la que se ha creado la plantilla de histograma.
ALTER_TIME	TIMESTAMP		Hora a la que se modificó por última vez la plantilla de histograma.
NUMBINS	INTEGER		Número de binarios de la plantilla de histograma, incluido el último binario con valor superior no vinculado.
REMARKS	VARCHAR (254)	S	Comentarios proporcionados por el usuario o valor nulo.

## SYSCAT.HISTOGRAMTEMPLATEUSE

Cada fila representa una relación entre un objeto de gestión de carga de trabajo que puede utilizar plantillas de histograma y una plantilla de histograma.

Tabla 268. Vista de catálogo SYSCAT.HISTOGRAMTEMPLATEUSE

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
TEMPLATENAME	VARCHAR (128)	S	Nombre de la plantilla de histograma.
TEMPLATEID	INTEGER		Identificador de la plantilla de histograma.

Tabla 268. Vista de catálogo SYSCAT.HISTOGRAMTEMPLATEUSE (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
HISTOGRAMTYPE	CHAR (1)		El tipo de información recopilada por histogramas basados en esta plantilla. <ul style="list-style-type: none"> <li>• C = Histograma de coste estimado de la actividad</li> <li>• E = Histograma de tiempo de ejecución de la actividad</li> <li>• I = Histograma de tiempo hasta la llegada de la actividad</li> <li>• L = Histograma de tiempo de vida de la actividad</li> <li>• Q = Histograma de tiempo en cola de la actividad</li> <li>• R = Histograma de tiempo de ejecución de la actividad</li> <li>• U = Unidad de histograma de tiempo de vida del trabajo</li> </ul>
OBJECTTYPE	CHAR (1)		El tipo de objeto WLM. <ul style="list-style-type: none"> <li>• b = Clase de servicio</li> <li>• k = Acción de trabajo</li> <li>• w = Carga de trabajo</li> </ul>
OBJECTID	INTEGER		Identificador del objeto WLM.
SERVICECLASSNAME	VARCHAR (128)	S	Nombre de la clase de servicio.
PARENTSERVICECLASSNAME	VARCHAR (128)	S	Nombre de la clase de servicio padre de la subclase de servicio que emplea la plantilla de histograma.
WORKACTIONNAME	VARCHAR (128)	S	Nombre de la acción de trabajo que emplea la plantilla de histograma.
WORKACTIONSETNAME	VARCHAR (128)	S	Nombre del conjunto de acciones de trabajo en el que se encuentra la acción de trabajo que emplea la plantilla de histograma.
WORKLOADNAME	VARCHAR (128)	S	Nombre de la carga de trabajo que emplea la plantilla de histograma.

## SYSCAT.SERVICECLASSES

Cada fila representa una clase de servicio.

Tabla 269. Vista de catálogo SYSCAT.SERVICECLASSES

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
SERVICECLASSNAME	VARCHAR (128)		Nombre de la clase de servicio.
PARENTSERVICECLASSNAME	VARCHAR (128)	S	Nombre de clase de servicio de la superclase de servicio padre.
SERVICECLASSID	SMALLINT		Identificador de la clase de servicio.
PARENTID	SMALLINT		Identificador de la clase de servicio padre para esta clase de servicio. 0 si esta clase de servicio es una super clase de servicio.

Tabla 269. Vista de catálogo SYSCAT.SERVICECLASSES (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
CREATE_TIME	TIMESTAMP		Hora en que se ha creado la clase de servicio.
ALTER_TIME	TIMESTAMP		Hora en que se modificó por última vez la clase de servicio.
ENABLED	CHAR (1)		Estado de la clase de servicio. <ul style="list-style-type: none"> <li>• N = Inhabilitado</li> <li>• Y = Habilitada</li> </ul>
AGENTPRIORITY	SMALLINT		Prioridad de hebra de los agentes de la clase de servicio relacionados con la prioridad normal de las hebras de DB2. <ul style="list-style-type: none"> <li>• De -20 a 20 (Linux y UNIX)</li> <li>• De -6 a 6 (Windows)</li> <li>• -32768 = no establecido</li> </ul>
PREFETCHPRIORITY	CHAR (1)		Prioridad de los agentes en la clase de servicio. <ul style="list-style-type: none"> <li>• H = Alto</li> <li>• L = Bajo</li> <li>• M = Medio</li> <li>• Blanco = no establecido</li> </ul>
MAXDEGREE	SMALLINT	S	Reservado para una utilización futura.
BUFFERPOOLPRIORITY	CHAR (1)		Prioridad de agrupación de almacenamientos intermedios de los agentes en la clase de servicio. <ul style="list-style-type: none"> <li>• H = Alto</li> <li>• L = Bajo</li> <li>• M = Medio</li> <li>• Blanco = No establecido</li> </ul>
INBOUNDCORRELATOR	VARCHAR (128)	S	Para su utilización en el futuro.
OUTBOUNDCORRELATOR	VARCHAR (128)	S	Serie utilizada para asociar la clase de servicio a una clase de servicio del gestor de carga de trabajo del sistema operativo.
COLLECTAGGACTDATA	CHAR (1)		Especifica los datos de actividad agregados que deben capturarse para la clase de servicio por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• B = Recopilar datos de actividad agregados básicos</li> <li>• E = Recopilar datos de actividad agregados ampliados</li> <li>• N = Ninguno</li> </ul>
COLLECTAGGREQDATA	CHAR (1)		Especifica los datos de petición agregados que deben capturarse para la clase de servicio por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• B = Recopilar datos de petición agregados básicos</li> <li>• N = Ninguno</li> </ul>

Tabla 269. Vista de catálogo SYSCAT.SERVICECLASSES (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
COLLECTACTDATA	CHAR (1)		<p>Especifica los datos de actividad que deben recopilarse por medio del supervisor de sucesos aplicable.</p> <ul style="list-style-type: none"> <li>• D = Datos de actividad con detalles</li> <li>• N = Ninguno</li> <li>• S = Datos de actividad con detalles y entorno de sección</li> <li>• V = Datos de actividad con detalles y valores</li> <li>• W = Datos de actividad sin detalles</li> <li>• X = Datos de actividad con detalles, entorno de sección y valores</li> </ul>
COLLECTACTPARTITION	CHAR (1)		<p>Especifica el lugar en el que se recopilarán los datos de actividad.</p> <ul style="list-style-type: none"> <li>• C = Miembro coordinador de la actividad</li> <li>• D = Todos los miembros</li> </ul>
COLLECTREQMETRICS	CHAR (1)		<p>Especifica el nivel de supervisión de las peticiones enviadas por una conexión que está asociada con la superclase de servicio.</p> <ul style="list-style-type: none"> <li>• B = Recopilar métricas de peticiones base</li> <li>• E = Recopilar métricas de peticiones ampliadas</li> <li>• N = Ninguno</li> </ul>
CPUSHARES	INTEGER		<p>El número de cuotas de CPU asignado a esta clase de servicio.</p>
CPUSHARETYPE	CHAR (1)		<p>Especifica el tipo de cuotas de CPU.</p> <ul style="list-style-type: none"> <li>• S = Cuotas flexibles</li> <li>• H = Cuotas fijas</li> </ul>
CPULIMIT	SMALLINT		<p>Porcentaje máximo de recursos de CPU que se puede asignar a la clase de servicio; -1 si no hay límite de CPU.</p>
SORTMEMORYPRIORITY	CHAR (1)		<p>Reservado para una utilización futura.</p>
SECTIONACTUALSOPTIONS	VARCHAR (32)		<p>Especifica qué datos reales de sección se recopilan durante la ejecución de una sección.</p> <p>La primera posición de la serie representa si la colección de datos reales de sección está habilitada.</p> <ul style="list-style-type: none"> <li>• B = Recuentos y estadísticas de cardinalidad de operador básico habilitados y recopilados para cada objeto al que hace referencia la sección (sólo sentencias DML).</li> <li>• N = No habilitado.</li> </ul> <p>La segunda posición siempre es 'N' y se reserva para utilizarlo en el futuro.</p>

Tabla 269. Vista de catálogo SYSCAT.SERVICECLASSES (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
COLLECTAGGUOWDATA	CHAR (1)		Especifica los datos agregados de unidad de trabajo que deben capturarse para la clase de servicio por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• B = Recopilar datos agregados básicos de unidad de trabajo</li> <li>• N = Ninguno</li> </ul>
REMARKS	VARCHAR (254)	S	Comentarios proporcionados por el usuario o valor nulo.

## SYSCAT.THRESHOLDS

Cada fila representa un umbral.

Tabla 270. Vista de catálogo SYSCAT.THRESHOLDS

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
THRESHOLDNAME	VARCHAR (128)		Nombre del umbral.
THRESHOLDID	INTEGER		Identificador del umbral.
ORIGIN	CHAR (1)		Origen del umbral. <ul style="list-style-type: none"> <li>• U = El usuario creó el umbral.</li> <li>• W = Un conjunto de acciones de trabajo creó el umbral.</li> </ul>
THRESHOLDCLASS	CHAR (1)		Clasificación del umbral. <ul style="list-style-type: none"> <li>• A = Umbral agregado</li> <li>• C = Umbral de actividad</li> </ul>
THRESHOLDPREDICATE	VARCHAR (15)		Tipo de umbral. Los valores posibles son: <ul style="list-style-type: none"> <li>• AGGTEMPSPACE</li> <li>• CONCDBC</li> <li>• CONCWCN</li> <li>• CONCWOC</li> <li>• CONNIDLETIME</li> <li>• CPUTIME</li> <li>• CPUTIMEINSC</li> <li>• DATATAGINSC</li> <li>• DATATAGNOTINSC</li> <li>• DBCONN</li> <li>• ESTSQLCOST</li> <li>• ROWSREAD</li> <li>• ROWSREADINSC</li> <li>• ROWSRET</li> <li>• SCCONN</li> <li>• TEMPSPACE</li> <li>• TOTALTIME</li> <li>• UOWTOTALTIME</li> </ul>

Tabla 270. Vista de catálogo SYSCAT.THRESHOLDS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
THRESHOLDPREDICATEID	SMALLINT		Identificador del predicado de umbral.
DOMAIN	CHAR (2)		<p>Dominio del umbral.</p> <ul style="list-style-type: none"> <li>• DB = Base de datos</li> <li>• SB = Subclase de servicio</li> <li>• SP = Superclase de servicio</li> <li>• WA = Conjunto de acciones de trabajo</li> <li>• WD = Definición de carga de trabajo</li> <li>• SQ = Sentencia de SQL</li> </ul>
DOMAINID	INTEGER		Identificador del objeto con el que está asociado el umbral. Puede ser una clase de servicio, acción de trabajo o ID exclusivo de carga de trabajo. Si es un umbral de base de datos, este valor es 0.
ENFORCEMENT	CHAR (1)		<p>Ámbito de implantación para el umbral.</p> <ul style="list-style-type: none"> <li>• D = Base de datos</li> <li>• P = Miembro</li> <li>• W = Aparición de carga de trabajo</li> </ul>
QUEUING	CHAR (1)		<ul style="list-style-type: none"> <li>• N = El umbral no está en cola</li> <li>• Y = El umbral está en cola</li> </ul>
MAXVALUE	BIGINT		Límite superior especificado por el umbral. Si THRESHOLDPREDICATE es 'DATATAGINSC' o 'DATATAGNOTINSC', este valor codifica uno o más códigos de datos.
DATATAGLIST	VARCHAR (256)	S	Si THRESHOLDPREDICATE es 'DATATAGINSC' o 'DATATAGNOTINSC', este valor representa uno o más códigos de datos como una lista separada por comas. De lo contrario, es un valor nulo.
QUEUESIZE	INTEGER		Si QUEUEING es 'Y', el tamaño de la cola. -1 en caso contrario.
OVERFLOWPERCENT	SMALLINT		Reservado para una utilización futura.
COLLECTACTDATA	CHAR (1)		<p>Especifica los datos de actividad que deben recopilarse por medio del supervisor de sucesos aplicable.</p> <ul style="list-style-type: none"> <li>• D = Datos de actividad con detalles</li> <li>• N = Ninguno</li> <li>• S = Datos de actividad con detalles y entorno de sección</li> <li>• V = Datos de actividad con detalles y valores</li> <li>• W = Datos de actividad sin detalles</li> <li>• X = Datos de actividad con detalles, entorno de sección y valores</li> </ul>

Tabla 270. Vista de catálogo SYSCAT.THRESHOLDS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
COLLECTACTPARTITION	CHAR (1)		Especifica el lugar en el que se recopilarán los datos de actividad. <ul style="list-style-type: none"> <li>• C = Miembro coordinador de la actividad</li> <li>• D = Todos los miembros</li> </ul>
EXECUTION	CHAR (1)		Indica la acción de ejecución que se lleva a cabo después de que se haya superado un umbral. <ul style="list-style-type: none"> <li>• C = La ejecución continúa</li> <li>• F = Se fuerza a que la aplicación salga del sistema</li> <li>• R = La ejecución se vuelve a correlacionar con una subclase de servicio diferente</li> <li>• S = La ejecución se detiene</li> </ul>
REMAPSCID	SMALLINT		ID de la subclase de servicio de destino de la acción REMAP ACTIVITY.
VIOLATIONRECORDLOGGED	CHAR (1)		Indica si se ha anotado un registro en el supervisor de sucesos tras producirse la violación de umbral. <ul style="list-style-type: none"> <li>• N = No</li> <li>• Y = Sí</li> </ul>
CHECKINTERVAL	INTEGER		Intervalo, en segundos, en el que se comprueba la condición de umbral si THRESHOLDPREDICATE es: <ul style="list-style-type: none"> <li>• 'CPUTIME'</li> <li>• 'CPUTIMEINSC'</li> <li>• 'ROWSREAD'</li> <li>• 'ROWSREADINSC'</li> </ul> De lo contrario, -1.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> <li>• N = Se ha inhabilitado este umbral.</li> <li>• Y = Se ha habilitado este umbral.</li> </ul>
CREATE_TIME	TIMESTAMP		Hora en que se ha creado el umbral.
ALTER_TIME	TIMESTAMP		Hora a la que se ha modificado por última vez el umbral.
REMARKS	VARCHAR (254)	S	Comentarios proporcionados por el usuario o valor nulo.

