

IBM DB2 10.1
for Linux, UNIX and Windows

*DB2 Workload Management
Handbuch und Referenz
Aktualisierung: Januar 2013*



IBM DB2 10.1
for Linux, UNIX and Windows

*DB2 Workload Management
Handbuch und Referenz
Aktualisierung: Januar 2013*



Anmerkung

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in Anhang E, „Bemerkungen“, auf Seite 559 gelesen werden.

Impressum

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 10.1 for Linux, UNIX, and Windows, DB2 Workload Management and Reference Guide,
IBM Form SC27-3891-01,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2007, 2013

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Januar 2013

Inhaltsverzeichnis

Zu diesem Handbuch. ix

Kapitel 1. Einführung in DB2-Workload-Management-Konzepte 1

Workload-Management-Domänen 1
Berechtigung WLADM (Workload-Management-Administrator). 4
Häufig gestellte Fragen zum DB2-Workload-Management 4

Kapitel 2. Identifizieren der Arbeit . . . 15

Aktivitäten 15
DDL-Anweisungen für Workload-Management . . 18
Identifizieren der Arbeit nach Ursprung mithilfe von Workloads. 19
 Workloadzuordnung 24
 Standardworkloads. 26
 Erstellen einer Workload 31
 Ändern einer Workload 33
 Aktivieren einer Workload 35
 Inaktivieren einer Workload 36
 Löschen einer Workload 36
 Erlauben des Datenbankzugriffs für Vorkommen einer Workload 37
 Verhindern des Datenbankzugriffs für Vorkommen einer Workload 38
 Erteilen des Zugriffsrechts USAGE für eine Workload 39
 Widerrufen des Zugriffsrechts USAGE für eine Workload 40
 Beispiel: Workloadzuordnung 41
 Beispiel: Workloadzuordnung bei Workloadattributen mit Einzelwerten 45
 Beispiel: Workloadzuordnung für eine UOW, wenn mehrere Workloads vorhanden sind . . . 47
 Beispiel: Workloadzuordnung bei Workloadattributen mit mehreren Werten 50
Identifizieren der Arbeit nach Typ, Aufwand oder Datenzugriff mithilfe von Arbeitsklassen 52
 Arbeitsklassen und Arbeitsklassensets 55
 Bewertungsreihenfolge von Arbeitsklassen in einem Arbeitsklassenset 63
 Zuordnen von Aktivitäten zu Arbeitsklassen . . 64
 Von Schwellenwerten unterstützte Arbeitsklassifikationen 65
 Beispiel: Analysieren von Workloads nach Aktivitätstyp 66
 Beispiel: Verwalten bestimmter Aktivitätstypen mithilfe von Arbeitsklassensets 67
 Beispiel: Verwenden einer Arbeitsklasse mit dem Schlüsselwort ALL 68

Kapitel 3. Aktivitätenverwaltung 71

Ressourcenzuordnung mit Serviceklassen 71
 Standardservicesuperklassen und -unterklassen 74

Zuordnung von Aktivitäten zu Serviceklassen . . 76
Agentenpriorität von Serviceklassen 80
Vorablesezugriffspriorität von Serviceklassen . . 81
Pufferpoolpriorität von Serviceklassen 82
Status von Verbindungen und Aktivitäten in einer Serviceklasse 83
Nicht von Serviceklassen verwaltete Entitäten auf Systemebene 84
Erstellen einer Serviceklasse 85
Ändern einer Serviceklasse 87
Löschen einer Serviceklasse 90
Beispiel: Verwenden von Serviceklassen 92
Beispiel: Analysieren einer serviceklassenbezogenen Leistungsminderung des Systems 97
Beispiel: Untersuchen der Agentenauslastung nach Serviceklasse 98
Anwenden von Steuerelementen auf Typen von Aktivitäten mit Arbeitsaktionssets 99
 Beziehung zwischen Arbeitsklassen, Arbeitsklassensets, Arbeitsaktionen und Arbeitsaktionssets und deren Zuordnung zu anderen DB2-Objekten 100
 Arbeitsaktionen und Arbeitsaktionssets. . . . 102
 Arbeitsaktionen und die Domäne des Arbeitsaktionssets 114
 In Arbeitsaktionen verwendbare Schwellenwerte 118
 Anwenden von Arbeitsaktionen auf Datenbankaktivitäten 118
 Steuerung des gemeinsamen Zugriffs auf Workloadenebene mithilfe von Arbeitsaktionssets . . 120
 Vergleich von Workload und Arbeitsaktionsset 122
 Beispiel: Verwenden von Datenbankarbeitsaktionssets und Datenbankschwellenwerten . . . 125
 Beispiel: Festlegen des Typs der auszuführenden Arbeit mithilfe von Arbeitsaktionssets 127
Steuern von Arbeit mit Schwellenwerten 128
 Schwellenwertdomäne und Umsetzungsbereich 132
 Bewertungsreihenfolge für Schwellenwerte . . 134
 Erstellen eines Schwellenwerts 136
 Ändern eines Schwellenwerts 137
 Löschen eines Schwellenwerts. 138
 Beispiel: Verwenden von Schwellenwerten. . . 139
 Verbindungsschwellenwerte 141
 Aktivitätsschwellenwerte 142
 Zusammengefasste Schwellenwerte 152
 UOW-Schwellenwerte 160
Prioritätssteuerung nach Verweildauer für laufende Aktivitäten 161
 Beispielskripts für die Prioritätssteuerung nach Verweildauer 166
 Neuzuordnung von Aktivitäten zwischen Serviceunterklassen 172
Workload-Management-Dispatcher - Übersicht . . 175
 Workload-Management-Dispatcher 176
 Feste CPU-Anteile. 200
 Veränderliche CPU-Anteile 208

Aktivieren und Festlegen von CPU-Anteilen . . .	212
CPU-Grenzwert	214
Minimale CPU-Ressourcenauslastung zum Klas- sifizieren der Serviceklasse als aktiv	230
Dispatcher-Thread-Parallelität	233
Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers	236
Abbrechen von Aktivitäten	240

Kapitel 4. Überwachung und Eingriff 241

Echtzeitüberwachung mit Tabellenfunktionen . . .	241
Beispiel: Verwenden von DB2-Workload-Ma- nagement-Tabellenfunktionen	246
Beispiel: Verwenden von DB2-Workload-Ma- nagement-Tabellenfunktionen zum Überwachen des aktuellen Systemverhaltens auf verschiede- nen Ebenen	248
Beispiel: Abrufen von Zeitpunktstatistikdaten aus Serviceklassen.	251
Beispiel: Zusammenfassen von Daten mithilfe von DB2-Workload-Management-Tabellenfunktio- nen	252
Beispiel: Ermitteln der von einem WLM-Schwel- lenwert in die Warteschlange eingereihten Akti- vitäten sowie der Reihenfolge dieser Aktivitäten.	253
Langzeitüberwachung mit WLM-Ereignismonitoren	254
Verfügbare Überwachungsdaten	260
Gespeicherte Prozeduren des DB2-Workload-Ma- nagements	263
Statistikdaten für DB2-Workload-Management-Ob- jekte	264
Statistikerfassung und Überwachung mit neu zugeordneten Aktivitäten	274
Histogramme beim Workload-Management	276
Protokollanalysetool	286
Erfassen von Workload-Management-Statistikda- ten mithilfe eines Ereignismonitors für Statistik- daten	286
Zurücksetzen von Statistikdaten für DB2-Work- load-Management-Objekte	289
Überwachungsmessdaten für DB2-Workload-Ma- nagement.	290
Integration von Workload-Management-Tabellen- funktionen und Snapshot Monitor	291
Überwachen von Schwellenwertverstößen	292
Generieren von E-Mail-Benachrichtigungen für Schwellenwertverstöße	294
Erfassen von Daten für einzelne Aktivitäten	297
Importieren von Aktivitäten in Designadvisor	299
Abbrechen von Aktivitäten	300
Richtlinien für das Erfassen von Informationen zu einer Rogue-Aktivität und Untersuchen dieser Akti- vität	300
Modellieren der Workload-Management-Leistung	301
Beispiel: Erfassen von Informationen zu einer Akti- vität für spätere Analyse	302

Kapitel 5. Integration in Betriebssystem-Workload-Manager 305

Integration von AIX Workload Manager in DB2- Workload-Management	305
Integration von Linux-Workload-Management in DB2-Workload-Management	311

Kapitel 6. Lernprogramm für DB2-Workload-Management 319

Übung 1: Einführung in die grundlegende Überwa- chung mit DB2-Workload-Management-Standard- objekten	319
Übung 2: Eingrenzen der Aktivitäten mit Service- klassen und Workloads	325
Übung 3: Verwenden von Schwellenwerten zum Steuern von Rogue-Aktivitäten und Verwenden des Schwellenwertverstoßmonitors	331
Übung 4: Unterscheiden von Aktivitäten nach Akti- vitätstyp	335
Übung 5: Verwenden von Histogrammen für Ser- viceklassen	343
Übung 6: Untersuchen von Verzögerungen mit WLM-Tabellenfunktionen	352
Übung 7: Abbrechen einer laufenden Aktivität	355
Übung 8: Ermitteln der auf dem System aktiven Aktivitäten	356
Übung 9: Erfassen von ausführlichen Informatio- nen zu einer zurzeit ausgeführten Aktivität	359
Übung 10: Generieren von Protokolldaten und Be- richten	361
Übung 11: Verwenden von erweiterten Zusammen- fassungen für Serviceklassen	364

Kapitel 7. Workload-Management-Sze- narios 371

Workload-Management-Beispielanwendung	371
Szenario: Untersuchen einer workloadbezogenen Leistungsminderung des Systems.	372
Szenario: Zusammenfassen von Aktivitätsmessda- ten über untergeordnete Aktivitäten hinweg	373
Szenario: Ermitteln von Aktivitäten, deren Ausfüh- rung zu lange dauert.	376
Szenario: Abbrechen von Aktivitäten, die länger als eine Stunde in der Warteschlange stehen	379
Szenario: Erkennen von Aktivitäten mit niedrigem geschätztem Aufwand und langer Laufzeit	382
Szenario: Abbrechen aller in einer Serviceunterklas- se ausgeführten Aktivitäten.	382
Szenario: Trennen der Verbindung aller Anwen- dungen, die Aktivitäten in einer Serviceklasse zu- geordnet sind oder solche Aktivitäten ausführen	383
Szenario: Optimieren einer DB2-Workload-Manage- ment-Konfiguration, wenn Kapazitätsplanungsin- formationen verfügbar sind.	385
Szenario: Optimieren einer DB2-Workload-Manage- ment-Konfiguration, wenn keine Kapazitätspla- nungsinformationen verfügbar sind	387

Kapitel 8. Referenz 393

Prozeduren und Tabellenfunktionen	393
WLM_CANCEL_ACTIVITY - Aktivität abbre- chen	393