## SYSCAT.WORKACTIONS

Cada fila representa una acción de trabajo definida para un conjunto de acciones de trabajo.

Tabla 271. Vista de catálogo SYSCAT.WORKACTIONS

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
ACTIONNAME	VARCHAR (128)		Nombre de la acción de trabajo.
ACTIONID	INTEGER		Identificador de la acción de trabajo.

Tabla 271. Vista de catálogo SYSCAT.WORKACTIONS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
ACTIONSETNAME	VARCHAR (128)	S	Nombre del conjunto de acciones de trabajo.
ACTIONSETID	INTEGER		Identificador del conjunto de acciones de trabajo al que pertenece esta acción de trabajo. Esta columna hace referencia a la columna ACTIONSETID en la vista SYSCAT.WORKACTIONSETS.
WORKCLASSNAME	VARCHAR (128)	S	Nombre de la clase de trabajo.
WORKCLASSID	INTEGER		Identificador de la clase de trabajo. Esta columna hace referencia a la columna WORKCLASSID en la vista SYSCAT.WORKCLASSES.
CREATE_TIME	TIMESTAMP		Hora a la que se ha creado la acción de trabajo.
ALTER_TIME	TIMESTAMP		Hora a la que se modificó por última vez la acción de trabajo.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> <li>• N = Se ha inhabilitado esta acción de trabajo.</li> <li>• Y = Se ha habilitado esta acción de trabajo.</li> </ul>

Tabla 271. Vista de catálogo SYSCAT.WORKACTIONS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
ACTIONTYPE	CHAR (1)		<p>El tipo de acción efectuada en cada actividad de DB2 que coincide con los atributos de la clase de trabajo que pertenece al ámbito.</p> <ul style="list-style-type: none"> <li>• B = Recopilar datos de actividad agregados básicos; sólo puede especificarse para los conjuntos de acciones de trabajo que se aplican a las clases de servicio o a las cargas de trabajo.</li> <li>• C = Permitir toda actividad de DB2 bajo la clase de trabajo asociada para ejecutar e incrementar el contador de la clase de trabajo.</li> <li>• D = Recopilar datos de actividad con detalles en el miembro coordinador de la actividad.</li> <li>• E = Recopilar datos de actividad agregados ampliados; sólo puede especificarse para los conjuntos de acciones de trabajo que se aplican a las clases de servicio o las cargas de trabajo.</li> <li>• F = Recopilar datos de actividad con detalles, secciones y valores en el miembro coordinador de la actividad.</li> <li>• G = Recopilar detalles de actividad y secciones en el miembro coordinador de la actividad y recopilar datos de actividad en todos los miembros.</li> <li>• H = Recopilar detalles de actividad, secciones y valores en el miembro coordinador de la actividad y recopilar datos de actividad en todos los miembros.</li> <li>• M = Correlacionar con una subclase de servicio; sólo puede especificarse para los conjuntos de acciones de trabajo que se aplican a las clases de servicio.</li> <li>• P = Impedir la ejecución de cualquier actividad de DB2 bajo la clase de trabajo con la que está asociada esta acción de trabajo.</li> <li>• S = Recopilar datos de actividad con detalles y secciones en el miembro coordinador de la actividad.</li> <li>• T = La acción representa un umbral; sólo puede especificarse para los conjuntos de acciones de trabajo que están asociados con una base de datos o una carga de trabajo.</li> <li>• U = Correlacionar todas las actividades con un nivel de anidamiento de 0, y todas las actividades anidadas bajo estas actividades con una subclase de servicio; sólo puede especificarse para los conjuntos de acciones de trabajo que se aplican a las clases de servicio.</li> </ul>

Tabla 271. Vista de catálogo SYSCAT.WORKACTIONS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
			<ul style="list-style-type: none"> <li>• V = Recopilar datos de actividad con detalles y valores en el miembro coordinador.</li> <li>• W = Recopilar datos de actividad sin detalles en el miembro coordinador.</li> <li>• X = Recopilar datos de actividad con detalles en el miembro coordinador y recopilar datos de actividad en todos los miembros.</li> <li>• Y = Recopilar datos de actividad con detalles y valores en el miembro coordinador y recopilar datos de actividad en todos los miembros.</li> <li>• Z = Recopilar datos de actividad sin detalles en todos los miembros.</li> </ul>
REFOBJECTID	INTEGER	S	Si ACTIONTYPE es 'M' (correlación) o 'N' (correlación anidada), este valor se establece en el ID de la subclase de servicio con la que se correlaciona la actividad de DB2. Si ACTIONTYPE es 'T' (umbral), este valor se establece en el ID del umbral a utilizar. Para todas las demás acciones, este valor es NULL.
REFOBJECTTYPE	VARCHAR (30)		Si ACTIONTYPE es 'M' o 'N', este valor se establece en 'SERVICE CLASS'; si ACTIONTYPE es 'T', este valor es 'THRESHOLD'; en caso contrario es un valor nulo.
SECTIONACTUALSOPTIONS	VARCHAR (32)		<p>Especifica qué datos reales de sección se recopilan durante la ejecución de una sección.</p> <p>La primera posición de la serie representa si la colección de datos reales de sección está habilitada.</p> <ul style="list-style-type: none"> <li>• B = Recuentos y estadísticas de cardinalidad de operador básico habilitados y recopilados para cada objeto al que hace referencia la sección (sólo sentencias DML).</li> <li>• N = No habilitado.</li> </ul> <p>La segunda posición siempre es 'N' y se reserva para utilizarlo en el futuro.</p>

## SYSCAT.WORKACTIONSETS

Cada fila representa un conjunto de acciones de trabajo.

Tabla 272. Vista de catálogo SYSCAT.WORKACTIONSETS

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
ACTIONSETNAME	VARCHAR (128)		Nombre del conjunto de acciones de trabajo.
ACTIONSETID	INTEGER		Identificador del conjunto de acciones de trabajo.
WORKCLASSETNAME	VARCHAR (128)	S	Nombre del conjunto de clases de trabajo.

Tabla 272. Vista de catálogo SYSCAT.WORKACTIONSETS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
WORKCLASSETID	INTEGER		El identificador del conjunto de clases de trabajo que ha de correlacionarse con el objeto especificado por medio de OBJECTID. Esta columna hace referencia a WORKCLASSETID en la vista SYSCAT.WORKCLASSETS.
CREATE_TIME	TIMESTAMP		Hora a la que se ha creado el conjunto de acciones de trabajo.
ALTER_TIME	TIMESTAMP		Hora a la que se modificó por última vez el conjunto de acciones de trabajo.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> <li>• N = Se ha inhabilitado este conjunto de acciones de trabajo.</li> <li>• Y = Se ha habilitado este conjunto de acciones de trabajo.</li> </ul>
OBJECTTYPE	CHAR (1)		<ul style="list-style-type: none"> <li>• b = Superclase de servicio</li> <li>• w = Carga de trabajo</li> <li>• Blanco = Base de datos</li> </ul>
OBJECTNAME	VARCHAR (128)	S	Nombre de la clase de servicio o la carga de trabajo.
OBJECTID	INTEGER		El identificador del objeto al que está correlacionado el conjunto de clases de trabajo (especificado mediante WORKCLASSETID). Si OBJECTTYPE es 'b', OBJECTID es el ID de la superclase de servicio. Si OBJECTTYPE es 'w', OBJECTID es el ID de la carga de trabajo. Si OBJECTTYPE es un blanco, OBJECTID es -1.
REMARKS	VARCHAR (254)	S	Comentarios proporcionados por el usuario o valor nulo.

## SYSCAT.WORKCLASSES

Cada fila representa una clase de trabajo definida para un conjunto de clases de trabajo.

Tabla 273. Vista de catálogo SYSCAT.WORKCLASSES

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
WORKCLASSNAME	VARCHAR (128)		Nombre de la clase de trabajo.
WORKCLASSETNAME	VARCHAR (128)	S	Nombre del conjunto de clases de trabajo.
WORKCLASSID	INTEGER		Identificador de la clase de trabajo.
WORKCLASSETID	INTEGER		Identificador del conjunto de clases de trabajo al que pertenece esta clase de trabajo. Esta columna hace referencia a la columna WORKCLASSETID en la vista SYSCAT.WORKCLASSETS.
CREATE_TIME	TIMESTAMP		Hora a la que se ha creado la clase de trabajo.

Tabla 273. Vista de catálogo SYSCAT.WORKCLASSES (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
ALTER_TIME	TIMESTAMP		Hora a la que se modificó por última vez la clase de trabajo.
EVALUATIONORDER	SMALLINT		Identifica exclusivamente el orden de evaluación para seleccionar una clase de trabajo en un conjunto de clases de trabajo.

## SYSCAT.WORKCLASSETS

Cada fila representa un conjunto de clases de trabajo.

Tabla 274. Vista de catálogo SYSCAT.WORKCLASSETS

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
WORKCLASSETNAME	VARCHAR (128)		Nombre del conjunto de clases de trabajo.
WORKCLASSETID	INTEGER		Identificador del conjunto de clases de trabajo.
CREATE_TIME	TIMESTAMP		Hora a la que se ha creado el conjunto de clases de trabajo.
ALTER_TIME	TIMESTAMP		Hora a la que se modificó por última vez el conjunto de clases de trabajo.
REMARKS	VARCHAR (254)	S	Comentarios proporcionados por el usuario o valor nulo.

## SYSCAT.WORKLOADAUTH

Cada fila representa un usuario, grupo o rol al que se ha otorgado el privilegio USAGE sobre una carga de trabajo.

Tabla 275. Vista de catálogo SYSCAT.WORKLOADAUTH

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
WORKLOADID	INTEGER		Identificador para la carga de trabajo.
WORKLOADNAME	VARCHAR (128)		Nombre de la carga de trabajo.
GRANTOR	VARCHAR (128)		El que ha otorgado el privilegio.
GRANTORTYPE	CHAR (1)		<ul style="list-style-type: none"> <li>• U = Se otorga a un usuario individual</li> </ul>
GRANTEE	VARCHAR (128)		El que mantiene el privilegio.
GRANTEETYPE	CHAR (1)		<ul style="list-style-type: none"> <li>• G = Se otorga a un grupo</li> <li>• R = Se otorga a un rol</li> <li>• U = Se otorga a un usuario individual</li> </ul>
USAGEAUTH	CHAR (1)		Indica si el otorgado retiene el privilegio de USAGE en la carga de trabajo. <ul style="list-style-type: none"> <li>• N = No se mantiene</li> <li>• Y = Se mantiene</li> </ul>

## SYSCAT.WORKLOADCONNATTR

Cada fila representa un atributo de conexión en la definición de una carga de trabajo.

Tabla 276. Vista de catálogo SYSCAT.WORKLOADCONNATTR

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
WORKLOADID	INTEGER		Identificador para la carga de trabajo.
WORKLOADNAME	VARCHAR (128)		Nombre de la carga de trabajo.
CONNATTRTYPE	VARCHAR (30)		Tipo del atributo de conexión. <ul style="list-style-type: none"> <li>• 1 = APPLNAME</li> <li>• 2 = SYSTEM_USER</li> <li>• 3 = SESSION_USER</li> <li>• 4 = SESSION_USER GROUP</li> <li>• 5 = SESSION_USER ROLE</li> <li>• 6 = CURRENT CLIENT_USERID</li> <li>• 7 = CURRENT CLIENT_APPLNAME</li> <li>• 8 = CURRENT CLIENT_WRKSTNNAME</li> <li>• 9 = CURRENT CLIENT_ACCTNG</li> <li>• 10 = ADDRESS</li> </ul>
CONNATTRVALUE	VARCHAR (1000)		Valor del atributo de conexión.

## SYSCAT.WORKLOADS

Cada fila representa una carga de trabajo.

Tabla 277. Vista de catálogo SYSCAT.WORKLOADS

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
WORKLOADID	INTEGER		Identificador para la carga de trabajo.
WORKLOADNAME	VARCHAR (128)		Nombre de la carga de trabajo.
EVALUATIONORDER	SMALLINT		Orden de evaluación utilizado para seleccionar una carga de trabajo.
CREATE_TIME	TIMESTAMP		Hora a la que se ha creado la carga de trabajo.
ALTER_TIME	TIMESTAMP		Hora a la que se ha modificado por última vez la carga de trabajo.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> <li>• N = Se ha inhabilitado esta carga de trabajo.</li> <li>• Y = Se ha habilitado esta carga de trabajo.</li> </ul>
ALLOWACCESS	CHAR (1)		<ul style="list-style-type: none"> <li>• N = Una UOW asociada a esta carga de trabajo se rechazará.</li> <li>• Y = Una unidad de trabajo (UOW) asociada a esta carga de trabajo podrá acceder a la base de datos.</li> </ul>

Tabla 277. Vista de catálogo SYSCAT.WORKLOADS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
MAXDEGREE	SMALLINT		Grado máximo de paralelismo para la carga de trabajo. Los valores válidos son: de 1 a 32767 y -1. Si MAXIMUM DEGREE es DEFAULT, el valor es -1.
SERVICECLASSNAME	VARCHAR (128)		Nombre de la subclase de servicio a la que se asigna una unidad de trabajo (asociada a esta carga de trabajo).
PARENTSERVICECLASSNAME	VARCHAR (128)	S	Nombre de la superclase de servicio a la que se asigna una unidad de trabajo (asociada a esta carga de trabajo).
COLLECTAGGACTDATA	CHAR (1)		Especifica los datos de actividad agregados que deben capturarse para la carga de trabajo por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• B = Recopilar datos de actividad agregados básicos</li> <li>• E = Recopilar datos de actividad agregados ampliados</li> <li>• N = Ninguno</li> </ul>
COLLECTACTDATA	CHAR (1)		Especifica los datos de actividad que deben recopilarse por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• D = Datos de actividad con detalles</li> <li>• N = Ninguno</li> <li>• S = Datos de actividad con detalles y entorno de sección</li> <li>• V = Datos de actividad con detalles y valores. Se aplica cuando la columna COLLECT se establece en 'C'</li> <li>• W = Datos de actividad sin detalles</li> <li>• X = Datos de actividad con detalles, entorno de sección y valores</li> </ul>
COLLECTACTPARTITION	CHAR (1)		Especifica el lugar en el que se recopilarán los datos de actividad. <ul style="list-style-type: none"> <li>• C = Miembro coordinador de la actividad</li> <li>• D = Todos los miembros</li> </ul>
COLLECTDEADLOCK	CHAR (1)		Especifica los sucesos de punto muerto que deben recopilarse por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• H = Recopilar datos de punto muerto con actividades anteriores únicamente</li> <li>• N = No recopilar datos de punto muerto</li> <li>• V = Recopilar datos de punto muerto con valores y actividades anteriores</li> <li>• W = Recopilar datos de punto muerto sin valores y actividades anteriores</li> </ul>

Tabla 277. Vista de catálogo SYSCAT.WORKLOADS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
COLLECTLOCKTIMEOUT	CHAR (1)		<p>Especifica los sucesos de tiempo de espera de bloqueo que deben recopilarse por medio del supervisor de sucesos aplicable.</p> <ul style="list-style-type: none"> <li>• H = Recopilar datos de tiempo de espera de bloqueo con actividades anteriores únicamente</li> <li>• N = No recopilar datos de tiempo de espera de bloqueo</li> <li>• V = Recopilar datos de tiempo de espera de bloqueo con valores y actividades anteriores</li> <li>• W = Recopilar datos de tiempo de espera de bloqueo sin valores y actividades anteriores</li> </ul>
COLLECTLOCKWAIT	CHAR (1)		<p>Especifica los sucesos de espera de bloqueo que deben recopilarse por medio del supervisor de sucesos aplicable.</p> <ul style="list-style-type: none"> <li>• H = Recopilar datos de espera de bloqueo con actividades anteriores únicamente</li> <li>• N = No recopilar datos de espera de bloqueo</li> <li>• V = Recopilar datos de espera de bloqueo con valores y actividades anteriores</li> <li>• W = Recopilar datos de espera de bloqueo sin valores y actividades anteriores</li> </ul>
LOCKWAITVALUE	INTEGER		<p>Especifica el tiempo en milisegundos que debe esperar un bloqueo antes de que el supervisor de sucesos aplicable recopile un suceso de bloqueo; 0 cuando COLLECTLOCKWAIT = 'N'</p>
COLLECTACTMETRICS	CHAR (1)		<p>Especifica el nivel de supervisión para las actividades enviadas por una aparición de la carga de trabajo.</p> <ul style="list-style-type: none"> <li>• B = Recopilar métricas básicas de actividad</li> <li>• E = Recopilar métricas ampliadas de actividad</li> <li>• N = Ninguno</li> </ul>

Tabla 277. Vista de catálogo SYSCAT.WORKLOADS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
COLLECTUOWDATAOPTIONS	VARCHAR (32)		<p>Especifica los datos de unidad de trabajo que se recopilan por medio del supervisor de sucesos aplicable. La primera posición de la serie representa si la colección de datos de unidad de trabajo está habilitada.</p> <ul style="list-style-type: none"> <li>• B = Datos básicos de unidad de trabajo habilitados y recopilados</li> <li>• N = No habilitada</li> </ul> <p>Empezando desde la segunda posición, las posiciones de la serie representan una opción ampliada específica:</p> <ul style="list-style-type: none"> <li>• 2 = Lista de referencia de paquete</li> <li>• 3 = Lista de ID ejecutables</li> </ul> <p>Las posiciones que representan una opción ampliada se establecerán en uno de los valores siguientes:</p> <ul style="list-style-type: none"> <li>• Y = La opción ampliada está incluida</li> <li>• N = La opción ampliada no está incluida</li> </ul>
COLLECTUOWDATA	CHAR (1)		<p>Especifica los datos de unidad de trabajo que deben recopilarse por medio del supervisor de sucesos aplicable.</p> <ul style="list-style-type: none"> <li>• B = Recopilar datos base de unidad de trabajo</li> <li>• N = Ninguno</li> <li>• P = Recopilar datos base de unidad de trabajo y la lista de paquetes</li> </ul> <p>Esta columna está en desuso. La información de la columna está disponible desde COLLECTUOWDATAOPTIONS.</p>
EXTERNALNAME	VARCHAR (128)	S	Reservado para una utilización futura.
SECTIONACTUALSOPTIONS	VARCHAR (32)		<p>Especifica qué datos reales de sección se recopilan durante la ejecución de una sección.</p> <p>La primera posición de la serie representa si la colección de datos reales de sección está habilitada.</p> <ul style="list-style-type: none"> <li>• B = Recuentos y estadísticas de cardinalidad de operador básico habilitados y recopilados para cada objeto al que hace referencia la sección (sólo sentencias DML).</li> <li>• N = No habilitado.</li> </ul> <p>La segunda posición siempre es 'N' y se reserva para utilizarlo en el futuro.</p>

Tabla 277. Vista de catálogo SYSCAT.WORKLOADS (continuación)

Nombre de columna	Tipo de datos	Posibil. de nulos	Descripción
COLLECTAGGUOWDATA	CHAR (1)		Especifica los datos agregados de unidad de trabajo que deben capturarse para la carga de trabajo por medio del supervisor de sucesos aplicable. <ul style="list-style-type: none"> <li>• B = Recopilar datos agregados básicos de unidad de trabajo</li> <li>• N = Ninguno</li> </ul>
REMARKS	VARCHAR (254)	S	Comentarios proporcionados por el usuario o valor nulo.

---

## Apéndice A. Normas de denominación generales

Existen normas de denominación para todos los objetos de bases de datos, nombres de usuario, contraseñas, grupos, archivos y vías de acceso. Algunas de estas normas son específicas de la plataforma en la que se trabaja.

Por ejemplo, en cuanto al uso de mayúsculas o minúsculas en los nombres de objetos visibles en el sistema de archivos (bases de datos, instancias, etc.):

- En las plataformas UNIX, los nombres son sensibles a las mayúsculas y las minúsculas. Por ejemplo, /data1 no es el mismo directorio que /DATA1 o /Data1.
- En las plataformas Windows, los nombres no son sensibles a las mayúsculas y las minúsculas. Por ejemplo, \data1 es lo mismo que \DATA1 y \Data1.

Si no se especifica otra cosa, todos los nombres pueden incluir los caracteres siguientes:

- Las letras de la A a la Z y de la a a la z, según la definición del juego de caracteres ASCII básico (de 7 bits). Cuando se utiliza en identificadores para objetos creados con sentencias de SQL, los caracteres en minúsculas de la “a” a la “z” se convierten a mayúsculas, si no se delimitan con comillas (“”).
- Del 0 al 9.
- ! % ( ) { } . - ^ ~ \_ (carácter de subrayado) @, #, \$ y el espacio.
- \ (barra inclinada invertida).

### Restricciones

- No cree nombres que empiecen por un número o un carácter de subrayado.
- No utilice palabras reservadas de SQL como nombres de tabla, vista, columna, índice o ID de autorización.
- Utilice únicamente las letras definidas en el juego de caracteres ASCII básico para los nombres de directorio y de archivo. Aunque el sistema operativo del sistema admita distintas páginas de códigos, es posible que los caracteres no ASCII no funcionen con fiabilidad. La utilización de caracteres no ASCII en un entorno distribuido puede ser especialmente problemática, porque puede que los distintos sistemas utilicen páginas de códigos diferentes.
- Existen otros caracteres especiales que pueden funcionar por separado en función del sistema operativo y de la ubicación donde se trabaje con la base de datos DB2. No obstante, aunque pueden funcionar, no existe ninguna garantía de que lo hagan correctamente. No se recomienda utilizar esos otros caracteres especiales para los nombres de los objetos de la base de datos.
- Los nombres de usuario y de grupo también deben seguir las reglas impuestas por los sistemas operativos específicos. Por ejemplo, en las plataformas Linux y UNIX, los caracteres para los nombres de usuario y de grupo deben ser los caracteres en minúsculas de la a a la z, del 0 al 9 y \_ (carácter de subrayado) para los nombres que no empiecen con un número del 0 al 9.
- Las longitudes deben ser inferiores o iguales a las longitudes que se indican en el apartado “Límites de SQL y XML” de la publicación *Consulta de SQL*.
- **Restricciones sobre el identificador AUTHID:** En DB2 Versión 9.5 y posteriores, puede haber un ID de autorización de 128 bytes. Sin embargo, cuando el ID de autorización se interpreta como un ID de usuario o un nombre de grupo de sistema operativo, se aplican las restricciones de denominación del sistema operativo. Por ejemplo, los sistemas operativos Linux y UNIX pueden contener

un máximo de 8 caracteres y el sistema operativo Windows puede contener un máximo de 30 caracteres para los ID de usuario y los nombres de grupo. Por lo tanto, aunque otorgue un ID de autorización de 128 bytes, no puede conectarse como usuario con ID de autorización. Si graba un plugin de seguridad propio, puede utilizar los tamaños ampliados para el ID de autorización. Por ejemplo, puede asignar al plugin de seguridad un ID de usuario de 30 bytes y éste devolverá un ID de autorización de 128 bytes durante la autenticación al que se podrá conectar.

También debe tener en cuenta las normas de denominación de objetos, las normas de denominación en un entorno con soporte multicultural y las normas de denominación en un entorno Unicode.

---

## Apéndice B. Roles

Los roles simplifican la administración y la gestión de privilegios ofreciendo una posibilidad equivalente como grupos, pero sin las mismas restricciones.

Un rol es un objeto de base de datos que agrupa uno o más privilegios y se puede asignar a usuarios, a grupos, a PUBLIC o a otros roles utilizando una sentencia GRANT o que se puede asignar a un contexto fiable utilizando una sentencia CREATE TRUSTED CONTEXT o ALTER TRUSTED CONTEXT. Se puede especificar un rol para el atributo de conexión SESSION\_USER ROLE en una definición de carga de trabajo.

Los roles proporcionan varias ventajas que facilitan la gestión de privilegios en un sistema de base de datos:

- Los administradores de seguridad pueden controlar el acceso a sus bases de datos de un modo que duplica la estructura de sus organizaciones (pueden crear roles en la base de datos que se correlaciona directamente con las funciones laborales de sus organizaciones).
- Se otorga a los usuarios la pertenencia en los roles que reflejen sus responsabilidades laborales. A medida que sus responsabilidades laborales cambian, su pertenencia como miembros en los roles se puede otorgar y revocar.
- Se simplifica la asignación de privilegios. En lugar de otorgar el mismo conjunto de privilegios a cada usuario individual en una función laboral concreta, el administrador puede otorgar este conjunto de privilegios a un rol que represente esa función laboral y, a continuación, otorgar ese rol a cada usuario de esa función laboral.
- Los privilegios de un rol se pueden actualizar y todos los usuarios a los que se haya otorgado ese rol reciben la actualización; el administrador no tiene que actualizar los privilegios para cada usuario de forma individual.
- Los privilegios y las autorizaciones otorgadas a los roles siempre se utilizan cuando se crean vistas, activadores, tablas de consulta materializada (MQT), rutinas de SQL estático y de SQL, mientras que los privilegios y las autorizaciones otorgados a los grupos (directa o indirectamente) no se utilizan. Ello se debe a que el sistema de base de datos DB2 no puede determinar cuando cambia la pertenencia en un grupo, ya que el grupo se gestiona mediante software de otro proveedor (por ejemplo, el sistema operativo o un directorio LDAP). Puesto que los roles se gestionan dentro de la base de datos, el sistema de base de datos DB2 puede determinar cuando cambia la autorización y actuar en consecuencia. Los roles otorgados a grupos no se tienen en cuenta y, por el mismo motivo, los grupos tampoco se tienen en cuenta.
- Todos los roles asignados a un usuario se habilitan cuando dicho usuario establece una conexión, de manera que todos los privilegios y las autorizaciones otorgados a los roles se tienen en cuenta cuando un usuario se conecta. Los roles no se pueden habilitar o inhabilitar de forma explícita.
- El administrador de seguridad puede delegar la gestión de un rol a otros.

Todos los privilegios y autorizaciones de DB2 que se pueden otorgar dentro de una base de datos se pueden otorgar a un rol. Por ejemplo, se puede otorgar a un rol cualquiera de las autorizaciones y privilegios siguientes:

- Autorizaciones de base de datos DBADM, SECADM, DATAACCESS, ACCESSCTRL, SQLADM, WLMADM, LOAD e IMPLICIT\_SCHEMA

- Autorizaciones de base de datos CONNECT, CREATETAB, CREATE\_NOT\_FENCED, BINDADD, CREATE\_EXTERNAL\_ROUTINE, o QUIESCE\_CONNECT
- Cualquier privilegio de objeto de base de datos (incluido CONTROL)

Los roles de un usuario se habilitan y se tienen en cuenta automáticamente para autorizaciones cuando un usuario se conecta a una base de datos; no tiene que activar un rol utilizando la sentencia SET ROLE. Por ejemplo, cuando se crea una vista, una tabla de consulta materializada (MQT), un activador, un paquete o una rutina de SQL, se aplican los privilegios que se obtienen mediante los roles. No obstante, no se aplican los privilegios que se obtienen mediante los roles otorgados a grupos de los que se es miembro.

Los roles no tienen propietario. El administrador de seguridad puede utilizar la cláusula WITH ADMIN OPTION de la sentencia GRANT para delegar la gestión del rol a otro usuario, de manera que el otro usuario pueda controlar la pertenencia de roles.

## Restricciones

Existen unas cuantas restricciones para el uso de roles:

- Un rol no puede ser propietario de objetos de base de datos.
- Los permisos y los roles otorgados a grupos no se tienen en cuenta al crear los siguientes objetos de base de datos:
  - Paquetes contengan SQL estático
  - Vistas
  - Tablas de consulta materializada (MQT)
  - Activadores
  - Rutinas de SQL

Al crear estos objetos, sólo se tienen en cuenta los roles otorgados al usuario que cree el objeto o a PUBLIC, directa o indirectamente (por ejemplo, mediante una jerarquía de roles)

---

## Apéndice C. Contextos fiables y conexiones fiables

Un contexto fiable es un objeto de base de datos que define una relación fiable para una conexión entre la base de datos y una entidad externa, como un servidor de aplicaciones.