WLM_CAPTURE_ACTIVITY_IN_PROGRESS - Aktivitätsinformationen für Aktivitätsereignis- monitor erfassen	394	bin_id - ID des Histogrammbins (Monitorele- ment)	446
WLM_COLLECT_STATS - Workload-Management-Statistikdaten erfassen und zurücksetzen	396	bottom - Untere Grenze von Histogrammbin (Monitorelement)	447
WLM_GET_ACTIVITY_DETAILS - Detaillierte Informationen zu einer bestimmten Aktivität zu- rückgeben	398	concurrent_act_top - Maximaler Wert für gleich- zeitige Aktivitäten (Monitorelement).	447
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten von Schwellenwertwarteschlangen zurückgeben.	404	concurrent_connection_top - Maximaler Wert für gleichzeitige Verbindungen (Monitorelement).	448
WLM_GET_SERVICE_CLASS_AGENTS (Tabel- lenfunktion) - In einer Serviceklasse aktive Agenten auflisten	407	concurrent_wlo_act_top - Maximaler Wert für gleichzeitige Workloadaktivitäten (Monitorele- ment)	448
WLM_GET_SERVICE_CLASS _WORKLOAD_OCCURRENCES - Workloadvor- kommen auflisten	414	concurrent_wlo_top - Maximaler Wert für gleichzeitige Workloadvorkommen (Monitorele- ment)	449
WLM_GET_SERVICE_SUBCLASS_STATS (Tabel- lenfunktion) - Statistikdaten zu Serviceunterklas- sen zurückgeben	418	concurrentdbcoordactivities_db_threshold_id - Kennung des Datenbankschwellenwerts für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)	449
WLM_GET_SERVICE_SUPERCLASS_STATS - Statistikdaten zu Servicesuperklassen zurückge- ben.	425	concurrentdbcoordactivities_db_threshold_ _queued - Durch Datenbankschwellenwert für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange (Monitorelement)	450
WLM_GET_WORK_ACTION_SET_STATS - Stat- istikdaten zu Arbeitsaktionssets zurückgeben	426	concurrentdbcoordactivities_db_threshold_value - Wert des Datenbankschwellenwerts für gleich- zeitige Datenbankkoordinatoraktivitäten (Moni- torelement)	450
WLM_GET_WORKLOAD _OCCURRENCE _ACTIVITIES - Liste von Akti- vitäten zurückgeben	428	concurrentdbcoordactivities_db_threshold_viola- ted - Verstoß gegen den Datenbankschwellen- wert für gleichzeitige Datenbankkoordinatorak- tivitäten (Monitorelement)	450
WLM_GET_WORKLOAD_STATS (Tabellenfunk- tion) - Workloadstatistikdaten zurückgeben	433	concurrentdbcoordactivities_subclass_threshol- d_id - Kennung des Schwellenwerts für Service- unterklasse für gleichzeitige Datenbankkoordi- natoraktivitäten (Monitorelement)	451
WLM_SET_CLIENT_INFO (Prozedur) - Client- informationen festlegen	436	concurrentdbcoordactivities_subclass_threshol- _queued - Durch Schwellenwert für Service- unterklasse für gleichzeitige Datenbankkoordi- natoraktivitäten in Warteschlange (Monitorelement)	451
Monitorelemente für Workload-Management	439	concurrentdbcoordactivities_subclass_threshol- _value - Wert des Schwellenwerts für Serviceun- terklasse für gleichzeitige Datenbankkoordi- natoraktivitäten (Monitorelement)	452
act_cpu_time_top - Maximaler Wert für CPU- Zeit von Aktivitäten (Monitorelement)	439	concurrentdbcoordactivities_subclass_threshol- _violated - Verstoß gegen den Schwellenwert für Serviceunterklasse für gleichzeitige Daten- bankkoordinatoraktivitäten (Monitorelement).	452
act_exec_time - Ausführungszeit für Aktivitäten (Monitorelement)	440	concurrentdbcoordactivities_superclass_ threshold_id - Kennung des Schwellenwerts für Servicesuperklasse für gleichzeitige Daten- bankkoordinatoraktivitäten (Monitorelement).	453
act_remapped_in - Neu zugeordnete Aktivitäten in Service (Monitorelement)	440	concurrentdbcoordactivities_superclass_ threshold_queued - Durch Schwellenwert für Servicesuperklasse für gleichzeitige Daten- bankkoordinatoraktivitäten in Warteschlange (Monitorelement)	453
act_remapped_out - Neu zugeordnete Aktivitä- ten aus Service (Monitorelement).	441	concurrentdbcoordactivities_superclass_ threshold_value - Wert des Schwellenwerts für Servicesuperklasse für gleichzeitige Daten- bankkoordinatoraktivitäten (Monitorelement).	453
act_rows_read_top - Maximaler Wert für von Aktivitäten gelesene Zeilen (Monitorelement).	441		
act_throughput - Aktivitätsdurchsatz (Monitor- element)	441		
act_total - Gesamtzahl der Aktivitäten (Monitor- element)	442		
activate_timestamp - Zeitmarke für Aktivierung (Monitorelement)	443		
activity_collected - Aktivitäten erfassen (Moni- torelement).	443		
activity_id - Aktivitäts-ID (Monitorelement)	443		
activity_secondary_id - Sekundäre Aktivitäts-ID (Monitorelement)	444		
activity_type - Aktivitätstyp (Monitorelement)	445		
agg_temp_tablespace_top - Zusammengefasster maximaler Wert für Tabellenbereiche für tempo- räre Tabellen (Monitorelement)	446		
arm_correlator - Korrelator zur Messung der Anwendungsantwortzeit (Monitorelement)	446		

concurrentdbcoordactivities_superclass_threshold_violated - Verstoß gegen den Schwellenwert für Servicesuperklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)	454	cpu_velocity - CPU-Geschwindigkeit (Monitorelement)	467
concurrentdbcoordactivities_wl_was_threshold_id - Kennung des Schwellenwerts für Workload-Arbeitsaktionssets für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement).	454	db_work_action_set_id - Set-ID für Datenbankarbeitsaktionen (Monitorelement).	468
concurrentdbcoordactivities_wl_was_threshold_queued - Durch Schwellenwert für Workload-Arbeitsaktionsset für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange gestellt (Monitorelement)	455	db_work_class_id - Datenbankarbeitsklassen-ID (Monitorelement)	468
concurrentdbcoordactivities_wl_was_threshold_value - Wert des Schwellenwerts für Workload-Arbeitsaktionssets für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement).	455	destination_service_class_id - Zielserviceklassen-ID (Monitorelement)	469
concurrentdbcoordactivities_wl_was_threshold_violated - Verstoß gegen den Schwellenwert für Workload-Arbeitsaktionssets für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)	456	estimated_cpu_entitlement - Geschätztes CPU-Nutzungsrecht (Monitorelement)	469
coord_act_aborted_total - Gesamtzahl abgebrochener Koordinatoraktivitäten (Monitorelement).	456	histogram_type - Histogrammtyp (Monitorelement)	469
coord_act_completed_total - Gesamtzahl abgeschlossener Koordinatoraktivitäten (Monitorelement)	457	last_wlm_reset - Zeitpunkt des letzten Zurücksetzens (Monitorelement)	471
coord_act_est_cost_avg - Durchschnittswert für den geschätzten Aufwand für Koordinatoraktivitäten (Monitorelement)	458	num_remaps - Anzahl der Neuuzuordnungen (Monitorelement)	471
coord_act_exec_time_avg - Durchschnittswert für die Ausführungszeit von Koordinatoraktivitäten (Monitorelement)	458	num_threshold_violations - Anzahl der Schwellenwertverstöße (Monitorelement)	471
coord_act_interarrival_time_avg - Durchschnittswert für die Ankunftszeit von Koordinatoraktivitäten (Monitorelement)	459	number_in_bin - Anzahl in Bin (Monitorelement)	472
coord_act_lifetime_avg - Durchschnittswert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)	460	parent_activity_id - ID der übergeordneten Aktivität (Monitorelement)	472
coord_act_lifetime_top - Maximaler Wert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)	461	parent_uow_id - UOW-ID der übergeordneten Aktivität (Monitorelement)	473
coord_act_queue_time_avg - Durchschnittliche Warteschlangenzeit für Koordinatoraktivitäten (Monitorelement)	462	prep_time - Vorbereitungszeit (Monitorelement)	473
coord_act_rejected_total - Gesamtzahl zurückgewiesener Koordinatoraktivitäten (Monitorelement)	463	queue_assignments_total - Gesamtzahl der Warteschlangenzuordnungen (Monitorelement)	474
coord_partition_num - Nummer der Koordinatorpartition (Monitorelement)	464	queue_size_top - Maximaler Wert für Warteschlangengröße (Monitorelement)	474
cost_estimate_top - Maximaler Wert für Kostenschätzung (Monitorelement)	465	queue_time_total - Gesamtwarteschlangenzeit (Monitorelement)	475
cpu_limit - CPU-Begrenzung des WLM-Dispatchers (Monitorelement)	465	request_exec_time_avg - Durchschnittswert für die Ausführungszeit von Anforderungen (Monitorelement)	475
cpu_share_type - CPU-Anteilstyp des WLM-Dispatchers (Monitorelement)	465	routine_id - Routinen-ID (Monitorelement)	476
cpu_shares - CPU-Anteile des WLM-Dispatchers (Monitorelement)	466	rows_fetched - Abgerufene Zeilen (Monitorelement)	477
cpu_utilization - CPU-Auslastung (Monitorelement)	466	rows_modified - Modifizierte Zeilen (Monitorelement)	478
		rows_returned - Zurückgegebene Zeilen (Monitorelement)	479
		rows_returned_top - Maximaler Wert für zurückgegebene Zeilen (Monitorelement)	481
		sc_work_action_set_id - Set-ID für Serviceklassenarbeitsaktionen (Monitorelement)	482
		sc_work_class_id - Arbeitsklassen-ID für Serviceklassen (Monitorelement)	482
		section_env - Abschnittsumgebung (Monitorelement)	483
		service_class_id - Serviceklassen-ID (Monitorelement)	483
		service_subclass_name - Name der Serviceunterklasse (Monitorelement)	484
		service_superclass_name - Name der Servicesuperklasse (Monitorelement).	486
		source_service_class_id - Quellenserviceklassen-ID (Monitorelement)	487
		statistics_timestamp - Zeitmarke für Statistikdatensatz (Monitorelement)	487
		stmt_invocation_id - Aufruf-ID der Anweisung (Monitorelement)	488

temp_tablespace_top - Maximaler Wert für Tabellenbereiche für temporäre Tabellen (Monitorelement)	488	wlm_queue_time_total - Gesamtwartezeit für Workload-Manager-Warteschlange (Monitorelement)	510
thresh_violations - Anzahl der Schwellenwertverstöße (Monitorelement)	489	wlo_completed_total - Gesamtzahl abgeschlossener Workloadvorkommen (Monitorelement)	511
threshold_action - Schwellenwertaktion (Monitorelement)	491	work_action_set_id - ID des Arbeitsaktionssets (Monitorelement)	512
threshold_domain - Schwellenwertdomäne (Monitorelement)	491	work_action_set_name - Name des Arbeitsaktionssets (Monitorelement)	512
threshold_maxvalue - Maximaler Wert für Schwellenwert (Monitorelement)	492	work_class_id - Arbeitsklassen-ID (Monitorelement)	513
threshold_name - Name des Schwellenwerts (Monitorelement)	492	work_class_name - Name der Arbeitsklasse (Monitorelement)	513
threshold_predicate - Vergleichselement für Schwellenwert (Monitorelement)	493	workload_id - Workload-ID (Monitorelement)	514
threshold_queuesize - Größe der Schwellenwertwarteschlange (Monitorelement)	494	workload_name - Name der Workload (Monitorelement)	515
thresholdid - Schwellenwert-ID (Monitorelement)	494	workload_occurrence_id - ID des Workloadvorkommens (Monitorelement)	516
time_completed - Abschlusszeitpunkt (Monitorelement)	495	workload_occurrence_state - Status des Workloadvorkommens (Monitorelement)	517
time_created - Erstellungszeitpunkt (Monitorelement)	495	Befehle	518
time_of_violation - Zeitpunkt des Verstoßes (Monitorelement)	495	SET WORKLOAD	518
time_started - Startzeitpunkt (Monitorelement)	495	Konfigurationsparameter	519
bottom - Obere Grenze von Histogrammbin (Monitorelement)	496	wlm_collect_int - Workload-Management-Erfassungsintervall	519
total_disp_run_queue_time - Gesamtwartezeit in Ausführungswarteschlange des Dispatchers (Monitorelement)	496	wlm_dispatcher - Workload-Management-Dispatcher	520
uow_completed_total - Gesamtzahl beendeter UOWs (Monitorelement)	499	wlm_disp_concur - WLM-Dispatcher-Thread-Parallelität	521
uow_comp_status - Fertigstellungsstatus der UOW	499	wlm_disp_cpu_shares - CPU-Anteil des WLM-Dispatchers	522
uow_elapsed_time - Abgelaufene Zeit der letzten UOW	500	wlm_disp_min_util - Minimale CPU-Auslastung des Workload-Manager-Dispatchers	522
uow_id - UOW-ID (Monitorelement)	501	Katalogsichten	523
uow_lifetime_avg - Durchschnittliche UOW-Lebensdauer (Monitorelement)	501	SYSCAT.HISTOGRAMTEMPLATEBINS	523
uow_lock_wait_time - Gesamtwartezeit der UOW auf Sperrern (Monitorelement)	502	SYSCAT.HISTOGRAMTEMPLATES	524
uow_log_space_used - Verwendeter UOW-Protokollspeicher (Monitorelement)	503	SYSCAT.HISTOGRAMTEMPLATEUSE	524
uow_start_time - Zeitmarke beim Start der UOW (Monitorelement)	503	SYSCAT.SERVICECLASSES	525
uow_status - UOW-Status	504	SYSCAT.THRESHOLDS	528
uow_stop_time - Zeitmarke beim Stopp der UOW (Monitorelement)	505	SYSCAT.WORKACTIONS	531
uow_throughput - UOW-Durchsatz (Monitorelement)	505	SYSCAT.WORKACTIONSETS	533
uow_total_time_top - Maximaler Wert für UOW-Gesamtzeit (Monitorelement)	506	SYSCAT.WORKCLASSES	534
wl_work_action_set_id - Arbeitsaktionsset-ID der Auslastung (Monitorelement)	507	SYSCAT.WORKCLASSESETS	534
wl_work_class_id - Auslastungsarbeitsklassen-ID (Monitorelement)	507	SYSCAT.WORKLOADAUTH	535
wlm_queue_assignments_total - Gesamtzahl der Warteschlangenzuweisungen durch den Workload-Manager (Monitorelement)	508	SYSCAT.WORKLOADCONNATTR	535
		SYSCAT.WORKLOADS	536

Anhang A. Allgemeine Namenskonventionen 541

Anhang B. Rollen. 543

Anhang C. Gesicherter Kontext und gesicherte Verbindungen 545

Anhang D. Übersicht über technische Informationen zu DB2. 549

Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format 550

Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor	552
Zugriff auf verschiedene Versionen des DB2 Information Center	552
Aktualisieren des auf Ihrem Computer oder Intranet-Server installierten DB2 Information Center	553
Manuelles Aktualisieren des auf Ihrem Computer oder Intranet-Server installierten DB2 Information Center.	554

DB2-Lernprogramme	557
Informationen zur Fehlerbehebung in DB2	557
Bedingungen	558

Anhang E. Bemerkungen 559

Index 563

Zu diesem Handbuch

Das vorliegende Handbuch enthält Informationen zu DB2-Workload-Management-Features und zur Funktionalität, die es Ihnen ermöglichen, eine stabile, vorhersehbare, auf Ihre Geschäftsziele abgestimmte Ausführungsumgebung zu erhalten. Bei der Verwendung von DB2-Workload-Management werden sowohl Anforderungen als auch Ressourcen verwaltet. Das vorliegende Handbuch enthält außerdem Informationen zur Überwachung und Durchführung einer Fehlersuche für die Workload auf Ihrem Datenserver.

Kapitel 1. Einführung in DB2-Workload-Management-Konzepte

Ein gutes Workload-Management-System trägt dazu bei, die gewünschten Ziele in der Umgebung, in der Arbeit anfällt, effizient zu erreichen. Sie können Beispiele für die Notwendigkeit eines guten Workload-Management-Systems überall in Ihrer Umgebung feststellen.

Sehen Sie sich z. B. ein Lebensmittelgeschäft an. Unterschiedliche Aktivitäten müssen berücksichtigt werden: Bedienung von Kunden, Auffüllen von Waren, Pflege des Lagerbestands usw. Außerdem müssen einige einfache Ziele festgelegt werden. Geschäftseigner wollen sowohl die Anzahl der Kunden im Laden als auch die Beträge für die Kundeneinkäufe maximieren und beide Ziele auf eine Weise erreichen, dass die Kunden zufrieden sind und gerne wiederkommen. Außerdem müssen Geschäftseigner sicherstellen, dass sie einen ausreichenden Warenbestand für ihre Kunden haben (wobei der Bestand aber nicht zu groß sein sollte, damit nicht zu viele Waren verderben). Geschäftseigner wollen auch verfolgen, was ihre Kunden kaufen, und diese Informationen zur Erstellung von Werbung verwenden, mit der die Kundschaft zum Wiederkommen bewegt werden soll. Überwachungsmechanismen verfolgen den Lagerbestand und senden Benachrichtigungen bei niedrigen Beständen. Zur Erkennung von Ladendiebstahl sind entsprechende Sicherungsvorrichtungen vorhanden. Spezielle Schnellkassen werden eingerichtet, so dass Käufer, die nur wenige Artikel kaufen wollen, dies tun können, ohne hinter Kunden mit vielen Waren warten zu müssen. Wenn alle diese Ziele erfüllt sind und alle Betriebsprozesse gut funktionieren, sind die Kunden zufrieden und kommen gern wieder, statt in ein anderes Geschäft zu gehen. Diese Ziele und Betriebsprozesse sind alles Aspekte von Workload-Management.

In einer Datenserverumgebung ist die Notwendigkeit für eine effektive Arbeitsverwaltung noch wichtiger, insbesondere jetzt, da Datenserver wie nie zuvor ausgelastet sind. Elektronische Kassensysteme generieren Tausende von Dateneinfügungen, es werden konstant Berichte generiert, um zu ermitteln, ob die Vertriebsziele erreicht werden, Stapelanwendungen werden ausgeführt, um erfasste Daten zu laden, und Verwaltungsaufgaben wie Sicherungen und Reorganisationen werden ausgeführt, um die Daten zu schützen und den Server optimal arbeiten zu lassen. Alle diese Operationen verwenden dasselbe Datenbanksystem und konkurrieren um dieselben Ressourcen.

Ein effizientes Workload-Management-System ist daher entscheidend, um die Ziele für den Betrieb eines Datenservers bestmöglich zu erreichen.

Workload-Management-Domänen

Das Workload-Management umfasst drei klar definierte Phasen: die Ermittlung der Arbeit vor dem Eintritt in den Datenserver, die Verwaltung der Arbeit während der Ausführung sowie die Überwachung, mit der sichergestellt wird, dass der Datenserver effizient genutzt wird.

Für ein erfolgreiches Workload-Management mit dem DB2-Workload-Manager muss eine Reihe von Aspekten berücksichtigt werden, beginnend mit der Kenntnis Ihrer Ziele. Im Beispiel des in Kapitel 1, „Einführung in DB2-Workload-Management-Konzepte“ beschriebenen Lebensmittelgeschäfts wären mögliche Ziele, die

Beträge für die Kundeneinkäufe zu maximieren, Ladendiebstahl zu minimieren und sicherzustellen, dass Kunden den Laden so zufrieden verlassen, dass sie wiederkommen.

Auch in einer Datenservertumgebung müssen Sie Ziele definieren. Manchmal sind die Ziele eindeutig, insbesondere wenn sie durch Service-Level-Agreements begründet sind. Beispielsweise dürfen Abfragen von einer bestimmten Anwendung höchstens 10 % der gesamten Prozessorressourcen verbrauchen. Ziele können auch an eine bestimmte Tageszeit gebunden sein. Beispielsweise muss ein über Nacht laufendes Stapeldienstprogramm bis 8 Uhr morgens das Laden von Daten abgeschlossen haben, damit die täglichen Verkaufsberichte rechtzeitig fertig sind. In anderen Situationen sind die Ziele möglicherweise nur schwer quantitativ zu bestimmen. Ein Ziel könnte sein, dass die Datenbankbenutzer zufrieden sind und dass keine anormale Datenbankaktivität sie in ihrer täglichen Arbeit behindert. Unabhängig davon, ob die Ziele quantitativ bestimmt werden können oder nicht, ist die Kenntnis der Ziele entscheidend, wenn die folgenden Phasen des Workload-Managements betrachtet werden:

Identifikation

Wenn Sie ein Ziel für eine bestimmte Art von Arbeit erreichen wollen, müssen Sie zuerst die Details zur Arbeit angeben können. Im Lebensmittelgeschäft können Sie Käuferinformationen über Kredit- und Kundenkarten oder nicht bezahlte Artikel über ein aktives Sicherheitsetikett am Artikel ermitteln. Beim Datenservert müssen Sie entscheiden, wie Sie die Arbeit angeben wollen, die im System eintrifft. Sie können den Namen der Anwendung, die die Arbeit übergibt, oder die Berechtigungs-ID, mit der die Arbeit übergeben wird, oder eine Kombination von Elementen verwenden, die eine Form von Identifikation bereitstellt.

Verwaltung

Die Verwaltungsphase umfasst Mechanismen, mit denen Sie Ihr Ziel immer besser erfüllen können, sowie Maßnahmen, die zu ergreifen sind, wenn das Ziel nicht erreicht wird. Ein Beispiel für einen Mechanismus ist die Handhabung von Preisprüfungen an Schnellkassen. Schnellkassen sollten zu einem höheren Durchsatz und zufriedeneren Kunden führen. Wenn jedoch eine Milchpackung falsch ausgezeichnet ist und eine Preisprüfung erforderlich wird, könnte das den Durchsatz an der Schnellkasse verringern. Eine Lösung für dieses Problem ist eine schnelle Preisprüfung verbunden mit dem Öffnen einer anderen Kasse (wenn möglich) und dem Versuch das Preisproblem so zu beheben, dass es nicht wieder auftritt. Beim Datenservert stellen Sie eventuell fest, dass die Gesamtleistung leidet, wenn einige schlecht geschriebene SQL-Anweisungen ausgeführt werden, dass in Stoßzeiten das Volumen sich stark vergrößert oder dass eine zu große Konkurrenz um dieselben Ressourcen durch unterschiedliche Anwendungen auftritt. Die Verwaltungsphase umfasst Mechanismen für die Zuordnung von Ressourcen, damit Sie Ihre Ziele erreichen können, sowie Maßnahmen für den Fall, in dem ein Ziel nicht erreicht wird. Zu diesen Workload-Management-Mechanismen, die CPU-Ressourcen indirekt und direkt steuern, gehören die folgenden Funktionen:

1. Schwellenwerte für gemeinsamen Zugriff, die mit einem für die Workload definierten Arbeitsaktionsset angewendet werden, um den gemeinsamen Zugriff von eingehender Arbeit zu steuern.
2. Möglichkeit zum Verschieben von Arbeit aus einer Serviceklasse in eine andere (zurzeit nur wirksam, wenn ein Workload-Management-Dispatcher für diese Serviceklassen verwendet wird)

3. Workload-Management-Dispatcher, um Workloads, die Serviceklassen zugewiesen sind, CPU-Ressourcen zuzuordnen, sodass eine differenziertere Steuerung der CPU-Ressourcen erzielt wird, wenn die ersten beiden Workload-Management-Mechanismen nicht adäquat sind.

Überwachung

Die Überwachung ist aus den folgenden Gründen wichtig. Um zu ermitteln, ob Sie ein Ziel erreichen, müssen Sie einen Mechanismus haben, um den Fortschritt zu diesem Ziel hin zu verfolgen. Außerdem hilft die Überwachung Ihnen bei der Ermittlung der Probleme, die das Erreichen des Ziels verhindern können. In einem Laden kann der Filialleiter den Kundenfluss beobachten, sich automatisch Probleme wie Ladendiebstahl oder einen extrem niedrigen Bestand eines Artikels melden lassen oder eine Analyse vergangener Kaufmuster durchführen, um die optimale Produktplatzierung im Laden zu ermitteln. Bei einem Datenserver gibt es oft explizite Ziele für Antwortzeiten von Datenbankaktivitäten und es ist wichtig, eine Möglichkeit zu haben, diese Messgrößen zu ermitteln und auf Trends hin zu untersuchen.

Die folgende Abbildung stellt die Phasen des Workload-Managements dar:

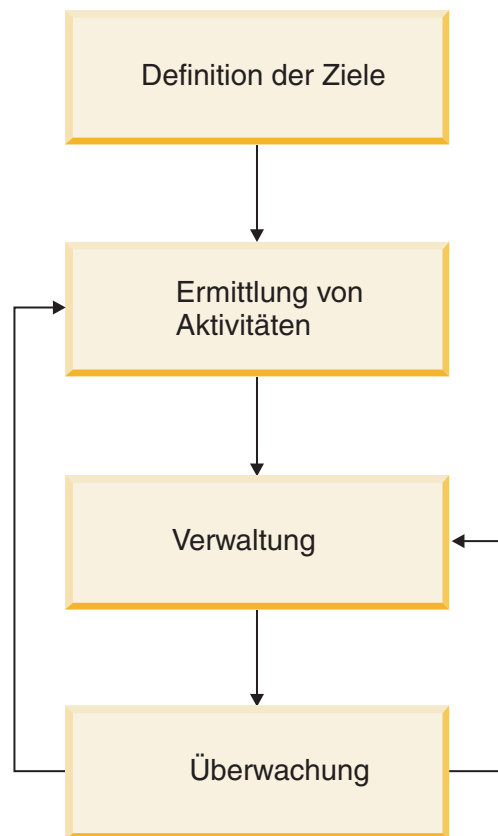


Abbildung 1. Workload-Management-Phasen

Berechtigung WLADM (Workload-Management-Administrator)

Sie benötigen die Berechtigung WLADM (Workload-Management-Administrator), um Workloadobjekte für eine bestimmte Datenbank verwalten zu können. Mit dieser Berechtigung können Sie DB2-Workload-Management-Objekte erstellen, ändern, löschen und kommentieren sowie Zugriff auf diese Objekte erteilen oder widerrufen.

Workload-Management-Objekte sind Systemobjekte ähnlich den Pufferpools und Tabellenbereichen. Deshalb ist ihnen kein Eigentümer zugeordnet.

Der Sicherheitsadministrator, d. h. ein Benutzer mit der Berechtigung SECADM, oder ein Benutzer mit der Berechtigung ACCESSCTRL kann einem Benutzer, einer Gruppe oder einer Rolle die Berechtigung WLMADM erteilen.