La relación de fiabilidad se basa en el siguiente conjunto de atributos:

- ID de autorización del sistema: representa al usuario que establece una conexión de base de datos.
- Dirección IP (o nombre de dominio): representa el sistema principal desde el que se establece una conexión de base de datos.
- Cifrado de secuencia de datos: representa el valor de cifrado (si lo hay) para la comunicación de datos entre el servidor de bases de datos y el cliente de bases de datos.

Cuando un usuario establece una conexión de base de datos, el sistema de base de datos DB2 comprueba si la conexión coincide con la definición de un objeto de contexto fiable en la base de datos. Cuando se produce una coincidencia, se dice que la conexión de base de datos es fiable.

Una conexión fiable permite que el iniciador de esta conexión fiable adquiera capacidades adicionales que puede que no estén disponibles fuera del ámbito de la conexión fiable. Las capacidades adicionales varían en función de si la conexión fiable es explícita o implícita.

El iniciador de una conexión fiable explícita tiene la capacidad de:

- Cambiar el ID de usuario actual en la conexión por un ID de usuario distinto con o sin autenticación
- Adquirir privilegios adicionales a través de la característica de herencia de roles de contextos fiables

Una conexión fiable implícita es una conexión fiable que no se ha solicitado explícitamente; la conexión fiable implícita se deriva de una petición de conexión normal y no de una petición de conexión fiable explícita. No se necesita ningún cambio en el código de la aplicación para obtener una conexión implícita. Además, tanto si se obtiene una conexión fiable implícita como si no, ello no afecta al código de retorno de conexión (cuando se solicita una conexión fiable explícita, el código de retorno de conexión indica si la petición ha sido satisfactoria o no). El iniciador de una conexión fiable implícita sólo puede adquirir privilegios adicionales a través de la característica de herencia de roles de los contextos fiables; no pueden cambiar el ID de usuario.

### Cómo mejora la seguridad el uso de contextos fiables

El modelo de aplicación de tres niveles amplía el modelo de cliente y servidor de dos niveles estándar colocando el nivel medio entre la aplicación cliente y el servidor de bases de datos. Ha alcanzado mucha popularidad en los últimos años, en concreto con la emergencia de las tecnologías basadas en web y la plataforma Java 2 Enterprise Edition (J2EE). Un ejemplo de un producto de software que admite el modelo de aplicación de tres niveles es IBM WebSphere Application Server (WAS).

En un modelo de aplicación de tres niveles, el nivel medio es el encargado de autenticar los usuarios que ejecutan las aplicaciones cliente y de gestionar las interacciones con el servidor de bases de datos. Tradicionalmente, todas las interacciones con el servidor de bases de datos tienen lugar a través de una conexión de base de datos establecida por el nivel medio utilizando una combinación de un ID de usuario y una credencial que identifican ese nivel medio en el servidor de bases de datos. Esto significa que el servidor de bases de datos utiliza los privilegios de base de datos asociados con el ID de usuario del nivel medio para todas las operaciones de comprobación de autorización y de auditoría que deben tener lugar para cualquier acceso de base de datos, incluido el acceso por parte del nivel medio en nombre de un usuario.

Aunque el modelo de aplicación de tres niveles tiene muchos beneficios, si todas las interacciones con el servidor de bases de datos (por ejemplo, una petición de usuario) tienen lugar con el ID de autorización del nivel medio, surgen varios contratiempos, que se resumen a continuación:

- Pérdida de identidad del usuario  
Algunas empresas prefieren conocer la identidad del usuario real que accede a la base de datos para controlar el acceso.
- Menor responsabilidad del usuario  
La responsabilidad a través de la auditoría es un principio básico en la seguridad de las bases de datos. Si no se conoce la identidad del usuario, es más difícil distinguir las transacciones realizadas por el nivel medio para su propio fin de las realizadas por el nivel medio en nombre de un usuario.
- Demasiados privilegios concedidos al ID de autorización del nivel medio  
El ID de autorización del nivel medio debe tener todos los privilegios necesarios para ejecutar todas las peticiones desde todos los usuarios. Esto plantea el problema de seguridad que supone habilitar usuarios que no necesitan acceso a cierta información para obtener acceso de todos modos.
- Seguridad debilitada  
Además del problema de privilegios planteado en el punto anterior, el enfoque actual exige que se otorgue privilegios al ID de autorización que utiliza el nivel medio que se debe conectar en todos los recursos a los que puedan acceder las peticiones de usuario. Si ese ID de autorización de nivel medio se ve comprometido en algún momento, se expondrán todos esos recursos.
- "Vertidos" entre usuarios de la misma conexión  
Los cambios realizados por un usuario anterior pueden afectar al usuario actual.

Sin lugar a dudas, es necesario un mecanismo por el cual la identidad y los privilegios de base de datos del usuario real se utilicen para las peticiones de base de datos realizadas por el nivel medio en nombre de ese usuario. El enfoque más directo para alcanzar este objetivo sería que el nivel medio estableciera una conexión nueva mediante el ID y la contraseña del usuario y, a continuación, dirigiera las peticiones del usuario a través de esa conexión. Aunque es sencillo, este enfoque tiene bastantes inconvenientes, entre los que destacan:

- No se puede aplicar en determinados niveles medios. Muchos servidores de nivel medio no tienen las credenciales de autenticación de usuario necesarias para establecer una conexión.
- Sobrecarga de rendimiento. Existe una sobrecarga de rendimiento obvia asociada con la creación de una nueva conexión física y la reautenticación del usuario en el servidor de bases de datos.
- Sobrecarga de mantenimiento. En las situaciones en las que no se utiliza una configuración de seguridad centralizada o el inicio de sesión único, se produce

una sobrecarga de mantenimiento al tener dos definiciones de usuario (una en el nivel medio y otra en el servidor). Ello exige cambiar las contraseñas en distintas ubicaciones.

La capacidad de contextos fiables corrige este problema. El administrador de seguridad puede crear un objeto de contexto fiable en la base de datos que defina una relación de confianza entre la base de datos y el nivel medio. El nivel medio puede establecer, entonces, una conexión fiable explícita con la base de datos, lo que da al nivel medio la capacidad de cambiar el ID de usuario actual en la conexión por otro ID de usuario, con o sin autenticación. Además de resolver el problema de confirmación de la identidad del usuario, los contextos fiables ofrecen otra ventaja: la capacidad de controlar cuándo hay un privilegio disponible para un usuario de base de datos. La falta de control sobre cuándo están disponibles los privilegios para un usuario puede debilitar la seguridad global. Por ejemplo, se pueden utilizar privilegios con fines distintos a los pretendidos originalmente. El administrador de seguridad puede asignar uno o más privilegios a un rol y asignar ese rol a un objeto de contexto fiable. Sólo las conexiones de base de datos fiables (explícitas o implícitas) que coincidan con la definición de dicho contexto fiable podrán beneficiarse de los privilegios asociados con ese rol.

## Mejora del rendimiento

Cuando se utilizan conexiones fiables, se puede maximizar el rendimiento gracias a las ventajas siguientes:

- No se establece ninguna conexión nueva cuando cambia el ID de usuario actual de la conexión.
- Si la definición de contexto fiable no exige la autenticación del ID de usuario al que se va a cambiar, no se incurre en la sobrecarga asociada con la autenticación de un usuario nuevo en el servidor de bases de datos.

## Ejemplo de creación de un contexto fiable

Supongamos que el administrador de seguridad crea el objeto de contexto fiable siguiente:

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID USUARIO2
  ATTRIBUTES (ADDRESS '192.0.2.1')
  DEFAULT ROLE managerRole
  ENABLE
```

Si el usuario *usuario1* solicita una conexión fiable de la dirección IP 192.0.2.1, el sistema de base de datos DB2 devuelve un aviso (SQLSTATE 01679, SQLCODE +20360) para indicar que no se ha podido establecer una conexión fiable y que el usuario *usuario1* simplemente ha obtenido una conexión no fiable. Sin embargo, si el usuario *usuario2* solicita una conexión fiable desde la dirección IP 192.0.2.1, la petición se reconoce porque el contexto fiable CTX1 cumple los atributos de conexión. Ahora que el usuario *usuario2* ha establecido una conexión fiable, puede adquirir todos los privilegios y las autorizaciones asociadas con el rol de contexto fiable *managerRole*. Puede que estos privilegios y autorizaciones no estén disponibles para el usuario *usuario2* fuera del ámbito de esta conexión fiable.



---

## Apéndice D. Visión general de la información técnica de DB2

La información técnica de DB2 está disponible en diversos formatos a los que se puede acceder de varias maneras.

La información técnica de DB2 está disponible a través de las herramientas y los métodos siguientes:

- DB2Centro de información
  - Temas (Tareas, concepto y temas de consulta)
  - Programas de ejemplo
  - Guías de aprendizaje
- Manuales de DB2
  - Archivos PDF (descargables)
  - Archivos PDF (desde el DVD con PDF de DB2)
  - Manuales en copia impresa
- Ayuda de la línea de mandatos
  - Ayuda de mandatos
  - Ayuda de mensajes

**Nota:** Los temas del Centro de información de DB2 se actualizan con más frecuencia que los manuales en PDF o impresos. Para obtener la información más actualizada, instale las actualizaciones de la documentación conforme pasen a estar disponibles, o consulte el Centro de información de DB2 en [ibm.com](http://ibm.com).

Puede acceder a información técnica adicional de DB2 como, por ejemplo, notas técnicas, documentos técnicos y publicaciones IBM Redbooks en línea, en el sitio [ibm.com](http://ibm.com). Acceda al sitio de la biblioteca de software de gestión de información de DB2 en <http://www.ibm.com/software/data/sw-library/>.

### Comentarios sobre la documentación

Agradecemos los comentarios sobre la documentación de DB2. Si tiene sugerencias sobre cómo podemos mejorar la documentación de DB2, envíe un correo electrónico a [db2docs@ca.ibm.com](mailto:db2docs@ca.ibm.com). El personal encargado de la documentación de DB2 lee todos los comentarios de los usuarios, pero no puede responderlos directamente. Proporcione ejemplos específicos siempre que sea posible de manera que podamos comprender mejor sus problemas. Si realiza comentarios sobre un tema o archivo de ayuda determinado, incluya el título del tema y el URL.

No utilice esta dirección de correo electrónico para contactar con el Soporte al cliente de DB2. Si tiene un problema técnico de DB2 que no está tratado por la documentación, consulte al centro local de servicio técnico de IBM para obtener ayuda.

## Biblioteca técnica de DB2 en copia impresa o en formato PDF

Las tablas siguientes describen la biblioteca de DB2 que está disponible en el Centro de publicaciones de IBM en [www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss). Los manuales de DB2 Versión 10.1 en inglés y las versiones traducidas en formato PDF se pueden descargar del sitio web [www.ibm.com/support/docview.wss?rs=71&uid=swg2700947](http://www.ibm.com/support/docview.wss?rs=71&uid=swg2700947).

Aunque las tablas identifican los manuales en copia impresa disponibles, puede que dichos manuales no estén disponibles en su país o región.

El número de documento se incrementa cada vez que se actualiza un manual. Asegúrese de que lee la versión más reciente de los manuales, tal como aparece a continuación:

**Nota:** El *Centro de información de DB2* se actualiza con más frecuencia que los manuales en PDF o impresos.

Tabla 278. Información técnica de DB2

Nombre	Número de documento	Copia impresa disponible	Última actualización
<i>Consulta de las API administrativas</i>	SC11-8067-00	Sí	Abril de 2012
<i>Rutinas y vistas administrativas</i>	SC11-8068-00	No	Abril de 2012
<i>Call Level Interface Guide and Reference Volume 1</i>	SC27-3866-00	Sí	Abril de 2012
<i>Call Level Interface Guide and Reference Volume 2</i>	SC27-3867-00	Sí	Abril de 2012
<i>Consulta de mandatos</i>	SC11-8069-00	Sí	Abril de 2012
<i>Database Administration Concepts and Configuration Reference</i>	SC27-3871-00	Sí	Abril de 2012
<i>Data Movement Utilities Guide and Reference</i>	SC27-3869-00	Sí	Abril de 2012
<i>Database Monitoring Guide and Reference</i>	SC27-3887-00	Sí	Abril de 2012
<i>Data Recovery and High Availability Guide and Reference</i>	SC27-3870-00	Sí	Abril de 2012
<i>Database Security Guide</i>	SC27-3872-00	Sí	Abril de 2012
<i>Guía y consulta de DB2 Workload Management</i>	SC11-8079-00	Sí	Abril de 2012
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-3873-00	Sí	Abril de 2012
<i>Developing Embedded SQL Applications</i>	SC27-3874-00	Sí	Abril de 2012
<i>Desarrollo de aplicaciones Java</i>	SC11-8065-00	Sí	Abril de 2012

Tabla 278. Información técnica de DB2 (continuación)

Nombre	Número de documento	Copia impresa disponible	Última actualización
<i>Desarrollo de aplicaciones Perl, PHP, Python y Ruby on Rails</i>	SC11-8066-00	No	Abril de 2012
<i>Developing User-defined Routines (SQL and External)</i>	SC27-3877-00	Sí	Abril de 2012
<i>Getting Started with Database Application Development</i>	GI13-2046-00	Sí	Abril de 2012
<i>Iniciación a la instalación y administración de DB2 en Linux y Windows</i>	GI13-1946-00	Sí	Abril de 2012
<i>Globalization Guide</i>	SC27-3878-00	Sí	Abril de 2012
<i>Instalación de servidores DB2</i>	GC11-8073-00	Sí	Abril de 2012
<i>Instalación de clientes de IBM Data Server</i>	GC11-8074-00	No	Abril de 2012
<i>Consulta de mensajes Volumen 1</i>	SC11-8079-00	No	Abril de 2012
<i>Consulta de mensajes Volumen 2</i>	SC11-8080-00	No	Abril de 2012
<i>Net Search Extender Guía de administración y del usuario</i>	SC11-8082-00	No	Abril de 2012
<i>Partitioning and Clustering Guide</i>	SC27-3882-00	Sí	Abril de 2012
<i>pureXML Guide</i>	SC27-3892-00	Sí	Abril de 2012
<i>Spatial Extender Guía del usuario y manual de consulta</i>	SC11-8081-00	No	Abril de 2012
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-3896-00	Sí	Abril de 2012
<i>Consulta de SQL - Volumen 1</i>	SC11-8070-00	Sí	Abril de 2012
<i>Consulta de SQL - Volumen 2</i>	SC11-8071-00	Sí	Abril de 2012
<i>Guía de Text Search</i>	SC11-8083-00	Sí	Abril de 2012
<i>Troubleshooting and Tuning Database Performance</i>	SC27-3889-00	Sí	Abril de 2012
<i>Actualización a DB2 Versión 10.1</i>	SC11-8072-00	Sí	Abril de 2012
<i>Novedades en DB2 Versión 10.1</i>	SC11-8078-00	Sí	Abril de 2012
<i>XQuery Reference</i>	SC27-3893-00	No	Abril de 2012

Tabla 279. Información técnica específica de DB2 Connect

Nombre	Número de documento	Copia impresa disponible	Última actualización
DB2 Connect Instalación y configuración de DB2 Connect Personal Edition	SC11-8075-00	Sí	Abril de 2012
DB2 Connect Instalación y configuración de servidores DB2 Connect	SC11-8076-00	Sí	Abril de 2012
Guía del usuario de DB2 Connect	SC11-8077-00	Sí	Abril de 2012

---

## Visualización de la ayuda para estados de SQL desde el procesador de línea de mandatos

Los productos DB2 devuelven un valor de SQLSTATE para las condiciones que pueden ser el resultado de una sentencia de SQL. La ayuda de SQLSTATE explica los significados de los estados de SQL y los códigos de las clases de estados de SQL.

### Procedimiento

Para iniciar la ayuda para estados de SQL, abra el procesador de línea de mandatos y entre:

*? sqlstate o ? código de clase*

donde *sqlstate* representa un estado de SQL válido de cinco dígitos y *código de clase* representa los dos primeros dígitos del estado de SQL.

Por ejemplo, *? 08003* visualiza la ayuda para el estado de SQL 08003, y *? 08* visualiza la ayuda para el código de clase 08.

---

## Acceso a diferentes versiones del Centro de información de DB2

La documentación correspondiente a otras versiones de los productos DB2 se encuentra en otros centros de información en [ibm.com](http://ibm.com).

### Acerca de esta tarea

Para los temas de DB2 Versión 10.1, el URL del *Centro de información de DB2* es <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1>.

Para los temas de DB2 Versión 9.8, el URL del *Centro de información de DB2* es <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

Para los temas de DB2 Versión 9.7, el URL del *Centro de información de DB2* es <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

Para los temas de DB2 Versión 9.5, el URL del *Centro de información de DB2* es <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>.

Para los temas de DB2 Versión 9.1, el URL del *Centro de información de DB2* es <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

Para los temas de DB2 Versión 8, vaya al URL del *Centro de información de DB2* en el sitio: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

---

## Actualización del Centro de información de DB2 instalado en el sistema o en el servidor de intranet

El Centro de información de DB2 instalado en local se debe actualizar periódicamente.

### Antes de empezar

Ya debe haber un Centro de información de DB2 Versión 10.1 instalado. Para obtener información adicional, consulte el tema “Instalación del Centro de información de DB2 utilizando el Asistente de instalación de DB2” en la publicación *Instalación de servidores DB2*. Todos los requisitos previos y las restricciones aplicables a la instalación del Centro de información se aplican también a la actualización del Centro de información.

### Acerca de esta tarea

Un Centro de información de DB2 existente se puede actualizar automática o manualmente:

- Las actualizaciones automáticas actualizan las funciones y los idiomas del Centro de información existentes. Una ventaja de las actualizaciones automáticas es que el Centro de información deja de estar disponible durante un período de tiempo más breve a cuando se realiza la actualización manual. Además, la ejecución de las actualizaciones automáticas se puede configurar como parte de otros trabajos de proceso por lotes que se ejecutan periódicamente.
- Las actualizaciones manuales se pueden utilizar para actualizar las funciones y los idiomas existentes del Centro de información. Las actualizaciones automáticas reducen el tiempo de inactividad durante el proceso de actualización. Sin embargo, debe utilizar el proceso manual cuando desee añadir funciones o idiomas. Por ejemplo, un Centro de información en local se instaló inicialmente tanto en inglés como en francés, y ahora se desea instalar el idioma alemán. Con la actualización manual, se instalará el alemán y se actualizarán además las funciones y los idiomas existentes del Centro de información. No obstante, la actualización manual requiere que el usuario detenga, actualice y reinicie manualmente el Centro de información. El Centro de información no está disponible durante todo el proceso de actualización. En el proceso de actualización automática, el Centro de información incurre en una interrupción de servicio para reiniciar el Centro de información solo después de la actualización.

Este tema detalla el proceso de las actualizaciones automáticas. Para conocer las instrucciones para la actualización manual, consulte el tema “Actualización manual del Centro de información de DB2 instalado en el sistema o en el servidor de intranet”.

### Procedimiento

Para actualizar automáticamente el Centro de información de DB2 instalado en el sistema o en el servidor de Intranet:

1. En sistemas operativos Linux,

- a. Navegue hasta la vía de acceso en la que está instalado el Centro de información. Por omisión, el Centro de información de DB2 se instala en el directorio `/opt/ibm/db2ic/V10.1`.
  - b. Navegue desde el directorio de instalación al directorio `doc/bin`.
  - c. Ejecute el script `update-ic`:  
`update-ic`
2. En sistemas operativos Windows,
    - a. Abra una ventana de mandatos.
    - b. Navegue hasta la vía de acceso en la que está instalado el Centro de información. Por omisión, el Centro de información de DB2 se instala en el directorio `<Archivos de programa>\IBM\DB2 Information Center\Versión 10.1`, siendo `<Archivos de programa>` la ubicación del directorio Archivos de programa.
    - c. Navegue desde el directorio de instalación al directorio `doc\bin`.
    - d. Ejecute el archivo `update-ic.bat`:  
`update-ic.bat`

## Resultados

El Centro de información de DB2 se reinicia automáticamente. Si hay actualizaciones disponibles, el Centro de información muestra los temas nuevos y actualizados. Si no había actualizaciones del Centro de información disponibles, se añade un mensaje al archivo de anotaciones cronológicas. El archivo de anotaciones cronológicas está ubicado en el directorio `doc\eclipse\configuration`. El nombre del archivo de anotaciones cronológicas es un número generado aleatoriamente. Por ejemplo, `1239053440785.log`.

---

## Actualización manual del Centro de información de DB2 instalado en el sistema o en el servidor de intranet

Si ha instalado localmente el Centro de información de DB2 localmente, puede obtener e instalar actualizaciones de la documentación de IBM.

### Acerca de esta tarea

Para actualizar manualmente el *Centro de información de DB2* instalado localmente es preciso que:

1. Detenga el *Centro de información de DB2* en el sistema, y reinicie el Centro de información en modalidad autónoma. La ejecución del Centro de información en modalidad autónoma impide que otros usuarios de la red accedan al Centro de información y permite al usuario aplicar las actualizaciones. La versión de estación de trabajo del Centro de información de DB2 siempre se ejecuta en modalidad autónoma.
2. Utilice la función Actualizar para ver qué actualizaciones están disponibles. Si hay actualizaciones que debe instalar, puede utilizar la función Actualizar para obtenerlas y actualizarlas.

**Nota:** Si su entorno requiere la instalación de actualizaciones del *Centro de información de DB2* en una máquina no conectada a Internet, duplique el sitio de actualizaciones en un sistema de archivos local utilizando una máquina que esté conectada a Internet y tenga instalado el *Centro de información de DB2*. Si muchos usuarios en la red van a instalar las actualizaciones de la documentación, puede reducir el tiempo necesario para realizar las

actualizaciones duplicando también el sitio de actualizaciones localmente y creando un proxy para el sitio de actualizaciones.

Si hay paquetes de actualización disponibles, utilice la característica Actualizar para obtener los paquetes. Sin embargo, la característica Actualizar sólo está disponible en modalidad autónoma.

3. Detenga el Centro de información autónomo y reinicie el *Centro de información de DB2* en su equipo.

**Nota:** En Windows 2008 y Windows Vista (y posterior), los mandatos listados más abajo deben ejecutarse como administrador. Para abrir un indicador de mandatos o una herramienta gráfica con privilegios de administrador completos, pulse con el botón derecho del ratón el atajo y, a continuación, seleccione **Ejecutar como administrador**.

## Procedimiento

Para actualizar el *Centro de información de DB2* instalado en el sistema o en el servidor de Intranet:

1. Detenga el *Centro de información de DB2*.
  - En Windows, pulse **Inicio > Panel de control > Herramientas administrativas > Servicios**. A continuación, pulse con el botón derecho del ratón en el servicio **Centro de información de DB2** y seleccione **Detener**.
  - En Linux, especifique el mandato siguiente:  
`/etc/init.d/db2icdv10 stop`
2. Inicie el Centro de información en modalidad autónoma.
  - En Windows:
    - a. Abra una ventana de mandatos.
    - b. Navegue hasta la vía de acceso en la que está instalado el Centro de información. Por omisión, el *Centro de información de DB2* se instala en el directorio `Archivos_de_programa\IBM\DB2 Information Center\Versión 10.1`, siendo `Archivos_de_programa` la ubicación del directorio Archivos de programa.
    - c. Navegue desde el directorio de instalación al directorio `doc\bin`.
    - d. Ejecute el archivo `help_start.bat`:  
`help_start.bat`
  - En Linux:
    - a. Navegue hasta la vía de acceso en la que está instalado el Centro de información. Por omisión, el *Centro de información de DB2* se instala en el directorio `/opt/ibm/db2ic/V10.1`.
    - b. Navegue desde el directorio de instalación al directorio `doc/bin`.
    - c. Ejecute el script `help_start`:  
`help_start`

Se abre el navegador Web por omisión de los sistemas para visualizar el Centro de información autónomo.
3. Pulse en el botón **Actualizar** (🔄). (JavaScript debe estar habilitado en el navegador.) En la derecha del panel del Centro de información, pulse en **Buscar actualizaciones**. Se visualiza una lista de actualizaciones para la documentación existente.
4. Para iniciar el proceso de instalación, compruebe las selecciones que desee instalar y, a continuación, pulse **Instalar actualizaciones**.
5. Cuando finalice el proceso de instalación, pulse **Finalizar**.

6. Detenga el Centro de información autónomo:

- En Windows, navegue hasta el directorio `doc\bin` del directorio de instalación y ejecute el archivo `help_end.bat`:  
`help_end.bat`

**Nota:** El archivo `help_end` de proceso por lotes contiene los mandatos necesarios para detener sin peligro los procesos que se iniciaron mediante el archivo `help_start` de proceso por lotes. No utilice `Control-C` ni ningún otro método para detener `help_start.bat`.

- En Linux, navegue hasta el directorio `doc/bin` del directorio de instalación y ejecute el script `help_end`:  
`help_end`

**Nota:** El script `help_end` contiene los mandatos necesarios para detener sin peligro los procesos que se iniciaron mediante el script `help_start`. No utilice ningún otro método para detener el script `help_start`.

7. Reinicie el *Centro de información de DB2*.

- En Windows, pulse **Inicio > Panel de control > Herramientas administrativas > Servicios**. A continuación, pulse con el botón derecho del ratón en el servicio **Centro de información de DB2** y seleccione **Iniciar**.
- En Linux, especifique el mandato siguiente:  
`/etc/init.d/db2icdv10 start`

## Resultados

El *Centro de información de DB2* actualizado muestra los temas nuevos y actualizados.

---

## Guías de aprendizaje de DB2

Las guías de aprendizaje de DB2 le ayudan a conocer diversos aspectos de productos de base de datos DB2. Se proporcionan instrucciones paso a paso a través de lecciones.

### Antes de comenzar

Puede ver la versión XHTML de la guía de aprendizaje desde el Centro de información en el sitio <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/>.

Algunas lecciones utilizan datos o código de ejemplo. Consulte la guía de aprendizaje para obtener una descripción de los prerrequisitos para las tareas específicas.

### Guías de aprendizaje de DB2

Para ver la guía de aprendizaje, pulse el título.

#### **“pureXML” en *pureXML Guide***

Configure una base de datos DB2 para almacenar datos XML y realizar operaciones básicas con el almacén de datos XML nativos.

---

## Información de resolución de problemas de DB2

Existe una gran variedad de información para la resolución y determinación de problemas para ayudarle en la utilización de productos de base de datos DB2.

### Documentación de DB2

Puede encontrar información sobre la resolución de problemas en la publicación *Troubleshooting and Tuning Database Performance* o en la sección sobre conceptos fundamentales sobre bases de datos del *Centro de información de DB2*, que contiene:

- Información sobre cómo aislar e identificar problemas con programas de utilidad y herramientas de diagnóstico de DB2.
- Soluciones a algunos de los problemas más comunes.
- Consejo para ayudarlo a resolver problemas que podría encontrar en los productos de base de datos DB2

### Portal de Soporte de IBM

Consulte el portal de soporte de IBM si tiene problemas y desea obtener ayuda para encontrar las causas y soluciones posibles. El sitio de soporte técnico tiene enlaces a las publicaciones más recientes de DB2, notas técnicas, Informes autorizados de análisis del programa (APAR o arreglos de defectos), fixpacks y otros recursos. Puede buscar en esta base de conocimiento para encontrar posibles soluciones a los problemas.

Acceda al portal de Soporte de IBM en el sitio [http://www.ibm.com/support/entry/portal/Overview/Software/Information\\_Management/DB2\\_for\\_Linux,\\_UNIX\\_and\\_Windows](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows)

---

## Términos y condiciones

Los permisos para utilizar estas publicaciones se otorgan sujetos a los siguientes términos y condiciones.

**Aplicación:** Además de las condiciones de uso del sitio web de IBM, se aplican estos términos y condiciones.

**Uso personal:** Puede reproducir estas publicaciones para su uso personal, no comercial, siempre y cuando se mantengan los avisos sobre la propiedad. No puede distribuir, visualizar o realizar trabajos derivados de estas publicaciones, o de partes de las mismas, sin el consentimiento expreso de IBM.

**Uso comercial:** Puede reproducir, distribuir y visualizar estas publicaciones únicamente dentro de su empresa, siempre y cuando se mantengan todos los avisos sobre la propiedad. No puede realizar trabajos derivados de estas publicaciones, ni reproducirlas, distribuirlas o visualizarlas, ni de partes de las mismas fuera de su empresa, sin el consentimiento expreso de IBM.

**Derechos:** Excepto lo expresamente concedido en este permiso, no se conceden otros permisos, licencias ni derechos, explícitos o implícitos, sobre las publicaciones ni sobre ninguna información, datos, software u otra propiedad intelectual contenida en el mismo.

IBM se reserva el derecho de retirar los permisos aquí concedidos cuando, a su discreción, el uso de las publicaciones sea en detrimento de su interés o cuando, según determine IBM, las instrucciones anteriores no se cumplan correctamente.

No puede descargar, exportar ni volver a exportar esta información excepto en el caso de cumplimiento total con todas las leyes y regulaciones vigentes, incluyendo todas las leyes y regulaciones sobre exportación de los Estados Unidos.

IBM NO GARANTIZA EL CONTENIDO DE ESTAS PUBLICACIONES. LAS PUBLICACIONES SE PROPORCIONAN "TAL CUAL" Y SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUYENDO PERO SIN LIMITARSE A LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN, NO VULNERACIÓN E IDONEIDAD PARA UN FIN DETERMINADO.

**Marcas registradas de IBM:** IBM, el logotipo de IBM e ibm.com son marcas registradas de International Business Machines Corp., que se han registrado en muchas otras jurisdicciones. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas. Puede consultarse en línea una lista actualizada de las marcas registradas de IBM en la web en [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

## Apéndice E. Avisos

Esta información ha sido desarrollada para productos y servicios que se ofrecen en Estados Unidos de América. La información acerca de productos que no son IBM se basa en la información disponible cuando se publicó este documento por primera vez y está sujeta a cambio.

Es posible que IBM no comercialice en otros países algunos productos, servicios o características descritos en este manual. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implicar que sólo se pueda utilizar dicho producto, programa o servicio de IBM. En su lugar se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patentes en tramitación que afecten al tema tratado en este documento. La posesión de este documento no confiere ninguna licencia sobre dichas patentes. Puede realizar consultas sobre licencias escribiendo a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
EE.UU.

Para realizar consultas sobre licencias referentes a información de juegos de caracteres de doble byte (DBCS), puede ponerse en contacto con el Departamento de Propiedad Intelectual de IBM de su país o escribir a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japón

**El párrafo siguiente no es aplicable al Reino Unido ni a ningún país/región en donde tales disposiciones sean incompatibles con la legislación local:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUIDAS, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener inexactitudes técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede

efectuar, en cualquier momento y sin previo aviso, mejoras y cambios en los productos y programas descritos en esta publicación.

Las referencias hechas en esta publicación a sitios web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos sitios web. La información de esos sitios web no forma parte de la información del presente producto de IBM y la utilización de esos sitios web se realiza bajo la responsabilidad del usuario.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciatarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Canada Limited  
U59/3600  
3600 Steeles Avenue East  
Markham, Ontario L3R 9Z7  
CANADÁ

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él, los proporciona IBM según los términos del Acuerdo de Cliente de IBM, el Acuerdo Internacional de Programas Bajo Licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se obtuvieron en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles comercialmente. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente manual deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación referente a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Este manual puede contener ejemplos de datos e informes que se utilizan en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos

estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es totalmente fortuita.

#### LICENCIA DE COPYRIGHT:

Este manual contiene programas de aplicaciones de ejemplo escritos en lenguaje fuente, que muestran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM con la intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas. Los programas de ejemplo se proporcionan "TAL CUAL", sin ningún tipo de garantía. IBM no se hará responsable de los daños derivados de la utilización que haga el usuario de los programas de ejemplo.

Cada copia o parte de estos programas de ejemplo o cualquier trabajo derivado debe incluir una nota de copyright como la siguiente:

© (*nombre de la empresa*) (*año*). Partes de este código proceden de programas de ejemplo de IBM Corp. © Copyright IBM Corp. *\_entre el o los años\_*. Reservados todos los derechos.

#### **Marcas registradas**

IBM, el logotipo de IBM e [ibm.com](http://ibm.com) son marcas registradas de International Business Machines Corp., que se han registrado en muchas otras jurisdicciones. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas. La lista actual de marcas registradas de IBM está disponible en la web, en "Copyright and trademark information", en la dirección [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Los siguientes términos son marcas registradas de otras empresas.

- Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/o en otros países.
- Java y todos los logotipos y marcas registradas basadas en Java son marcas registradas de Oracle, sus filiales o ambos.
- UNIX es una marca registrada de The Open Group en los Estados Unidos y/o en otros países.
- Intel, el logotipo de Intel, Intel Inside, el logotipo de Intel Inside, Celeron, Intel SpeedStep, Itanium y Pentium son marcas registradas de Intel Corporation o de sus empresas subsidiarias en Estados Unidos y en otros países.
- Microsoft, Windows, Windows NT y el logotipo de Windows son marcas registradas de Microsoft Corporation en los Estados Unidos y/o en otros países.

Otros nombres de empresas, productos o servicios, pueden ser marcas registradas o marcas de servicio de otras empresas.



---

# Índice

## A

- acciones de trabajo
  - alterar 111
  - asignar a actividades de la base de datos 121
  - asociación con otros objetos (ejemplo) 103
  - conjuntos de acciones de trabajo 117
  - crear 107
  - descartar 113
  - inhabilitar 112
  - umbrales 121
- actividades
  - agregar métricas
    - en elementos hijo 377
  - aplicación de acciones de trabajo 121
  - Asesor de diseño 301
  - asignación a clases de trabajo 64
  - cancelar 383
    - caso de ejemplo 380
    - procedimiento 241, 302
  - colocar en cola 153
  - correlacionar con clases de servicio 78
  - coste estimado alto y tiempo de ejecución alto
    - caso de ejemplo 386
  - ejecución larga
    - caso de ejemplo 380
  - ejemplo de análisis 66
  - elementos de supervisor
    - act\_throughput 446
    - act\_total 447
    - activity\_collected 448
    - activity\_id 448
    - activity\_secondary\_id 449
    - activity\_type 450
    - coord\_act\_aborted\_total 462
    - coord\_act\_completed\_total 462
    - coord\_act\_rejected\_total 469
    - parent\_activity\_id 478
  - estados en clases de servicio 86
  - informes de inteligencia empresarial
    - controlar (caso de ejemplo) 170, 171
  - malintencionadas 302
  - recopilación de datos
    - ejemplo 304
    - procedimiento 299
  - umbrales 143
  - visión general 15
- ACTIVITYTOTALTIME, umbral de actividad
  - detalles 144
- actualizaciones
  - Centro de información de DB2 559, 560
- agentes
  - uso de clase de servicio 101
- AGGSQLTEMPSPACE, umbral
  - detalles 153
- agregación
  - datos
    - ejemplo 254
- anotaciones cronológicas
  - elementos de supervisor
    - uow\_log\_space\_used 508

- API
  - sqleseti
    - asignación de cargas de trabajo 41
- Asesor de diseño
  - importar información de actividad 301
- asignador
  - gestión de carga de trabajo
    - visión general 176
- asignador de gestión de carga de trabajo
  - AIX
    - entornos de micropartición 197
  - cuotas de CPU fijas 200
    - habilitar y establecer 207
  - cuotas de CPU flexibles 209
    - habilitar y establecer 213
  - detalles 177
  - elementos de supervisor
    - cpu\_limit 471
    - cpu\_share\_type 471
    - cpu\_shares 471
    - cpu\_utilization 472
    - cpu\_velocity 472
    - estimated\_cpu\_entitlement 475
    - total\_disp\_run\_queue\_time 501
  - entornos de micropartición 197
  - entornos de varios miembros
    - comportamientos 197
  - garantizar la máxima precisión de planificación 199
  - habilitar 198
  - Límite de CPU 215
    - valor 229
  - nivel de simultaneidad de asignación 234
    - valor 236
  - supervisión
    - detalles 237
  - utilización de CPU mínima 231
    - valor 233
  - visión general 176
- asignador de gestión de carga de trabajo, parámetro de configuración 525