Die Berechtigung WLMADM erlaubt die Ausführung der folgenden Tasks:

- Absetzen von CREATE-, ALTER-, COMMENT ON- und DROP-Anweisungen für die folgenden DB2-Workload-Management-Objekte:
 - Histogrammschablonen
 - Serviceklassen
 - Schwellenwerte
 - Arbeitsaktionssets
 - Arbeitsklassensets
 - Workloads
- Absetzen von GRANT- und REVOKE-Anweisungen für Workloadberechtigungen

Die Datenbankadministratorberechtigung (DBADM) impliziert die Berechtigung WLMADM.

Häufig gestellte Fragen zum DB2-Workload-Management

In diesem Abschnitt finden Sie Antworten auf häufig gestellte Fragen zum DB2-Workload-Management.

Allgemeines

- Auf welchen DB2-Plattformen kann ich das DB2-Workload-Management verwenden?
- Ich arbeite nicht unter AIX. Bedeutet dies, dass ich keine Kontrolle über Prozessorressourcen oder E/A-Aktivitäten habe?
- Query Patroller wird nicht weiterverwendet und DB2 Governor wird nicht weiter unterstützt. Wie migriere ich jetzt auf DB2-Workload-Manager?
- Gibt es eine Möglichkeit, wie WebSphere Application Server die von der DB2-Workload verwendeten Clientinformationsfelder übergeben kann?
- Warum wird meine Arbeit nicht der korrekten Workload zugeordnet?
- Warum beeinflusst der DB2-Workload-Manager die Befehle REORGCHK, IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors?
- Gibt es eine Möglichkeit, die Serviceklasse, der eine Aktivität zugeordnet ist, während der Ausführung zu ändern?
- Ein großer Teil meiner Stapelverarbeitungen erfolgt über Befehlszeilenprozessorscripts unter derselben ID. Wie kann ich diese eindeutig angeben, sodass ich sie unterschiedlich verwalten kann?

- Wann sollte ich die Klausel COLLECT AGGREGATE ACTIVITY DATA und wann die Klausel COLLECT ACTIVITY DATA verwenden?
- Wie funktioniert DB2-Workload-Management mit dem neuen AIX-WPAR-Feature?
- Welche Beziehung besteht zwischen der Registry-Variablen DB2_OPT_MAX_TEMP_SIZE und einem DB2-Schwellenwert, der auf SQLTEMPSPACE basiert?

Lizenzierung

- Welche Lizenzvoraussetzungen gelten für DB2-Workload-Manager?

Überwachung

- Welche Informationen stellen die verschiedenen Ereignismonitore bereit, die zum Workload-Management gehören?

Betriebssystem-Workload-Management (AIX-WLM und Linux-WLM)

- Warum sollte ich AIX-WLM oder Linux-WLM verwenden?
- Ich arbeite nicht unter AIX. Bedeutet dies, dass ich keine Kontrolle über Prozessorressourcen oder E/A-Aktivitäten habe?
- Kann ich mit AIX-WLM E/A-Aktivitäten verwalten?
- Kann ich mit AIX-WLM die Speicherbelegung verwalten?
- Wie funktioniert DB2-WLM mit dem neuen AIX-WPAR-Feature?

Plattformen

- Auf welchen DB2-Plattformen kann ich den DB2-Workload-Manager verwenden?
- Ich arbeite nicht unter AIX. Bedeutet dies, dass ich keine Kontrolle über Prozessorressourcen oder E/A-Aktivitäten habe?
- Wie funktioniert DB2-Workload-Management mit dem neuen AIX-WPAR-Feature?
- Warum sollte ich AIX-WLM oder Linux-WLM verwenden?

Query Patroller und Governor

- Wie wirkt sich diese neue Funktionalität auf Query Patroller und DB2 Governor aus?
- Query Patroller und DB2 Governor werden nun nicht weiter unterstützt. Wie migriere ich jetzt auf DB2-Workload-Manager?

Schwellenwerte

- Kann ich mehrere CONCURRENTDBCOORDACTIVITIES-Schwellenwerte für gemeinsamen Zugriff für dieselbe Arbeit erstellen?
- Wie stelle ich fest, welche Aktivitäten von einem Workload-Management-Schwellenwert in die Warteschlange gestellt werden, sowie deren Reihenfolge?

Workload-Management-Dispatcher

- Muss ich den Workload-Management-Dispatcher verwenden?
- Wie ändert sich das Verhalten, wenn ich den Workload-Management-Dispatcher aktiviere?
- Sind Schwellenwerte für gemeinsamen Zugriff wie CONCURRENTDBCOORDACTIVITIES durch die Einführung des Workload-Management-Dispatchers nicht mehr erforderlich oder nützlich?

Auf welchen DB2-Plattformen kann ich das DB2-Workload-Management verwenden?

Das DB2-Workload-Management ist auf allen Plattformen verfügbar, die von DB2 9.5 für Linux, UNIX und Windows oder höher unterstützt werden. Die optionale enge Integration, die zwischen dem DB2-Workload-Management auf Serviceklassenebene und Workload-Management-Funktionen des Betriebssystems angeboten wird, ist auf AIX-Plattformen und auf allen Linux-Plattformen verfügbar, die auf dem Kernel der Version 2.6.26 oder höher basieren.

Muss ich den Workload-Management-Dispatcher verwenden?

Die meisten Workload-Management-Konfigurationen beginnen mit Schwellenwerten für gemeinsamen Zugriff, die sich auf den Verbrauch aller Ressourcen auswirken, indem gesteuert wird, wie viel Arbeit zu einem beliebigen Zeitpunkt ausgeführt werden kann. In einigen Situationen ist ein Schwellenwert für gemeinsamen Zugriff jedoch nicht in der Lage, die Gesamtmenge der verbrauchten Verarbeitungsressourcen effektiv einzuschränken, wodurch Arbeit mit hoher Priorität beeinträchtigt wird. Dies gilt z. B. in einem Szenario, in dem komplexe Arbeit auf eine aktive Abfrage eingeschränkt ist, die allerdings trotzdem so viele Ressourcen verbraucht, dass Arbeit von höherer Priorität unterbrochen wird. In solchen Fällen wird der Workload-Management-Dispatcher verwendet, um die CPU-Belegung explizit zu steuern und die Arbeit mit höherer Priorität zu schützen.

Sie können den Workload-Management-Dispatcher in allen oder einigen der folgenden Situationen verwenden:

- Sie wollen den Anteil der CPU-Ressourcen unter mehreren Benutzern oder Anwendungen verwalten und Sie verwenden ein Betriebssystem, das keinen Betriebssystem-Workload-Manager enthält, der über das Feld `outbound_correlator` in jeder Serviceklasse in das DB2-Workload-Management integriert werden könnte.
- Sie wollen den Anteil der CPU-Ressourcen unter mehreren Benutzern oder Anwendungen verwalten und Sie verfügen für das Betriebssystem über keine Rootberechtigung.
- Sie wollen den Anteil der CPU-Ressourcen unter mehreren Benutzern oder Anwendungen in einer Umgebung mit mehreren Mitgliedern systemübergreifend verwalten und diese Verwaltung über den Betriebssystem-Workload-Manager auf jedem System erfordert zu viel Verwaltung.
- Sie wollen den Anteil der CPU-Ressourcen unter mehreren Benutzern oder Anwendungen über feste Anteile verwalten, um bestimmte Serviceklassen zu begrenzen, selbst wenn die CPU nicht ausgelastet ist, und diese Funktion ist in Ihrem Betriebssystem-Workload-Manager nicht verfügbar oder erzeugt nicht das gewünschte Ergebnis.

Wie wirkt sich diese neue Funktionalität auf Query Patroller und DB2 Governor aus?

Der DB2-Workload-Manager führt eine unabhängige Workload-Management-Methode ein. Er benötigt und interagiert in keiner Weise mit Query Patroller oder DB2 Governor. Query Patroller wird seit dem Version 10.1-Release nicht weiterverwendet. DB2 Governor ist zwar funktionsfähig, wird jedoch in DB2 Version 9.7 nicht weiter unterstützt, weil es in der DB2-Workload-Management-Strategie nicht mehr eine zentrale Rolle spielt. In zukünftige Releases von DB2 Governor wird nicht investiert.

Bei der Erstinstallation von DB2 9.5 oder höher wird die Standardbenutzerserviceklasse automatisch definiert und die gesamte eingehende Arbeit wird zur Ausführung an diese Klasse gesendet. DB2 Governor kann zwar Agenten in jeder beliebigen Serviceklasse überwachen, darf die Agentenpriorität jedoch nur für Agenten in der Standardbenutzerserviceklasse anpassen.

Ich arbeite nicht unter AIX. Bedeutet dies, dass ich keine Kontrolle über Prozessorressourcen oder E/A-Aktivitäten habe?

Benutzer auf allen Plattformen haben dieselbe Möglichkeit, die Prozessorressourcen und die E/A-Aktivität zwischen Serviceklassen über SQL zu steuern, z. B. über die Anweisungen CREATE SERVICE CLASS und ALTER SERVICE CLASS.

Zur Steuerung der CPU-Belastung können Sie bei Aktivierung des Workload-Management-Dispatchers mit dem Attribut für den CPU-Grenzwert der DB2-Serviceklasse das CPU-Ressourcenvolumen begrenzen, das eine Serviceklasse verbrauchen kann. Wenn der Konfigurationsparameter des Datenbankmanagers für die CPU-Anteile des Workload-Management-Dispatchers (**wlm_disp_cpu_shares**) ebenfalls aktiviert ist, können Sie mit dem Attribut für CPU-Anteile der DB2-Serviceklasse den Anteil der CPU-Ressourcen angeben, den eine Serviceklasse relativ zur CPU-Belegung der anderen Serviceklassen verbrauchen kann. Unter AIX und auf einigen Linux-Plattformen können Sie als Ergänzung (oder Ersatz) dieser Methoden die Vorteile der Workload-Management-Funktionen nutzen, die von diesen Betriebssystemen angeboten werden, um die CPU-Belegung zu steuern.

Bei der E/A-Aktivität können Benutzer auf allen Plattformen das Attribut für den Pufferpool oder die Vorablesezugriffspriorität einer DB2-Serviceklasse auf HIGH, MEDIUM oder LOW setzen. Alle Serviceklassen werden standardmäßig mit einer mittleren Priorität ausgeführt.

Kann ich mit AIX- oder Linux-WLM bzw. dem DB2-Workload-Management-Dispatcher E/A-Aktivitäten verwalten?

Zurzeit unterstützt weder AIX-WLM, noch Linux-WLM direkte Steuerungsmöglichkeiten der E/A-Aktivität auf Threadebene. Es ist allerdings möglich, die E/A-Aktivität indirekt über Schwellenwerte für gemeinsamen Zugriff zu steuern oder die CPU-Ressourcen mithilfe des DB2-Workload-Management-Dispatcher, des AIX-WLM oder des Linux-WLM zu manipulieren. Je mehr CPU-Ressourcen für einen ausgeführten Thread verfügbar sind, desto seltener fordert dieser Thread E/A-Ressourcen an.

Sie können das Verhalten des Pufferpools beeinflussen, indem Sie das Attribut BUFFERPOOL PRIORITY einer beliebigen DB2-Serviceklasse verwenden. Außerdem können Sie die DB2-Vorablesezugriffs-E/A-Aktivität mit dem Attribut PREFETCH PRIORITY jeder DB2-Serviceklasse steuern.

Kann ich mit AIX-WLM oder Linux-WLM die Speicherbelegung verwalten?

Der DB2-Datenserver nutzt vorwiegend gemeinsam genutzten Speicher, auf den mehrere Agenten von unterschiedlichen Serviceklassen zugreifen. Deshalb ist es nicht möglich, mit AIX-WLM oder Linux-WLM die Speicherzuordnung zwischen unterschiedlichen Serviceklassen aufzuteilen.

Der Speicher (z. B. Sortierspeicher), der während der Ausführung einer SQL-Anweisung verbraucht wird, kann indirekt durch die Verwendung von Schwellenwer-

ten für gemeinsamen Zugriff beeinflusst werden, weil der Verbrauch erst beginnt, wenn die Anweisung ausgeführt werden darf. Im Gegensatz zur E/A-Aktivität hat das Einschränken der CPU-Belegung jedoch keine Auswirkungen auf die verbrauchte Speichermenge. Das Einschränken der CPU-Belegung kann die Speichersituation sogar verschlimmern, weil Anfragen langsamer ausgeführt werden und den zugeordneten Speicher länger belegen.

Gibt es eine Möglichkeit, wie WebSphere Application Server die von der DB2-Workload verwendeten Clientinformationsfelder übergeben kann?

WebSphere Application Server Version 6.0 und Version 6.1 können die CLIENT INFO-Felder festlegen oder an den DB2-Datenserver übergeben, entweder explizit von Ihren Anwendungen (siehe Clientinformationen an eine Datenbank übergeben) oder implizit durch WebSphere Application Server (siehe Clientinformationen implizit definieren).