- avisos 565
- ayuda
  - sentencias SQL 558

## B

- bases de datos particionadas
  - elementos de supervisor
    - coord\_partition\_num 470
- bloqueos
  - elementos de supervisor
    - uow\_lock\_wait\_time 507

## C

- CALL, sentencia
  - clasificación por esquema 55
- cancelación de actividades
  - clases de servicio
    - ejemplo 386

- capturar previamente
  - prioridad de clases de servicio 83
- cargas de trabajo
  - alterar 34
  - asignación
    - detalles 24
    - ejemplos 41
  - asignación de conexión a la carga de trabajo de administración por omisión 29
  - comparación con conjuntos de acciones de trabajo 125
  - crear 31
  - datos de supervisión 262
  - descartar 37
  - ejemplos
    - analizar la ralentización del sistema 376
    - asignación cuando existen varias cargas de trabajo 47
    - asignación cuando los atributos de carga de trabajo tienen sólo un valor 45
    - asignación cuando los atributos de la carga de trabajo tienen varios valores 50
  - elementos de supervisor
    - wlo\_completed\_total 516
    - workload\_id 519
    - workload\_name 520
    - workload\_occurrence\_id 521
    - workload\_occurrence\_state 522
  - habilitar 35
  - inhabilitar 36
  - orden de evaluación 24
  - permitir acceso a la base de datos 37
  - por omisión 27
  - posición en la lista de cargas de trabajo 24
  - prevención de acceso a la base de datos 38
  - privilegio USAGE
    - conceder 39
    - revocar 40
  - visión general 19
- cargas de trabajo por omisión 27
- casos de ejemplo
  - agregar métricas de actividad
    - en elementos hijo 377
  - cancelar
    - actividades 383
- Centro de información de DB2
  - actualización 559, 560
  - versiones 558
- clases de servicio
  - alterar
    - cambios producidos en restablecimiento de estadísticas 291
    - procedimiento 89
  - analizar la ralentización del sistema 99
  - correlacionar actividades 78
  - crear 88
  - descartar 93
  - ejemplos 94, 99
  - entidades cuyo seguimiento no realizan 87
  - estadísticas puntuales 253
  - estados de actividad 86
  - estados de conexión 86
  - prioridad de agente 82
  - prioridad de agrupación de almacenamientos intermedios 84
  - prioridad de captación previa 83
  - subclases de servicio por omisión 77
  - superclases de servicio por omisión 77
- clases de trabajo
  - alterar 61
  - asignar actividades 64
  - crear 58
  - descartar 61
  - ejemplos
    - asociación con otros objetos 103
    - definidas con la palabra clave ALL 68
  - orden de evaluación 64
  - umbrales soportados 65
  - visión general 52
- colas
  - captación previa 83
- CONCURRENTDBCOORDACTIVITIES, umbral agregado
  - detalles 154
- CONCURRENTWORKLOADACTIVITIES, umbral agregado 156
- CONCURRENTWORKLOADOCCURRENCES, umbral agregado 158
- conexiones
  - asignar
    - atributos de conexión de carga de trabajo con varios valores 50
    - carga de trabajo de administración por omisión 29
  - cargas de trabajo 24
  - correlacionar con cargas de trabajo
    - ejemplo 45
  - estados en clases de servicio 86
  - transitorio 159
- conexiones fiables
  - visión general 551
- conjuntos de acciones de trabajo
  - acciones de trabajo que especifican umbrales 121
  - alterar 114
  - control de simultaneidad 123
  - crear 113
  - descartar 116
  - dominio y acciones de trabajo permitidas 117
  - ejemplos
    - asociación con otros objetos 103
    - conjunto de acciones de trabajo y umbral de base de datos 128
    - determinar tipos de trabajo en ejecución 129
  - inhabilitar 116
  - nivel de carga de trabajo
    - control de simultaneidad 123
  - visión general 105
- conjuntos de clases de trabajo
  - alterar 62
  - asociación con otros objetos (ejemplo) 103
  - crear 62
  - descartar 63
  - gestionar actividades de DML (ejemplo) 67
  - orden de evaluación de clases de trabajo 64
  - visión general 55
- CONNECTIONIDLETIME, umbral de conexión 142
- consultas
  - elementos de supervisor
    - queue\_assignments\_total 480
    - queue\_size\_top 480
    - queue\_time\_total 481
- contextos fiables
  - visión general 551
- CPUTIME, umbral de actividad
  - detalles 145
- CPUTIMEINSC, umbral de actividad 146

- cuotas de cpu
  - elementos de supervisor
    - cpu\_shares 471
- cuotas de CPU de asignador de gestor de carga de trabajo,
  - parámetro de configuración 527
- cuotas de CPU fijas
  - habilitar y establecer 207
- cuotas de CPU flexibles
  - habilitar y establecer 213

## D

- DATATAGINSC, umbral de actividad
  - detalles 147
- DDL
  - sentencias
    - gestión de carga de trabajo de DB2 18
- denominación, convenios de
  - generales 547
- descartar
  - cargas de trabajo 37
  - clases de servicio 93
  - conjuntos de acciones de trabajo 116
  - conjuntos de clases de trabajo 63
  - plantillas de histograma 286
  - umbrales 139
- desconexión de todas las aplicaciones
  - clases de servicio
    - ejemplo 387
- determinación de problemas
  - guías de aprendizaje 563
  - información disponible 563
- documentación
  - archivos PDF 556
  - copia impresa 556
  - términos y condiciones de uso 563
  - visión general 555

## E

- ejemplos
  - clase de trabajo definida con la palabra clave ALL 68
  - clases de servicio 94
  - conjunto de acciones de trabajo y umbral 128
  - correlacionar conexiones con cargas de trabajo 45
  - gestionar actividades mediante un conjunto de clases de trabajo 67
- elemento de supervisor de identificador de invocación de
  - sentencia 493
- elementos de supervisor
  - concurrentdbcoordactivities\_db\_threshold\_value 455
  - concurrentdbcoordactivities\_db\_threshold\_violated 455
  - concurrentdbcoordactivities\_subclass\_threshold\_id 456
- elementos de supervisor de actividades
  - act\_exec\_time 445
  - act\_remapped\_in
    - detalles 445
  - act\_remapped\_out
    - detalles 445
  - actividades
    - act\_throughput 446
    - act\_total 447
    - activity\_collected 448
    - activity\_id 448
    - activity\_secondary\_id 449
    - activity\_type 450

- elementos de supervisor de actividades (*continuación*)
  - actividades (*continuación*)
    - coord\_act\_aborted\_total 462
    - coord\_act\_completed\_total 462
    - coord\_act\_rejected\_total 469
    - parent\_activity\_id 478
  - agg\_temp\_tablespace\_top 450
  - asignador de WLM
    - cpu\_limit 471
    - cpu\_share\_type 471
    - cpu\_shares 471
    - cpu\_utilization 472
    - cpu\_velocity 472
    - estimated\_cpu\_entitlement 475
    - total\_disp\_run\_queue\_time 501
  - bloqueos
    - uow\_lock\_wait\_time 507
  - cargas de trabajo
    - wlo\_completed\_total 516
    - workload\_id 519
    - workload\_name 520
    - workload\_occurrence\_id 521
    - workload\_occurrence\_state 522
  - CONCURRENTDBCOORDACTIVITIES, umbral
    - concurrentdbcoordactivities\_wl\_was\_threshold\_id 460
    - concurrentdbcoordactivities\_wl\_was\_threshold\_queued 460
    - concurrentdbcoordactivities\_wl\_was\_threshold\_value 461
    - concurrentdbcoordactivities\_wl\_was\_threshold\_violated 461
    - concurrentdbcoordactivities\_db\_threshold\_id 454
    - concurrentdbcoordactivities\_subclass\_threshold\_value 457
    - concurrentdbcoordactivities\_subclass\_threshold\_queued 456
    - concurrentdbcoordactivities\_subclass\_threshold\_violated 457
    - concurrentdbcoordactivities\_superclass\_threshold\_id 458
    - concurrentdbcoordactivities\_superclass\_threshold\_queued 458
    - concurrentdbcoordactivities\_superclass\_threshold\_value 459
    - concurrentdbcoordactivities\_superclass\_threshold\_violated 459
  - consultas
    - queue\_assignments\_total 480
    - queue\_size\_top 480
    - queue\_time\_total 481
  - coord\_act\_est\_cost\_avg 463
  - coord\_act\_exec\_time\_avg 464
  - coord\_act\_interarrival\_time\_avg 465
  - coord\_act\_lifetime\_avg 466
  - coord\_act\_queue\_time\_avg 468
  - destination\_service\_class\_id 474
  - espacio de anotaciones cronológicas
    - uow\_log\_space\_used 508
  - filas
    - rows\_fetched 483
    - rows\_modified 483
    - rows\_returned 485
  - gestión de carga de trabajo
    - asignaciones totales en cola 513
    - tiempo total en cola 515
    - visión general 444

elementos de supervisor de actividades (*continuación*)

- gestión de carga de trabajo (*continuación*)
  - wl\_work\_action\_set\_id 512
  - wl\_work\_class\_id 513
- histogramas
  - extremo superior 501
  - histogram\_type 475
  - number\_in\_bin 477
- hora
  - prep\_time 479
  - time\_completed 500
  - time\_created 500
  - time\_of\_violation 500
  - time\_started 500
- hora de activación
  - last\_wlm\_reset 476
- identificadores
  - arm\_correlator 451
  - bin\_id 451
  - db\_work\_action\_set\_id 473
  - db\_work\_class\_id 474
  - sc\_work\_action\_set\_id 487
  - sc\_work\_class\_id 488
  - service\_class\_id 488
  - work\_action\_set\_id 517
  - work\_class\_id 518
- indicaciones de fecha y hora
  - activate\_timestamp 447
  - statistics\_timestamp 492
- marcas de límite
  - act\_cpu\_time\_top 444
  - act\_rows\_read\_top 445
  - concurrent\_act\_top 452
  - concurrent\_connection\_top 452
  - concurrent\_wlo\_act\_top 453
  - concurrent\_wlo\_top 453
  - coord\_act\_lifetime\_top 467
  - cost\_estimate\_top 470
  - rows\_returned\_top 486
  - temp\_tablespace\_top 494
  - uow\_total\_time\_top 511
- nombres
  - service\_subclass\_name 489
  - service\_superclass\_name 491
  - work\_action\_set\_name 517
  - work\_class\_name 518
- num\_remaps 477
- particiones
  - coord\_partition\_num 470
- rangos
  - fin 451
- request\_exec\_time\_avg 481
- rutinas
  - routine\_id 482
- secciones
  - section\_env 488
- sentencias
  - stmt\_invocation\_id 493
- source\_service\_class\_id 492
- umbrales
  - num\_threshold\_violations 477
  - thresh\_violations 494
  - threshold\_action 496
  - threshold\_domain 496
  - threshold\_maxvalue 497
  - threshold\_name 497
  - threshold\_predicate 498

elementos de supervisor de actividades (*continuación*)

- umbrales (*continuación*)
  - threshold\_queuesize 499
  - thresholdid 499
- unidades de trabajo (UOW)
  - parent\_uow\_id 478
  - uow\_comp\_status 504
  - uow\_completed\_total 503
  - uow\_elapsed\_time 505
  - uow\_id 505
  - uow\_lifetime\_avg 506
  - uow\_start\_time 508
  - uow\_status 509
  - uow\_stop\_time 510
  - uow\_throughput 511
- elementos de supervisor de marca de límite
  - act\_cpu\_time\_top 444
  - act\_rows\_read\_top 445
  - concurrent\_act\_top 452
  - concurrent\_connection\_top 452
  - concurrent\_wlo\_act\_top 453
  - concurrent\_wlo\_top 453
  - coord\_act\_lifetime\_top 467
  - cost\_estimate\_top 470
  - rows\_returned\_top 486
  - temp\_tablespace\_top 494
  - uow\_total\_time\_top 511
- espacios de tablas
  - SQLTEMPSPACE, umbral 152
- esquemas
  - clasificación de la sentencia CALL 55
- estadísticas
  - objetos de la gestión de carga de trabajo de DB2 265
  - recopilación
    - gestión de carga de trabajo 288
    - supervisor de sucesos 256
  - ESTIMATEDSQLCOST, umbral de actividad 148

## F

filas

- elementos de supervisor
  - rows\_fetched 483
  - rows\_modified 483
  - rows\_returned 485
  - rows\_returned\_top 486

funciones

- tabla
  - Función de tabla
    - WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 420
    - WLM\_GET\_ACTIVITY\_DETAILS 402
    - WLM\_GET\_QUEUE\_STATS 409
    - WLM\_GET\_SERVICE\_CLASS\_AGENTS 412
    - WLM\_GET\_SERVICE\_SUBCLASS\_STATS 423
    - WLM\_GET\_SERVICE\_SUPERCLASS\_STATS 430
    - WLM\_GET\_WORK\_ACTION\_SET\_STATS 432
    - WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES 433
    - WLM\_GET\_WORKLOAD\_STATS 438

funciones de tabla

- agregar datos 254
- determinación de la información de cola del umbral de WLM
  - ejemplo 254
- ejemplo de utilización 248

- funciones de tabla (*continuación*)
  - Función de tabla
    - WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES 420
  - supervisión en diferentes niveles
    - ejemplo 250
  - supervisor de instantáneas 293
  - WLM\_COLLECT\_STATS 291
  - WLM\_GET\_SERVICE\_CLASS\_AGENTS 412
  - WLM\_GET\_SERVICE\_SUBCLASS\_STATS 423
  - WLM\_GET\_WORKLOAD\_STATS 438

## G

- gestión de carga de trabajo
  - ejemplos
    - cancelación de todas las actividades 386
    - desconexión de todas las aplicaciones 387
  - violaciones de umbral
    - notificaciones por correo electrónico 296
- gestión de carga de trabajo de DB2
  - acciones de trabajo
    - alterar 111
    - asignar a actividades de la base de datos 121
    - crear 107
    - definir para un conjunto de acciones de trabajo 117
    - descartar 113
    - inhabilitar 112
    - umbrales 121
  - actividades
    - aislar 327
    - análisis de cargas de trabajo por tipo de actividad (ejemplo) 66
    - asignación a clases de trabajo 64
    - asignación de acciones de trabajo 121
    - cancelar 241, 302, 357
    - controlar informes de inteligencia empresarial (caso de ejemplo) 170, 171
    - correlacionar con clases de servicio 78
    - descubrir 358
    - diferenciar 337
    - identificar actividades con costes estimados bajos y tiempos de ejecución altos (ejemplo) 386
    - identificar actividades de ejecución larga (ejemplo) 380
    - importar información de actividad en el Asesor de diseño 301
    - malintencionadas 302
    - recopilación de datos 299, 361
    - recopilación de datos (ejemplo) 304
    - umbrales 143
    - visión general 15
    - volver a correlacionar 166
  - agentes
    - prioridad 82
    - uso por clase de servicio (ejemplo) 101
  - agregados ampliados 367
  - agregar métricas
    - actividades 377
  - ajuste
    - información de planificación de capacidad disponible (ejemplo) 389
    - información de planificación de capacidad no disponible (ejemplo) 391
  - aplicación de ejemplo 375
  - asignador
    - detalles 177
    - entornos de varios miembros 197

- gestión de carga de trabajo de DB2 (*continuación*)
  - asignador (*continuación*)
    - garantizar la máxima precisión de planificación 199
    - habilitar 198
    - visión general 176
  - asignador de gestión de carga de trabajo
    - cuotas de CPU fijas 200
    - cuotas de CPU flexibles 209
    - límite de CPU 215
    - utilización de CPU mínima 231
  - autorización WLMADM
    - visión general 4
  - cancelar
    - actividades 383
  - cargas de trabajo
    - acceso a la base de datos, permitir 37
    - acceso a la base de datos, prevención 38
    - alterar 34
    - analizar la ralentización del sistema (ejemplo) 376
    - asignación (ejemplo) 41
    - asignación cuando existen varias cargas de trabajo (ejemplo) 47
    - asignación cuando los atributos de carga de trabajo tienen varios valores (ejemplo) 50
    - asignación de conexión a la carga de trabajo de administración por omisión 29
    - crear 31
    - descartar 37
    - habilitar 35
    - inhabilitar 36
    - por omisión 27
    - visión general 19
  - clases de servicio
    - alterar 89
    - analizar la ralentización del sistema (ejemplo) 99
    - asignación de recursos 73
    - crear 88
    - descartar 93
    - ejemplo 94
    - entidades sin seguimiento 87
    - obtener estadísticas puntuales (ejemplo) 253
    - prioridad de captación previa 83, 84
  - clases de trabajo
    - alterar 61
    - clasificación de la sentencia CALL por esquema 55
    - crear 58
    - definidas con la palabra clave ALL (ejemplo) 68
    - descartar 61
    - orden de evaluación 64
    - visión general 52
  - colas de actividades 153
  - conexiones
    - asignación a cargas de trabajo 24