Kann ich mehrere CONCURRENTDBCOORDACTIVITIES-Schwellenwerte für gemeinsamen Zugriff für dieselbe Arbeit erstellen?

Sie können mehrere CONCURRENTDBCOORDACTIVITIES-Schwellenwerte für gemeinsamen Zugriff erstellen, die auf dieselbe Gruppe von Aktivitäten angewendet werden, indem Sie sie auf der Ebene der Datenbank, in der Serviceklasse, in der die Arbeit ausgeführt wird, oder in einem Arbeitsaktionsset definieren, das auf Datenbank- oder Workloadebene angewendet wird. Berücksichtigen Sie dabei, dass jeder neue Schwellenwert für gemeinsamen Zugriff, der auf eine Aktivität angewendet wird, zusätzlichen Systemaufwand für seine Einhaltung erfordert. Vergewissern Sie sich, dass Sie tatsächlich mehr als eine Schwellenwertebene für gemeinsamen Zugriff benötigen.

Warum wird meine Arbeit nicht der korrekten Workload zugeordnet?

Es gibt eine Reihe von Gründen, warum eine Verbindung möglicherweise nicht der gewünschten Workload zugeordnet wird. Die häufigsten Ursachen sind, dass versäumt wurde, das Zugriffsrecht USAGE für die Workload zu erteilen, eine falsche Schreibweise der Verbindungsattribute (die Groß-/Kleinschreibung unterscheiden) oder das Vorhandensein einer übereinstimmenden Workloaddefinition, die sich in der Bewertungsreihenfolge weiter vorn befindet.

Bevor eine Verbindung einer Workload zugeordnet werden kann, müssen die Verbindungsattribute denen der Workloaddefinition entsprechen, und die Sitzungsberechtigungs-ID muss das Zugriffsrecht USAGE für die Workload haben. Ein gängiger Fehler ist auch, dass die Workload erstellt wird, das Zugriffsrecht USAGE für die Workload jedoch den Benutzern nicht erteilt wird (siehe „Anweisung GRANT (Workloadzugriffsrechte)“ in *SQL Reference*). Nur Benutzer mit der Berechtigung ACCESSCTRL, SECADM oder WLMADM können anderen Benutzern das Zugriffsrecht USAGE für Workloads erteilen. Benutzer mit der Berechtigung ACCESSCTRL, DATAACCESS, DBADM, SECADM oder WLMADM haben implizite Nutzungsberechtigung für alle Workloads.

Verbindungsattribute für Workloads unterscheiden Groß-/Kleinschreibung. Wenn die Systembenutzer-ID z. B. in Großbuchstaben geschrieben ist, müssen Sie das Verbindungsattribut für SYSTEM_USER ebenfalls in Großbuchstaben schreiben.

Damit Sie herausfinden können, warum eine Verbindung nicht der erwarteten Workload zugeordnet wird, sollten Sie einige Informationen sammeln. Welcher Workload wird die Arbeit zugeordnet? Steht diese Workload in der Bewertungsreihenfolge der Workloaddefinitionen vor oder nach der Workload, deren Verwendung Sie erwartet haben? (Tipp: Versuchen Sie, die Workloaddefinitionen auszuwählen, die in aufsteigender Reihenfolge nach dem Wert der Spalte EVALUATIONORDER in SYSCAT.WORKLOADS geordnet sind).

Wenn Sie nicht wissen, wie die Verbindungsattribute für die Zielverbindung lauten, können Sie die Werte für die Verbindung auf mehrere Arten herausfinden:

- Setzen Sie auf dem System eine Abfrage mit der Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES ab, während die Verbindung aktiv ist.
- Öffnen Sie einen Cursor auf einer Verbindung und verwenden Sie die gespeicherte Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS für diesen Cursor, um die Aktivitätsinformationen im Aktivitätsereignismonitor erfassen zu lassen. (Tipp: Vergessen Sie nicht, den Aktivitätsereignismonitor zu erstellen und zu aktivieren.)
- Aktivieren Sie die Datenerfassung detaillierter Aktivitätsinformationen für die von der Verbindung verwendete Workload, setzen Sie eine Anweisung ab, um die Aktivitätsinformationen zu erfassen, und inaktivieren Sie dann die Datenerfassung.

Warum beeinflusst der DB2-Workload-Manager die Befehle REORGCHK, IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors?

Diese Befehlszeilenprozessor-Befehle sind von DB2-Workload-Management-Schwellenwerten betroffen, da die Datenbanksteuerkomponente nicht zwischen Systemanfragen, die von diesen Dienstprogrammen ausgehen, und anderen Anfragen unterscheiden kann, die direkt von Benutzern im interaktiven Front-End des Befehlszeilenprozessors eingeleitet werden.

Gibt es eine Möglichkeit, die Zuordnung einer Aktivität zu einer Serviceklasse während der Ausführung zu ändern?

Ja. Sie können die Serviceunterklasse, in der eine Aktivität ausgeführt wird, in eine andere Serviceunterklasse in derselben übergeordneten Servicesuperklasse ändern, indem Sie einen Schwellenwert CPUTIMEINSC, DATATAGINSC oder SQLROWSINSC mit der Aktion REMAP ACTIVITY in der ursprünglichen Serviceunterklasse definieren. Zu Anfang ordnet das DB2-Workload-Management eine Aktivität einer Serviceklasse basierend auf der relevanten Workloaddefinition für die Verbindung zu, ändert sie nach Bedarf, wenn ein Arbeitsaktionsset für diese Serviceklasse vorhanden ist, und richtet dann den DB2-Agenten so ein, dass er in der zugeordneten Serviceklasse ausgeführt wird. Wenn eine Aktivität gegen einen Schwellenwert verstößt, für den eine Aktion REMAP ACTIVITY definiert ist, ordnet der Agent sich selbst der angegebenen Zielserviceunterklasse (unter derselben Superklasse) zu, sobald der Schwellenwertverstoß erkannt wurde. Die Aktivität wird in der neuen Serviceunterklasse weiter ausgeführt.

Ein großer Teil meiner Stapelverarbeitungen erfolgt über Befehlszeilenprozessorscripts unter derselben ID. Wie kann ich diese eindeutig bezeichnen, sodass ich sie unterschiedlich verwalten kann?

Sie haben folgende Möglichkeiten:

Es wurde eine Erweiterung am Befehlszeilenprozessor vorgenommen, sodass der Name der Clientanwendung automatisch auf den Scriptdateinamen des Befehlszeilenprozessors gesetzt wird, wobei dem Namen das Präfix **CLP** vorangestellt wird (der Wert dieses Felds auf dem Server wird im Sonderregister CURRENT CLIENT_APPLNAME angezeigt). Wenn der Dateiname des Befehlszeilenprozessorscripts z. B. **batch.db2** lautet, wird der Wert des Sonderregisters CURRENT CLIENT_APPLNAME bei der Ausführung des Scripts vom Befehlszeilenprozessor auf **CLP batch.db2** gesetzt. Mit diesem Feature können unterschiedliche Befehlszeilenprozessorscripts basierend auf dem Clientanwendungsnamen unterschiedlichen Workloads zugeordnet werden.

Wenn Sie z. B. eine Workload für die Befehlszeilenprozessordatei **batch1.db2** erstellen wollen, können Sie die folgende DDL-Anweisung absetzen:

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

Wenn Sie eine Workload für die Befehlszeilenprozessordatei **batch2.db2** erstellen wollen, können Sie die folgende DDL-Anweisung absetzen:

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

Da diese beiden Stapeldateien unterschiedlichen Workloads zugeordnet sind, können sie unterschiedlichen Serviceklassen zugeordnet und unterschiedlich verwaltet werden.

Eine weitere Möglichkeit ist die neue gespeicherte Prozedur WLM_SET_CLIENT_INFO, mit der Sie die Werte aller Clientinformationsfelder auf dem Server mit einer einfachen SQL-Anweisung CALL festlegen können. Durch das Einfügen einer Anweisung CALL in Ihre vorhandenen Befehlszeilenprozessorscripts können Sie diese mithilfe dieser Felder eindeutig angeben und sie unterschiedlichen Workloaddefinitionen zuordnen.

Weitere Informationen finden Sie in „WLM_SET_CLIENT_INFO (Prozedur)“ in *Administrative Routines and Views*.

Wann sollte ich die Klausel COLLECT AGGREGATE ACTIVITY DATA und wann die Klausel COLLECT ACTIVITY DATA verwenden?

Die Antwort hängt davon ab, warum die Überwachung gewünscht wird und was mit den Informationen geschehen soll.

Zusammengefasste Aktivitätsinformationen beziehen sich auf die gesamte Arbeit, die in einem durch die Klausel abgedeckten Geltungsbereich ausgeführt wurde, und erfassen Zusammenfassungsmerkmale zu dieser Arbeit; nicht erfasst werden spezifische Details zu den einzelnen Aktivitäten. Die Klausel COLLECT AGGREGATE ACTIVITY DATA kann für DB2-Workloads, DB2-Serviceklassen und DB2-Arbeitsaktionssets angegeben werden. Verwenden Sie für die normale Betriebsüberwachung die Klausel COLLECT AGGREGATE ACTIVITY DATA, da sie sehr

einfach ist, automatisch vom Statistikereignismonitor zur Protokollanalyse erfasst werden kann und wichtige Informationen zu den allgemeinen Antwortzeitmustern bereitstellt. Wenn ein genaueres Verständnis eines bestimmten Typs der Arbeit erforderlich ist, verwenden Sie die Aktionen COUNT ACTIVITY oder COLLECT AGGREGATE ACTIVITY DATA in einem DB2-Arbeitsaktionsset, um (mit minimalem Aufwand) differenziertere Informationen zu den unterschiedlichen Typen der Arbeit in einer Workload, Serviceklasse oder Datenbank zu erfassen.

Dagegen enthalten Aktivitätsinformationen detaillierte Informationen zu jeder einzelnen Aktivität, die in dem von der Klausel COLLECT ACTIVITY DATA abgedeckten Bereich ausgeführt wird. Diese Klausel kann für DB2-Workloads, DB2-Serviceklassen, DB2-Arbeitsaktionssets und DB2-Schwellenwerte angegeben werden. Sie erlaubt eine detailliertere Analyse der einzelnen erfassten Aktivitäten, um den Ablauf und den Typ von SQL-Anweisungen zu verstehen, die z. B. von einer neuen Anwendung übergeben werden, oder um mit Tools wie EXPLAIN oder dem Designadvisor nach Möglichkeiten zur Leistungsoptimierung zu suchen. Da sie viel mehr Informationen zu jeder betroffenen Aktivität erfasst, ist die Auswirkung dieser Klausel bei den betroffenen Aktivitäten höher als andere Überwachungsmethoden und sollte daher sorgfältig kontrolliert werden.

Wie funktioniert DB2-Workload-Management mit dem neuen AIX-WPAR-Feature?

Alle Aspekte des DB2-Workload-Managements funktionieren in einer AIX-WPAR (WPAR - Workloadpartition). Da AIX-WPARS jedoch nicht die Verwendung von AIX-WLM-Features unterstützen, bringt die Option zur nahtlosen Integration von DB2-Serviceklassen in AIX-WLM-Serviceklassen in dieser Umgebung keinen Vorteil.

Welche Beziehung besteht zwischen der Registry-Variablen DB2_OPT_MAX_TEMP_SIZE und einem DB2-Schwellenwert, der auf SQLTEMPSPACE basiert?

Es besteht keine direkte Beziehung. Die Registry-Variable **DB2_OPT_MAX_TEMP_SIZE** weist den Abfragecompiler an, die Größe des Tabellenbereichs für temporäre Tabellen zu beschränken, den eine Abfrage verwenden kann. Dies kann dazu führen, dass das Optimierungsprogramm einen Plan auswählt, der aufwendiger (möglicherweise weniger effizient) ist, der aber weniger Platz in den Tabellenbereichen für temporäre Systemtabellen verwendet. Ein DB2-Schwellenwert basierend auf SQLTEMPSPACE wirkt sich nicht auf den Typ des Plans aus, der vom Optimierungsprogramm ausgewählt wird. Er veranlasst den DB2-Datenserver einfach, die Nutzung des Tabellenbereichs für temporäre Systemtabellen durch diese Abfrage in jedem Datenbankmember zu überwachen und generiert einen Schwellenwertverstoß, wenn der angegebene Grenzwert bei der normalen Verarbeitung überschritten wird.

Query Patroller wird nicht weiterverwendet und DB2 Governor wird nicht weiter unterstützt. Wie migriere ich jetzt auf DB2-Workload-Manager?