    - estados en clases de servicio 86
  - conjuntos de acciones de trabajo
    - acciones de trabajo 102
    - alterar 114
    - crear 113
    - descartar 116
    - determinar tipos de trabajo en ejecución (ejemplo) 129
    - inhabilitar 116
    - utilización con umbral de base de datos (ejemplo) 128
    - visión general 105
  - conjuntos de clases de trabajo
    - alterar 62
    - crear 62
    - descartar 63

- gestión de carga de trabajo de DB2 (*continuación*)
  - conjuntos de clases de trabajo (*continuación*)
    - gestión de actividades de DML (ejemplo) 67
    - visión general 55
  - cuotas de CPU fijas
    - detalles 200
    - habilitar y establecer 207
  - cuotas de CPU flexibles
    - detalles 209
    - habilitar y establecer 213
  - datos históricos 363
  - DB2 Governor
    - relación 4
  - definición de niveles de clase de servicio
    - scripts de muestra 166
    - visión general 161
  - disminución de la prioridad
    - detalles 161
    - scripts de muestra 166
  - dominios 1
  - elementos de supervisor 444
    - asignaciones totales en cola 513
    - tiempo total en cola 515
  - estadísticas
    - recopilación utilizando un supervisor de sucesos de estadísticas 288
    - restablecer 291
    - visión general 265
  - etapa de gestión 73, 102, 130
  - funciones de tabla
    - agregar datos (ejemplo) 254
    - ejemplo para obtener información de cola del umbral de WLM 254
    - entender qué se ejecuta en el servidor de datos (ejemplo) 248
    - información de operación 243
    - utilizar con funciones de tabla del supervisor de instantáneas 293
  - guía de aprendizaje 321
  - herramienta de análisis histórico 288
  - histogramas
    - calcular promedios y desviación estándar (ejemplo) 286
    - crear 345
    - visión general 278
  - informes 363
  - integración con los gestores de carga de trabajo del sistema operativo 307
  - integración de gestión de carga de trabajo de Linux 314
  - integración del gestor de carga de trabajo de AIX 307
  - Límite de CPU
    - detalles 215
    - valor 229
  - métricas 292
  - modelado del rendimiento 303
  - nivel de simultaneidad de asignación
    - detalles 234
    - valor 236
  - plantillas de histograma
    - alterar 285
    - crear 284
    - descartar 286
  - preguntas frecuentes 4
  - privilegio USAGE en cargas de trabajo
    - conceder 39
    - revocar 40
- gestión de carga de trabajo de DB2 (*continuación*)
  - procedimientos almacenados
    - WLM\_CANCEL\_ACTIVITY 265
    - WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 265
    - WLM\_COLLECT\_STATS 265
    - WLM\_SET\_CLIENT\_INFO 265
  - propiedad de objetos 18
  - Query Patroller 4
  - sentencias DDL 18
  - SET WORKLOAD, mandato 523
  - subclases de servicio 73
  - supervisión
    - comportamiento del sistema en diferentes niveles (ejemplo) 250
    - datos 262
    - niveles de clase de servicio 276
    - recopilación de datos 321
    - retardos de aplicación 354
    - supervisores de sucesos 256
    - tiempo real 243
    - trabajo 243
    - visión general 243
  - supervisores de sucesos
    - visión general 256
  - trabajo
    - controlar 73
    - identificar 15
  - umbrales
    - actividad 143
    - actividades malintencionadas 334
    - ACTIVITYTOTALTIME 144
    - AGGSQLTEMPSPACE 153
    - agregado 153
    - alterar 139
    - CONCURRENTDBCOORDACTIVITIES 154
    - CONCURRENTWORKLOADACTIVITIES 156
    - CONCURRENTWORKLOADOCCURRENCES 158
    - conexión 142
    - conjunto de acciones de trabajo y umbral de base de datos (ejemplo) 128
    - CONNECTIONIDLETIME 142
    - controlar el trabajo 130
    - CPUTIME 145
    - CPUTIMEINSC 146
    - crear 138
    - DATATAGINSC 147
    - descartar 139
    - ESTIMATEDSQLCOST 148
    - gestionar recursos de base de datos entre departamentos (ejemplo) 140
    - infracciones, supervisar 295
    - orden de evaluación 135
    - resolución de ámbitos 143
    - resumen 134
    - SQLROWSREAD 149
    - SQLROWSREADINSC 150
    - SQLROWSRETURNED 151
    - SQLTEMPSPACE 152
    - TOTALMEMBERCONNECTIONS 158
    - TOTALSCMEMBERCONNECTIONS 159
    - UOWTOTALTIME 160
    - visión general 130
    - volver a correlacionar actividades 161, 173
  - unidad de trabajo
    - asignación de carga de trabajo cuando existen varias cargas de trabajo (ejemplo) 47

- gestión de carga de trabajo de DB2 (*continuación*)
  - utilización de CPU mínima
    - detalles 231
    - valor 233
  - visión general 1
- gestor de carga de trabajo de AIX
  - integrar gestión de carga de trabajo de DB2 307
  - prioridad de procesador 82
- guías de aprendizaje
  - determinación de problemas 563
  - lista 562
  - pureXML 562
  - resolución de problemas 563

## H

- histogramas
  - ejemplo 286
  - elementos de supervisor
    - extremo superior 501
    - histogram\_type 475
    - number\_in\_bin 477
  - visión general 278
- hora
  - elementos de supervisor
    - prep\_time 479
    - time\_completed 500
    - time\_created 500
    - time\_of\_violation 500
    - time\_started 500

## I

- identificador AUTHID
  - restricciones 547
- identificadores
  - elementos de supervisor
    - arm\_correlator 451
    - bin\_id 451
    - db\_work\_action\_set\_id 473
    - db\_work\_class\_id 474
    - sc\_work\_action\_set\_id 487
    - sc\_work\_class\_id 488
    - service\_class\_id 488
    - work\_action\_set\_id 517
    - work\_class\_id 518
- indicaciones de fecha y hora
  - elementos de supervisor
    - activate\_timestamp 447
    - statistics\_timestamp 492
    - uow\_start\_time 508
    - uow\_stop\_time 510
- intervalos
  - finalidad 286

## L

- límite de cpu
  - elementos de supervisor
    - cpu\_limit 471
- Límite de CPU
  - valor 229
- Linux
  - integración de gestión de carga de trabajo con la gestión de carga de trabajo de DB2 314

## M

- mandatos
  - SET WORKLOAD
    - detalles 523
    - realización de correcciones 29
- métricas
  - objetos de la gestión de carga de trabajo de DB2 292

## N

- nivel de simultaneidad de asignación
  - valor 236
- nombres
  - elementos de supervisor
    - service\_subclass\_name 489
    - service\_superclass\_name 491
    - work\_action\_set\_name 517
    - work\_class\_name 518
  - nombres de archivo
    - generales 547
- notificaciones
  - violaciones de umbral 296

## O

- objetos de base de datos
  - gestión de carga de trabajo de DB2 18
  - roles 549
- orden de evaluación
  - cargas de trabajo 24
  - umbrales de la gestión de carga de trabajo de DB2 135

## P

- parámetro de configuración wlm\_dispatcher 525
- parámetros de configuración
  - wlm\_collect\_int 524
  - wlm\_disp\_concur 526
  - wlm\_disp\_cpu\_shares 527
  - wlm\_disp\_min\_util 527
  - wlm\_dispatcher 525
- plantillas de histograma
  - alterar 285, 291
  - crear 284
  - descartar 286
- prioridad de agente
  - clases de servicio 82
- prioridad de agrupación de almacenamientos intermedios
  - clases de servicio 84
- privilegios
  - roles 549
- procedimientos
  - WLM\_CANCEL\_ACTIVITY 397
  - WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 398
  - WLM\_COLLECT\_STATS 400
  - WLM\_SET\_CLIENT\_INFO 441
- procedimientos almacenados
  - WLM\_CANCEL\_ACTIVITY 265
  - WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS 265
  - WLM\_COLLECT\_STATS 265
  - WLM\_SET\_CLIENT\_INFO 265
- promedio de vida útil de las unidades de trabajo
  - elementos de supervisor
    - uow\_lifetime\_avg 506

propiedad  
objetos de la gestión de carga de trabajo de DB2 18

## R

rangos  
elementos de supervisor  
fin 451

REMAP ACTIVITY, acción  
definir 161  
scripts de muestra 166

rendimiento  
gestión de carga de trabajo de DB2  
ejemplos 389, 391  
modelado del rendimiento 303

rendimiento de las actividades  
elementos de supervisor  
act\_throughput 446

rendimiento de las unidades de trabajo  
elementos de supervisor  
uow\_throughput 511

representación de datos  
agregación 254

resolución de problemas  
guías de aprendizaje 563  
información en línea 563

revocar  
privilegio USAGE en carga de trabajo 40

roles  
detalles 549

routine\_id, elemento de supervisor 482

rutinas  
elementos de supervisor  
routine\_id 482  
WLM\_CANCEL\_ACTIVITY, ejemplo 380

## S

secciones  
elementos de supervisor  
section\_env 488

seguridad  
contextos fiables 551

sentencias de SQL  
elementos de supervisor  
stmt\_invocation\_id 493

sentencias SQL  
ayuda  
visualización 558

SET WORKLOAD, mandato  
asignar conexión a la carga de trabajo de administración  
por omisión 29  
detalles 523

simultaneidad de hebras de asignador de gestor de carga de  
trabajo, parámetro de configuración 526

sistemas operativos  
integrar gestión de carga de trabajo de DB2 307

sqlietisi, API  
asignación de cargas de trabajo 41

SQLROWSREAD, umbral de actividad  
detalles 149

SQLROWSREADINSC, umbral de actividad 150

SQLROWSRETURNED, umbral de actividad 151

SQLTEMPSPACE, umbral de actividad  
detalles 152

subclases de servicio  
alterar 89  
crear 88  
datos de supervisión 262  
descartar 93

superclase de servicio de mantenimiento por omisión  
visión general 77

superclase de servicio de sistema por omisión 77

superclase de servicio de usuario por omisión 77

superclases de servicio  
alterar 89  
crear 88  
datos de supervisión 262  
descartar 93

supervisión  
asignador de gestión de carga de trabajo  
detalles 237  
rendimiento 237  
tipos de cargas de trabajo 237

datos 262

disminución de la prioridad 276

entidades cuyo seguimiento no realizan clases de  
servicio 87

niveles de clase de servicio 276

tendencias históricas 256

tiempo real 243

visión general 243

supervisión de instantáneas  
funciones de tabla suplementarias 293

supervisor de sucesos de violaciones de umbral 256

supervisores de sucesos  
datos de actividad 299  
gestión de carga de trabajo de DB2  
recopilación de estadísticas 288

tipos 256

violaciones de umbral 295

supervisores de sucesos de actividad 256

SYSDEFAULTMAINTENANCECLASS, superclase de servicio  
visión general 77

SYSDEFAULTSYSTEMCLASS, superclase de servicio  
visión general 77

SYSDEFAULTUSERCLASS, superclase de servicio  
visión general 77

## T

términos y condiciones  
publicaciones 563

tiempo total en cola de asignador  
elementos de supervisor  
total\_disp\_run\_queue\_time 501

tipo de cuota de cpu  
elementos de supervisor  
cpu\_share\_type 471

titularidad de CPU estimada  
elementos de supervisor  
estimated\_cpu\_entitlement 475

total de unidades de trabajo completadas  
elementos de supervisor  
uow\_completed\_total 503

TOTALMEMBERCONNECTIONS, umbral de conexión 158

TOTALSCMEMBERCONNECTIONS, umbral de  
conexión 159

## U

- umbrales
  - acción 130
  - acciones de trabajo 121
  - actividad 143
  - ACTIVITYTOTALTIME 144
  - AGGSQLTEMPSPACE
    - detalles 153
  - agregado 130, 153
  - alterar 139
  - ámbito de implantación 134
  - clasificaciones de trabajo 65
  - colocar actividades en cola 153
  - CONCURRENTDBCOORDACTIVITIES 154
  - CONCURRENTWORKLOADACTIVITIES 156
  - CONCURRENTWORKLOADOCCURRENCES 158
  - conexión 142
  - CONNECTIONIDLETIME 142
  - CPUTIME
    - detalles 145
  - CPUTIMEINSC 146
  - crear 138
  - DATATAGINSC 147
  - descartar 139
  - dominio
    - visión general 134
  - ejemplos
    - conjunto de acciones de trabajo y umbral de base de datos 128
    - uso 140
  - elementos de supervisor
    - num\_threshold\_violations 477
    - thresh\_violations 494
    - threshold\_action 496
    - threshold\_domain 496
    - threshold\_maxvalue 497
    - threshold\_name 497
    - threshold\_predicate 498
    - threshold\_queue\_size 499
    - thresholdid 499
  - ESTIMATEDSQLCOST 148
  - orden de evaluación 135
  - resolución de ámbitos de actividad 143
  - SQLROWSREAD
    - detalles 149
  - SQLROWSREADINSC 150
  - SQLROWSRETURNED 151
  - SQLTEMPSPACE 152
  - supervisar infracciones 295
  - TOTALMEMBERCONNECTIONS 158
  - TOTALSCMEMBERCONNECTIONS 159
  - unidad de trabajo 160
  - UOWTOTALTIME 160
  - visión general 130
  - volver a correlacionar actividades 161, 173
- umbrales agregados
  - definición 153
- umbrales de tipo en clase de servicio 161
- unidades de trabajo
  - asignación a la carga de trabajo por omisión 27
  - coincidencia con carga de trabajo
    - ejemplo 47
  - elementos de supervisor
    - parent\_uow\_id 478
    - uow\_comp\_status 504
    - uow\_completed\_total 503
    - uow\_elapsed\_time 505

- unidades de trabajo (*continuación*)
  - elementos de supervisor (*continuación*)
    - uow\_id 505
    - uow\_lifetime\_avg 506
    - uow\_lock\_wait\_time 507
    - uow\_log\_space\_used 508
    - uow\_start\_time 508
    - uow\_status 509
    - uow\_stop\_time 510
    - uow\_throughput 511
  - umbrales 160
- UOWTOTALTIME, umbral 160
- utilización de cpu
  - elementos de supervisor
    - cpu\_utilization 472
- utilización de CPU mínima
  - valor 233
- utilización de CPU mínima del asignador del gestor de carga de trabajo, parámetro de configuración 527

## V

- velocidad de cpu
  - elementos de supervisor
    - cpu\_velocity 472
- vías de acceso
  - normas de denominación 547
- violaciones de umbral
  - notificaciones por correo electrónico 296
- vistas de catálogo
  - HISTOGRAMTEMPLATEBINS 528
  - HISTOGRAMTEMPLATES 529
  - HISTOGRAMTEMPLATEUSE 529
  - SERVICECLASSES 530
  - THRESHOLDS 533
  - WORKACTIONS 535
  - WORKACTIONSETS 538
  - WORKCLASSES 539
  - WORKCLASSESETS 540
  - WORKLOADAUTH 540
  - WORKLOADCONNATTR 541
  - WORKLOADS 541
- volver a correlacionar actividades
  - detalles 173
  - scripts de muestra 166

## W

- WLM\_CANCEL\_ACTIVITY, procedimiento 397
- WLM\_CAPTURE\_ACTIVITY\_IN\_PROGRESS, procedimiento 398
- wlm\_collect\_int, parámetro de configuración de base de datos 524
- WLM\_COLLECT\_STATS, procedimiento
  - detalles 400
  - restablecer estadísticas 291
- wlm\_disp\_concur, parámetro de configuración 526
- wlm\_disp\_concur, parámetro de configuración del gestor de bases de datos
  - valor 236
- wlm\_disp\_cpu\_shares, parámetro de configuración 527
- wlm\_disp\_min\_util, parámetro de configuración 527
- wlm\_disp\_min\_util, parámetro de configuración del gestor de bases de datos
  - valor 233

- wlm\_dispatcher, parámetro de configuración del gestor de bases de datos
  - valor 198
- WLM\_GET\_ACTIVITY\_DETAILS, función de tabla 402
- WLM\_GET\_QUEUE\_STATS, función de tabla 409
- WLM\_GET\_SERVICE\_CLASS\_AGENTS, función de tabla
  - detalles 412
  - información de cola de WLM
    - ejemplo 254
  - investigar uso de agentes por clase de servicio (ejemplo) 101
- WLM\_GET\_SERVICE\_CLASS\_WORKLOAD\_OCCURRENCES, función de tabla
  - detalles 420
  - ejemplos
    - agregar datos 254
- WLM\_GET\_SERVICE\_SUBCLASS\_STATS, función de tabla
  - detalles 423
  - ejemplos
    - agregar datos 254
    - analizar la ralentización del sistema 99, 376
    - obtener estadísticas puntuales 253
- WLM\_GET\_SERVICE\_SUPERCLASS\_STATS, función de tabla 430
- WLM\_GET\_WORK\_ACTION\_SET\_STATS, función de tabla
  - análisis de cargas de trabajo (ejemplos) 66
  - detalles 432
- WLM\_GET\_WORKLOAD\_OCCURRENCE\_ACTIVITIES, función de tabla
  - agregar datos (ejemplos) 254
  - descripción 433
  - ejemplos
    - identificar actividades de ejecución larga 380
- WLM\_GET\_WORKLOAD\_STATS, función de tabla 438
- WLM\_SET\_CLIENT\_INFO, procedimiento 441
- WLMADM (administración de carga de trabajo), autorización
  - detalles 4





SC11-8079-00



Spine information:

IBM DB2 10.1 para Linux, UNIX y Windows

Guía y consulta de DB2 Workload Management