Seit der Einführung von DB2-Workload-Manager als strategische Workload-Management-Lösung in DB2 Version 9.5 wird Query Patroller im Version 10.1-Release nicht weiterverwendet und wird DB2 Governor seit DB2 Version 9.7 nicht weiter unterstützt und beide Produkte werden möglicherweise in einem künftigen Release entfernt.

DB2 Governor wird in diesem Release zwar noch unterstützt, Sie sollten aber die neuen Features und Leistungsmerkmale von DB2-Workload-Manager, einschließlich der in diesem Release eingeführten, schon jetzt verwenden. DB2-Workload-Manager bietet Ihnen wesentlich mehr Optionen, mit denen Sie sich vertraut machen sollten. Möglicherweise müssen Sie dann Ihre Methode zur Steuerung der Arbeit auf Ihrem DB2-Datenserver im Hinblick auf das aktuelle Workload-Management neu überdenken. Der Artikel *Implementing DB2 workload management in a data warehouse* zu bewährten DB2-Verfahren enthält einen Zusatz, der speziell für Benutzer konzipiert ist, die von Query Patroller migrieren. Im Abschnitt mit den zugehörigen Tasks sind ebenfalls relevante Artikel verfügbar.

Zur Vereinfachung der Migration von DB2 Query Patroller auf DB2-Workload-Manager wird das Fixpack für DB2 Version 9.7 mit einem Beispielscript (`qpwlmmig.pl`) geliefert. Details zum Migrieren von Query Patroller auf DB2-Workload-Manager finden Sie in einer der folgenden Tasks:

- Migrieren von Query Patroller auf DB2-Workload-Manager mithilfe des Beispielscripts
- Migrieren von Query Patroller auf DB2-Workload-Manager

Welche Lizenzvoraussetzungen gelten für DB2-Workload-Manager?

Eine Untergruppe der Workload-Management-Funktionen auf dem DB2-Datenserver wird durch Lizenzierung in der Verwendung eingeschränkt. Diese lizenzierte Untergruppe wird als DB2-Workload-Manager bezeichnet. Sie steuert die Erstellung aller Serviceklassen, Workloads, Schwellenwerte und Arbeitsaktionssets. Für den Zugriff auf diese Untergruppe der Workload-Management-Funktionalität ist eines der folgenden Lizenzprodukte erforderlich:

- DB2 Enterprise Server Edition for Linux, UNIX and Windows
- DB2 Advanced Enterprise Server Edition for Linux, UNIX and Windows
- Database Enterprise Developer Edition for Linux, UNIX and Windows
- IBM® InfoSphere Warehouse, alle Editionen
- IBM Smart Analytics System

Die folgenden Workload-Management-Funktionen werden nicht durch eine Lizenz eingeschränkt:

- Verwenden oder Ändern der Standardserviceklassen und -workloads. Dies schließt alle Überwachungsfunktionen ein.
- Erstellen, Ändern oder Löschen von Histogrammschablonen
- Verwenden der Tabellenfunktionen oder gespeicherten Prozeduren des DB2-Workload-Managements
- Erstellen, Aktivieren, Stoppen oder Löschen von Ereignismonitoren für das Workload-Management
- Erteilen, Ändern oder Widerrufen von Workloadzugriffsrechten

Welche Informationen stellen die verschiedenen Ereignismonitore bereit, die zum Workload-Management gehören?

Die Ereignismonitore für Schwellenwertverstöße, Statistikdaten und Aktivitäten erfassen Informationen zu Schwellenwertverstößen, Betriebsstatistikdaten sowie zusammengefasste und einzelne Aktivitätsdaten (siehe „Langzeitüberwachung mit WLM-Ereignismonitoren“ auf Seite 254).

Jeder Ereignismonitor erfasst mindestens eine logische Datengruppe (siehe „Zuordnungen von Ereignistypen zu logischen Datengruppen“ in *Datenbanküberwachung - Handbuch und Referenz*) und es gibt in jeder logischen Datengruppe mindestens ein Monitorelement (siehe „Logische Datengruppen und Monitorelemente des Ereignismonitors“ in *Datenbanküberwachung - Handbuch und Referenz*).

Wenn Sie beispielsweise wissen wollen, welche Informationen vom Ereignismonitor für Schwellenwertverstöße erfasst werden, sehen Sie sich zuerst in Tabelle 3 im Abschnitt zu den Zuordnungen von Ereignistypen zu logischen Datengruppen an. Diese Tabelle zeigt, dass der Ereignismonitor für Schwellenwertverstöße Informationen in nur einer Datengruppe namens `event_thresholdviolations` erfasst (einige Ereignismonitore wie der Aktivitätsereignismonitor erfassen Informationen in mehreren logischen Datengruppen). Sehen Sie sich dann die logische Datengruppe `event_thresholdviolations` im Abschnitt zu den logischen Datengruppen und Monitorelementen des Ereignismonitors an. Diese Abschnitt zeigt, welche Monitorelemente in der logischen Datengruppe `event_thresholdviolations` ausgegeben werden. Hierzu gehören die folgenden Elemente:

- **activate_timestamp** - Zeitmarke für Aktivierung
- **activity_collected** - Aktivitäten erfassen
- **activity_id** - Aktivitäts-ID
- **agent_id** - Anwendungskennung (Agenten-ID)
- **appl_id** - Anwendungs-ID
- **coord_partition_num** - Nummer der Koordinatorpartition
- **destination_service_class_id** - Zielserviceklassen-ID
- **source_service_class_id** - Quellenserviceklassen-ID
- **threshold_action** - Schwellenwertaktion
- **threshold_maxvalue** - Maximaler Wert für Schwellenwert
- **threshold_predicate** - Vergleichselement für Schwellenwert
- **threshold_queuesize** - Größe der Schwellenwertwarteschlange
- **thresholdid** - Schwellenwert-ID
- **time_of_violation** - Zeitpunkt des Verstoßes
- **uow_id** - UOW-ID

Mithilfe der in diesem Beispiel dargestellten Methode kann ermittelt werden, welche Daten von den jeweiligen Ereignismonitoren erfasst werden.

Wie stelle ich fest, welche Aktivitäten von einem Workload-Management-Schwellenwert in die Warteschlange gestellt werden, sowie deren Reihenfolge?

Hierzu können Sie zuerst mithilfe der Tabellenfunktion `WLM_GET_SERVICE_CLASS_AGENTS` eine Sicht erstellen und dann Anweisungen ausführen, um die in die Warteschlange gestellten Aktivitäten in der Reihenfolge des Zeitpunkts ihres Eingangs in die Warteschlange aufzulisten. Beispiele für die hierzu erforderliche Vorgehensweise finden Sie in „Beispiel: Ermitteln der von einem WLM-Schwellenwert in die Warteschlange eingereichten Aktivitäten sowie der Reihenfolge dieser Aktivitäten“ auf Seite 253.

Wie ändert sich das Verhalten, wenn ich den Workload-Management-Dispatcher aktiviere?

Wenn Sie den Workload-Management-Dispatcher über den Konfigurationsparameter `wlm_dispatcher` des Datenbankmanagers aktivieren und die Arbeit von Service-

klassen bisher über die Agentenpriorität priorisiert haben, ist dies nun nicht mehr möglich. Alle Serviceklassen werden jetzt so behandelt, als ob ihnen die Standard-agentenpriorität zugeordnet worden wäre.

Wenn Sie CPU-Anteile über den Konfigurationsparameter `wlm_disp_cpu_shares` des Datenbankmanagers aktivieren und für Ihre Serviceklasse keine CPU-Anteile oder CPU-Grenzwerte angeben, empfangen alle Serviceklassen einen gleichen veränderlichen Anteil der CPU-Ressourcen auf Ihrem System. Dies kann zu einer anderen Zuordnung von CPU-Ressourcen zu Serviceklassen als in vorherigen DB2-Releases führen. Legen Sie daher für Ihre Workload geeignete Werte für CPU-Anteile oder CPU-Grenzwerte fest. Weitere Informationen zum Ermitteln von Werten für CPU-Anteile oder CPU-Grenzwerte finden Sie in „Workload-Management-Dispatcher“ auf Seite 176.

Sind Schwellenwerte für gemeinsamen Zugriff wie CONCURRENTDBCOORDACTIVITIES durch die Einführung des Workload-Management-Dispatchers nicht mehr erforderlich oder nützlich?

Der DB2-Workload-Management-Dispatcher und Schwellenwerte für gemeinsamen Zugriff können nicht miteinander kombiniert werden. Schwellenwerte für gemeinsamen Zugriff sind für die Steuerung des Volumens ausgeführter Arbeit weiterhin sehr nützlich. Für jede Aktivität, die ausgeführt wird, stellt der DB2-Datenbankmanager zusätzlich zu CPU-Ressourcen weitere Ressourcen bereit, die die Aktivität gewöhnlich für die Dauer ihrer Ausführung beibehält. Zu diesen Nicht-CPU-Ressourcen gehören (unter anderem) der DB2-Agent, der Sortierspeicher, der Tabellenbereich für temporäre Tabellen, Sperren und die Ein-/Ausgabe. Durch das Verhindern der Aktivitätsausführung werden diese zusätzlichen Nicht-CPU-Ressourcen nicht verbraucht und sind für andere Aktivitäten verfügbar.

Außerdem können die Schwellenwerte für gemeinsamen Zugriff an verschiedenen Punkten im DB2-Datenbankmanager angewendet werden, um den Ursprung der ausgeführten Arbeit zu ermitteln. Beispielsweise beschränkt die Definition eines Schwellenwerts für gemeinsamen Zugriff für große Abfragen aus einer bestimmten Workload den Verbrauch oder den Anteil der verfügbaren Ressourcen auf diese Workload in einer Serviceklasse und schließt andere Workloads aus, die zur selben Serviceklasse beitragen.

Zusammenfassend kann festgestellt werden, dass mit Schwellenwerten für gemeinsamen Zugriff gesteuert werden kann, wann die Ausführung von Aktivitäten und deren Verbrauch von CPU-Ressourcen und Nicht-CPU-Ressourcen auf dem System anfangen. Mit dem Workload-Management-Dispatcher kann gesteuert werden, wie viele CPU-Ressourcen diese Aktivitäten nach dem Start ihrer Ausführung belegen.

Warum sollte ich AIX-WLM oder Linux-WLM verwenden?

Selbst wenn Sie die CPU-Belegung Ihrer DB2-Workloads mit dem DB2-Workload-Management-Dispatcher steuern, sprechen die folgenden Gründe für die Verwendung von AIX-WLM oder Linux-WLM:

- Betriebssystem-Workload-Manager liefern Überwachung der Ressourcenauslastung auf der Betriebssystemebene.
- Betriebssystem-Workload-Manager können die Steuerelemente für alle Prozesse oder Threads auf dem gesamten Host oder in der gesamten logischen Partition bereitstellen, nicht nur für DB2-Datenbankmanagerthreads. Dies kann beim Steuern von Prozessen hilfreich sein, die mit dem DB2-Datenbankmanager um Ressourcen konkurrieren.

Kapitel 2. Identifizieren der Arbeit

Ein wichtiger Teil der erfolgreichen Implementierung einer Lösung für das DB2-Workload-Management ist das Identifizieren der Arbeit.

Es gibt drei Methoden beim Identifizieren der Arbeit: Workloads, Arbeitsklassen und Datentags.

- Mit Workloads kann die Quelle der Arbeit identifiziert werden. Sie können die Quelle auf der Verbindungs- oder Transaktionsebene identifizieren, indem Sie wichtige Sitzungsattribute wie z. B. den Anwendungsnamen oder die Systemberechtigungs-ID verwenden, über die die Arbeit übergeben wurde.
- Mithilfe einer Arbeitsklasse können Sie Arbeit identifizieren, indem bestimmte Merkmale der Arbeit genau bestimmt werden. Sie können eine Arbeitsklasse definieren, um relevante Arbeit zu identifizieren, wie beispielsweise Anweisungen, die nur Daten auf dem Datenserver ändern, z. B. INSERT-, UPDATE- oder DELETE-Anweisungen.
- Datentags sind als Datentags für Tabellenbereiche und Speichergruppen verfügbar. Sie ermöglichen es Ihnen, Arbeit indirekt über den Typ der Daten zu identifizieren, auf die während der Ausführung zugegriffen wird. Durch das Tagging relevanter Daten können Sie die Arbeit identifizieren, die auf diese Daten zugeht.

Aktivitäten

Eine Möglichkeit zur Überwachung und Steuerung von Workloads basiert auf einzelnen Aktivitäten. Jedes Mal, wenn Ihr DB2-Datenserver den Zugriffsplan für eine SQL- oder XQuery-Anweisung oder das Dienstprogramm LOAD ausführt, wird eine entsprechende Aktivität erstellt.

Zur Überwachung der Workload stellen häufig verwendete Monitorelemente Informationen zu Aktivitätseinheiten bereit. Sie können beispielsweise Informationen zum Workloadvolumen und zur Antwortzeit von Monitorelementen abrufen, beispielsweise vom Monitorelement für die Ausführungszeit von Aktivitäten (coord_act_exec_time) und für die obere Grenze für gleichzeitig ablaufende Aktivitäten (concurrent_act_top).

Bei der Workloadsteuerung gelten die meisten Workloadsteuerelemente und Schwellenwerte für jede Aktivität. Der Schwellenwert ACTIVITYTOTALTIME steuert beispielsweise die Zeitdauer, die Ihr Datenserver höchstens für die Verarbeitung einer Aktivität aufwenden kann.

Anweisungen und Befehle, die Aktivitäten auf Ihrem Datenserver auslösen

Die folgenden Anweisungen und Befehle lösen Aktivitäten auf Ihrem Datenserver aus:

- Alle DML-Anweisungen
- Alle DDL-Anweisungen
- Die Anweisung CALL
- Das Dienstprogramm LOAD

Lebensdauer von Aktivitäten

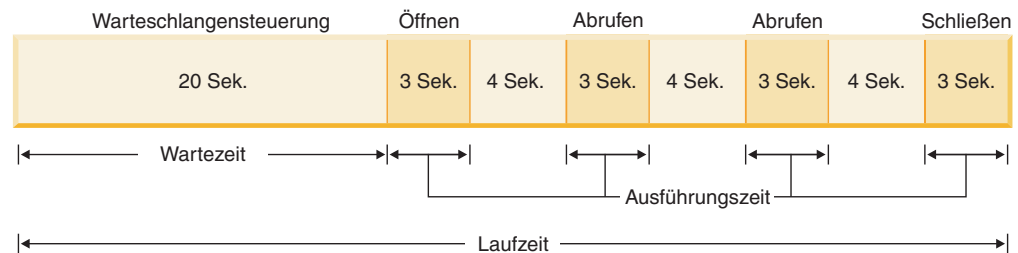
Die Lebensdauer einer Aktivität für eine DML-Anweisung umfasst nicht die Verarbeitung, die vor oder außerhalb der Ausführung des Zugriffsplans stattfindet. Dies heißt, dass die auf Aktivitäten basierende Überwachung keine Operationen wie die Herstellung einer Verbindung zur Datenbank oder die Kompilierung von SQL in einen Zugriffsplan umfasst.

Eine Aktivität kann sich während ihrer Lebensdauer in verschiedenen Status befinden, die vom Ereignismonitorelement `activity_state` ausgegeben werden. Eine Aktivität kann sich beispielsweise in folgenden Status befinden:

- EXECUTING - Dieser Status gibt an, dass der Koordinatoragent an der Aktivität arbeitet. Für eine Aktivität, die einen Wartestatus für Sperren feststellt, wird ausgegeben, dass sie ausgeführt wird.
- IDLE - Dieser Status gibt an, dass der Koordinatoragent auf die nächste Anforderung eines Clients wartet.
- QUEUED - Einige Schwellenwerte enthalten eine integrierte Warteschlange. Dieser Status gibt an, dass die Aktivität in der Warteschlange auf ihre Ausführung wartet.

Die Überwachungsdaten für die Aktivität werden am Ende der Lebensdauer einer Aktivität zusammengefasst.

Die folgende Abbildung zeigt, wie sich die Laufzeit einer lang laufenden Abfrage aus Zeit in der Warteschlange und Ausführungszeit zusammensetzt:



Typen von SQL-Anweisungen und Anwendungsentwicklung

In diesem Abschnitt wird beschrieben, welche Aktivitäten für verschiedene SQL-Anweisungen erstellt werden, und es werden die Start- und Endpunkte in der Lebensdauer dieser Aktivitäten angegeben. Diese Informationen verdeutlichen die Überwachung und Steuerung von SQL-Anweisungen über Aktivitäten.

SELECT-Anweisungen: Eine SELECT-Anweisung wird durch eine Aktivität dargestellt. Hierzu gehören alle Cursoranforderungen wie FETCH-Operationen und Subselects oder Unterabfragen. Die Aktivität beginnt, wenn Ihr DB2-Datenserver mit der Verarbeitung einer Anweisung oder Anforderung zum Öffnen des Cursors beginnt, und endet, wenn Ihr Datenserver die Verarbeitung einer Anweisung oder Anforderung zum Schließen des Cursors beendet hat.

SELECT-Anweisungen, die mit WITH HOLD definierte Cursor verwenden: Wenn ein mit WITH HOLD definierter Cursor verwendet wird, kann eine Anwendung einen Cursor in einer UOW (Unit of Work - Arbeitseinheit) öffnen und den Cursor in einer nachfolgenden UOW schließen. Der Cursor bleibt für mehrere UOWs geöffnet. Die entsprechende Aktivität bleibt so lange vorhanden, wie der Cursor geöffnet ist, da die Lebensdauer der Aktivität erst endet, wenn der Cursor geschlossen wird.

Anweisung CALL und gespeicherte Prozeduren: Eine Anweisung CALL wird von einer einzigen Aktivität dargestellt, aber die Nutzdaten der gespeicherten Prozedur können verschachtelte Aktivitäten wie folgt starten:

Tabelle 1. Inhalt gespeicherter Prozeduren und von ihnen erstellte Aktivitäten

Inhalt der gespeicherten Prozedur	Zusätzliche erstellte Aktivitäten
Eine einzelne SQL-Anweisung	Eine
Keine SQL-Anweisungen in der gespeicherten Prozedur	0
SQL-Prozeduren, mehrere SQL-Anweisungen und Schleifenlogik	Mehrere Aktivitäten, von denen jeweils eine jedem Aufruf jeder Anweisung entspricht
Ein Aufruf einer anderen gespeicherten Prozedur	Aktivitäten für diese gespeicherte Prozedur

Die der Anweisung CALL zugeordnete Aktivität startet, wenn Ihr DB2-Datenserver mit der Verarbeitung der Anweisung oder Anforderung beginnt, und endet, wenn die Verarbeitung der gespeicherten Prozedur abgeschlossen ist.

Trigger und benutzerdefinierte Funktionen (UDFs): Wenn eine SQL-Anweisung einen Trigger oder eine benutzerdefinierte Funktion aufruft, wird keine zusätzliche Aktivität erstellt. Die von diesem Trigger oder dieser benutzerdefinierten Funktion geleistete Arbeit fällt der Aktivität für die SQL-Anweisung zu, die ihn bzw. sie aufgerufen hat. Fälle, in denen der Trigger oder die benutzerdefinierte Funktion zusätzliche SQL-Anweisungen ausführt, werden wie eine beliebige Anweisungsausführung behandelt, d. h., es wird für jede Anweisung eine Aktivität erstellt.

Anweisung PREPARE: Es wird keine Aktivität erstellt, da Aktivitäten erst nach der Ausführung eines Zugriffsplans erstellt werden.

Verschachtelte Aktivitäten

Verschachtelte Aktivitäten haben keine bedeutende Auswirkung auf die auf Aktivitäten basierende Überwachung und Steuerung von Workloads. Es gelten jedoch einige zusätzliche Informationen.

Folgende Aktivitäten können verschachtelte Aktivitäten enthalten:

- Eine gespeicherte Prozedur
- Ein anonymer Block
- Eine autonome Routine
- Eine DML-Aktivität, die eine benutzerdefinierte Funktion ausführt
- Laden von einem Cursor (eine Ladeaktivität, in der die Cursoraktivität verschachtelt ist)
- Eine DML-Aktivität, die einem Trigger unterliegt, der alle oben aufgeführten Aktivitäten als Teil der Trigger-Definition enthält

Die verschachtelten Aktivitäten werden in den Überwachungsdaten wie folgt ausgegeben:

- Eine verschachtelte Aktivität wird durch eine übergeordnete UOW-ID ungleich null und eine übergeordnete Aktivitäts-ID ungleich null angegeben.
- Eine verschachtelte Aktivität wird nicht zu Histogrammen oder aus Histogrammen abgeleiteten Statistiken gezählt.

- Daten für eine verschachtelte Aktivität werden nicht gleichzeitig als Teil der Messdaten für die übergeordnete Aktivität ausgegeben. Wenn zum Beispiel eine durch eine Anweisung CALL ausgeführte Prozedur eine Einfügeoperation durchführt, die 10 Sekunden Prozessorzeit verbraucht, wird diese Prozessorzeit nur zu den Prozessorzeitmessdaten für die Einfügeaktivität und nicht zu den Prozesszeitmessdaten für die übergeordnete CALL-Aktivität gezählt.

Die Workloadsteuerung fasst verschachtelte Aktivitäten wie folgt auf:

- Eine in einer benutzerdefinierten Funktion oder in einem Trigger verschachtelte Aktivität trägt nicht zum Schwellenwert CONCURRENTDBCOORDACTIVITIES bei.
- Eine in einer Ladeaktivität verschachtelte Cursoraktivität trägt nicht zum Schwellenwert CONCURRENTDBCOORDACTIVITIES bei.

Aktivitäten und das Dienstprogramm LOAD

Beim Ausführen des Dienstprogramms LOAD werden mehrere Aktivitäten erstellt, darunter eine Ladeaktivität und mehrere andere Aktivitäten des Typs READ, WRITE oder OTHER. Beim Laden von einem Cursor wird eine zusätzliche Aktivität für den Cursor erstellt, von dem die Ladeaktivität lädt. Diese Cursoraktivität ist eine verschachtelte Aktivität der Ladeaktivität.

Überwachen von Aktivitätsereignissen

Beim Überwachen von Aktivitäten mit einem Aktivitätsereignismonitor in einer Umgebung mit Datenbanken mit mehreren Mitgliedern müssen Sie beachten, wie Informationen zu Ereignissen erfasst werden. Wenn Sie Aktivitätsereignisse für alle Mitglieder erfassen, sehen Sie für eine bestimmte Aktivität möglicherweise mehrere Ereignisse für andere Mitglieder als den Koordinator. Möglicherweise werden pro Aktivität mehrere Datensätze aufgezeichnet, da die Aktivität abhängig von der Planung der Ereignisreihenfolge in einem Abschnitt für die fernen Mitglieder (die ausführbare Form einer SQL-Anweisung) auftreten und wieder verschwinden kann. Daher können für die Aktivität für einen Mitglied, der kein Koordinator ist, mehrere Datensätze aufgezeichnet werden. Sie müssen alle Datensätze für die Aktivität berücksichtigen, um die Verarbeitungsvorgänge der Aktivität für den fernen Mitglied zu verstehen. Es kann beispielsweise sinnvoll sein, Messdaten aller Datensätze für die Aktivität für den fernen Mitglied zusammenzufassen.

Im Unterschied dazu wird eine Aktivität in den für den Koordinatormitglied erfassten Ereignisdaten nur einmal aufgezeichnet.

DDL-Anweisungen für Workload-Management

DDL-Anweisungen für das Workload-Management bestehen aus den CREATE-, ALTER- und DROP-Anweisungen, die Sie beim Arbeiten mit Serviceklassen, Workloads, Arbeitsklassensets, Arbeitsaktionssets, Schwellenwerten und Histogrammen verwenden.

Es gibt die folgenden DDL-Anweisungen für das DB2-Workload-Management:

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

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

DDL-Anweisungen beim Workload-Management unterscheiden sich von anderen DB2-DDL-Anweisungen:

- Für alle Datenbankmember ist jeweils nur eine nicht festgeschriebene DDL-Anweisung für das Workload-Management zulässig. Wenn eine nicht festgeschriebene DDL-Anweisung für das Workload-Management vorhanden ist, warten nachfolgende DDL-Anweisungen für das Workload-Management, bis für die nicht festgeschriebene DDL-Anweisung für das Workload-Management entweder COMMIT oder ROLLBACK ausgeführt wurde. DDL-Anweisungen für das Workload-Management werden in der Reihenfolge verarbeitet, in der sie abgesetzt werden.
- Auf jede DDL-Anweisung für das Workload-Management muss eine Anweisung COMMIT oder ROLLBACK folgen.
- Eine DDL-Anweisung für das Workload-Management kann nicht in einer XA-Transaktion abgesetzt werden. Nachdem eine Verbindung eine DDL-Anweisung für das Workload-Management absetzt, muss dieselbe Verbindung direkt nach dieser Anweisung eine Anweisung COMMIT oder ROLLBACK absetzen. Bei XA-Transaktionen ist es möglich, dass mehrere Verbindungen mit einer Transaktion verknüpft sind und dass jede der Verbindungen ein Commit oder Rollback für die Transaktion durchführen kann. In dieser Situation kann nicht sichergestellt werden, dass die Workload-Management-Umgebung korrekt implementiert wird.
- DB2 for z/OS erkennt DDL-Anweisungen für das Workload-Management in DB2 Database for Linux, UNIX, and Windows nicht.

Identifizieren der Arbeit nach Ursprung mithilfe von Workloads

Workloads identifizieren eingehende Arbeit basierend auf ihrer Quelle, sodass sie später unabhängig von aller anderen Arbeit durch das DB2-Workload-Management überwacht oder verwaltet werden kann. Die Quelle wird anhand der Attribute der Datenbankverbindung ermittelt, unter der die Arbeit übergeben wird.

Die Attribute einer Verbindung werden zum ersten Mal bewertet, wenn die Verbindung hergestellt wird und die Verbindung mit einer Workloaddefinition abgeglichen wird. Diese Beziehung zwischen einer Verbindung und einer bestimmten Workloaddefinition wird als *Workloadvorkommen* bezeichnet. Wenn sich eines der Verbindungsattribute während der Verbindung ändert, wird die Workloadzuweisung am Anfang der nächsten UOW (Unit of Work - Arbeitseinheit) nach der Änderung neu bewertet. Wenn zu diesem Zeitpunkt eine neue Workloaddefinition eine größere Übereinstimmung mit der Verbindung aufweist, wird das alte Workloadvorkommen (für die zuvor zugeordnete Workload) beendet und ein neues Vorkommen für die neu zugeordnete Workloaddefinition wird gestartet. Obwohl jede Verbindung jeweils nur einer einzigen Workload zugeordnet ist, können ein und derselben Workload mehrere Verbindungen (Workloadvorkommen) gleichzeitig zugeordnet sein. Weitere Informationen finden Sie in „Workloadzuordnung“ auf Seite 24.

Wenn z. B. alle von der Anwendung Accounts hergestellten Verbindungen einer Workload REPORTING zugeordnet werden sollen, die die Aktivitäten unter diesen Verbindungen zur Ausführung in der Serviceklasse Marketing zuordnet, dann setzen Sie eine Anweisung CREATE WORKLOAD wie die folgende ab:

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

Dadurch wird die folgende Workload erstellt:



Abbildung 2. Workload REPORTING

Erteilen Sie PUBLIC anschließend das Zugriffsrecht USAGE für die Workload REPORTING:

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

Wenn alle von der Anwendung Accounts erstellten Aktivitäten unter den Verbindungen, die zur Sitzungsbenutzergruppe Deptmgr gehören, der Workload SUMMARY zugeordnet werden sollen, die die Aktivitäten der Serviceklasse HumanResources zuordnet, setzen Sie eine Anweisung wie die folgende ab:

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

Dadurch wird die folgende Workload erstellt:

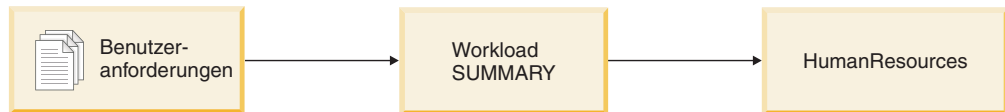


Abbildung 3. Workload SUMMARY

Erteilen Sie PUBLIC anschließend das Zugriffsrecht USAGE für die Workload SUMMARY:

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

Sie können Ihre Workloaddefinitionen durch Abfragen der Sicht SYSCAT.WORKLOADS anzeigen. Die Verbindungsattribute, die Sie für jede Workload angegeben haben, können Sie durch Abfragen der Sicht SYSCAT.WORKLOADCONNATTR anzeigen. Sie können durch Abfragen der Sicht SYSCAT.WORKLOADAUTH anzeigen, wer berechtigt ist, eine Workload zu verwenden. Verwenden Sie die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES oder MON_GET_WORKLOAD, um anzuzeigen, welche Workloadvorkommen zu einem beliebigen Zeitpunkt auf dem System vorhanden sind.

SYSDEFAULTUSERWORKLOAD ist die Standardworkload. Jede Verbindung, die bei der Workloadbewertung nicht einer benutzerdefinierten Workload zugeordnet wurde, wird dieser Standardworkload zugeordnet, wodurch sichergestellt wird, dass alle Datenbankverbindungen mit einer Workload verknüpft sind. Arbeit, die der Standardworkload SYSDEFAULTUSERWORKLOAD zugeordnet ist, wird standardmäßig in der Serviceklasse SYSDEFAULTUSERCLASS ausgeführt.

Unterstützte Datenbankverbindungsattribute

Sie müssen mindestens ein Datenbankverbindungsattribut in der Workloaddefinition angeben. Jedes Verbindungsattribut kann einen oder mehrere Werte haben. Wenn Sie in der Workloaddefinition keinen Wert für ein bestimmtes Verbindungsattribut angeben, untersucht der Datenserver dieses Attribut bei der Workloadbewertung nicht.

Tabelle 2. Verbindungsattribute in einer Workloaddefinition

Verbindungsattribut	Beschreibung
Adresse	Die tatsächliche Kommunikationsadresse, die vom Client zur Kommunikation mit dem Datenbankserver verwendet wird. Als Protokoll wird nur TCP/IP unterstützt. Die Adresse muss eine IPv4-Adresse, eine IPv6-Adresse oder der Name einer gesicherten Domäne sein.
Anwendungsname	Der Name der Anwendung, die auf dem Client ausgeführt wird, so wie er dem Datenserver bekannt ist. Der Anwendungsname entspricht dem Wert im Feld Anwendungsname in der Systemmonitorausgabe. Weitere Informationen finden Sie unter dem Monitorelement appl_name .
Systemberechtigungs-ID	Die Berechtigungs-ID des Benutzers, der die Verbindung zur Datenbank hergestellt hat, so wie sie im Sonderregister SYSTEM_USER festgelegt ist. Sie können den Wert von SYSTEM_USER ändern, indem Sie die Verbindung als Benutzer mit einer anderen Berechtigungs-ID herstellen.
Sitzungsberechtigungs-ID	Die Berechtigungs-ID, die für die aktuelle Sitzung der Anwendung verwendet wird, so wie sie im Sonderregister SESSION_USER festgelegt ist. Sie können den Wert von SESSION_USER mit einem gesicherten Kontext oder der Anweisung SET SESSION AUTHORIZATION ändern.
Gruppe der Sitzungsberechtigungs-ID	Die Gruppen, zu denen der aktuelle Sitzungsbenutzer gehört.
Rolle der Sitzungsberechtigungs-ID	Die Rollen, die dem aktuellen Sitzungsbenutzer erteilt wurden.
Clientbenutzer-ID	Die Clientbenutzer-ID aus den Clientinformationen, so wie sie im Sonderregister CURRENT CLIENT_USERID (oder CLIENT_USERID) festgelegt ist. Sie können den Wert der Clientbenutzer-ID mit einem der definierten Mechanismen, die von bestimmten DB2-Clients bereitgestellt werden, mit der API sqleseti (Clientinformationen festlegen) oder mit der Prozedur WLM_SET_CLIENT_INFO ändern.

Tabelle 2. Verbindungsattribute in einer Workloaddefinition (Forts.)

Verbindungsattribut	Beschreibung
Clientanwendungsname	Der Anwendungsname aus den Clientinformationen, so wie er im Sonderregister CURRENT CLIENT_APPLNAME (oder CLIENT APPLNAME) festgelegt ist. Sie können den Wert des Clientanwendungsnamens mit einem der definierten Mechanismen, die von bestimmten DB2-Clients bereitgestellt werden, mit der API sqleseti oder mit der Prozedur WLM_SET_CLIENT_INFO ändern.
Clientworkstationname	Der Workstationname aus den Clientinformationen, so wie er im Sonderregister CURRENT CLIENT_WRKSTNNAME (oder CLIENT WRKSTNNAME) festgelegt ist. Sie können den Wert des Clientworkstationnamens mit einem der definierten Mechanismen, die von bestimmten DB2-Clients bereitgestellt werden, mit der API sqleseti oder mit der Prozedur WLM_SET_CLIENT_INFO ändern.
Clientabrechnungszeichenfolge	Die Abrechnungszeichenfolge aus den Clientinformationen, so wie sie im Sonderregister CURRENT CLIENT_ACCTNG (oder CLIENT ACCTNG) festgelegt ist. Sie können den Wert der Clientabrechnungszeichenfolge mit der API sqleseti oder mit der Prozedur WLM_SET_CLIENT_INFO ändern.

Verwendung von Platzhaltern in Verbindungsattributen

Einige Verbindungsattribute unterstützen die Angabe eines Sterns (*) als Platzhalterzeichen in den Anweisungen CREATE WORKLOAD und ALTER WORKLOAD. Sie können Platzhalter in Situationen einsetzen, in denen ein Verbindungsattribut mehrere ähnliche Werte annehmen kann. Diese werden durch einen regulären Ausdruck mit Platzhalterzeichen abgeglichen, ohne dass Verbindungsattribute für jeden der möglichen Werte definiert werden müssen.

Der Stern (*) als Platzhalterzeichen entspricht null oder mehr Zeichen. Wenn eine Übereinstimmung mit einem Stern erforderlich ist, verwenden Sie einen doppelten Stern (**), um den Stern als Literalzeichen anzugeben.

Beispiel: Wenn Sie mehrere Debitorenbuchhaltungsanwendungen (*accrec01*, *accrec02* ... *accrec15*) haben, die alle zur selben Workload gehören sollen, damit sie vom DB2-Workload-Manager gleich behandelt werden, dann definieren Sie beim Erstellen oder Ändern Ihrer Workload das Verbindungsattribut *CURRENT CLIENT_APPLNAME('accrec*')* so, dass sich eine Übereinstimmung mit allen diesen Anwendungen ergibt. Analog dazu ergibt sich für eine Debitorenbuchhaltungsanwendung *acc*rec* (ein Name, der ein Sternzeichen enthält) eine Übereinstimmung mit dem Verbindungsattribut *CURRENT CLIENT_APPLNAME('acc**rec')*.

Die folgenden Workloadverbindungsattribute unterstützen die Verwendung von Platzhalterzeichen:

- APPLNAME
- CURRENT CLIENT_ACCTNG
- CURRENT CLIENT_APPLNAME
- CURRENT CLIENT_USERID
- CURRENT CLIENT_WRKSTNNAME

Festlegen von Clientinformationen zum Identifizieren von Anforderungen

Viel Anwendungsserver stellen die Verbindungen für alle von ihnen verarbeiteten Clientanforderungen mit denselben Clientinformationen her und geben dieselben Clientinformationen weiter. Einige Produkte, beispielsweise WebSphere und Cognos, stellen eine Funktion für das Senden eindeutiger Informationen zu den einzelnen Anforderungen über die Clientinformationsfelder im Pushdown-Modus bereit. Diese Informationen geben die Endbenutzeranforderung innerhalb von DB2 eindeutig an. Bei den meisten anderen Produkten kann der Anwendungsserver auf andere Weise so angepasst werden, dass vor dem Beginn der Verarbeitung einer Endbenutzeranforderung eindeutige Clientinformationen an DB2 gesendet werden.

Durch die Angabe eindeutiger Clientattribute über den Anwendungsserver wird die Sonderbehandlung von Anforderungen in DB2 und die Zuordnung von Anforderungen unterschiedlicher Clients zu unterschiedlichen Workloads (und zu unterschiedlichen Serviceklassen) aktiviert.

Bewertungsreihenfolge für Verbindungsattribute

Wenn Sie die Nutzungskennndaten Ihrer Umgebung analysieren, können Sie mit der Anweisung CREATE WORKLOAD Ihre eigenen Workloads erstellen und diese bestimmten Serviceklassen zuordnen. Beim Erstellen der Workload definieren Sie sowohl die Werte, die zur Bewertung der Verbindungsattribute bei der Workloadzuordnung verwendet werden, als auch die Reihenfolge, in der die Workload relativ zu anderen Workloads bewertet wird. Da mehr als eine Workload mit eingehenden Verbindungsattributen übereinstimmen kann, können Sie durch die Änderung der Bewertungsreihenfolge festlegen, welche übereinstimmende Workload ausgewählt wird. Welche übereinstimmende Workload ausgewählt wird, hängt auch davon ab, ob der Sitzungsbenutzer das Zugriffsrecht USAGE für die Workload hat. Weitere Informationen finden Sie in „Workloadzuordnung“ auf Seite 24.

Die folgende Abbildung zeigt, wie mehrere Anforderungen in Bezug auf Workloads in der Reihenfolge A, B, C und D bewertet und dann bestimmten Workloads zugeordnet und in der zutreffenden Serviceklasse ausgeführt werden. Anforderungen, die mit keiner vorhandenen Workload übereinstimmen, werden der Workload SYSDEFAULTUSERWORKLOAD zugeordnet. Informationen zu den Aktivitätstypen, die in der Standardverwaltungsklasse und der Standard-systemklasse ausgeführt werden, finden Sie in „Standardservicesuperklassen und -unterklassen“ auf Seite 74.

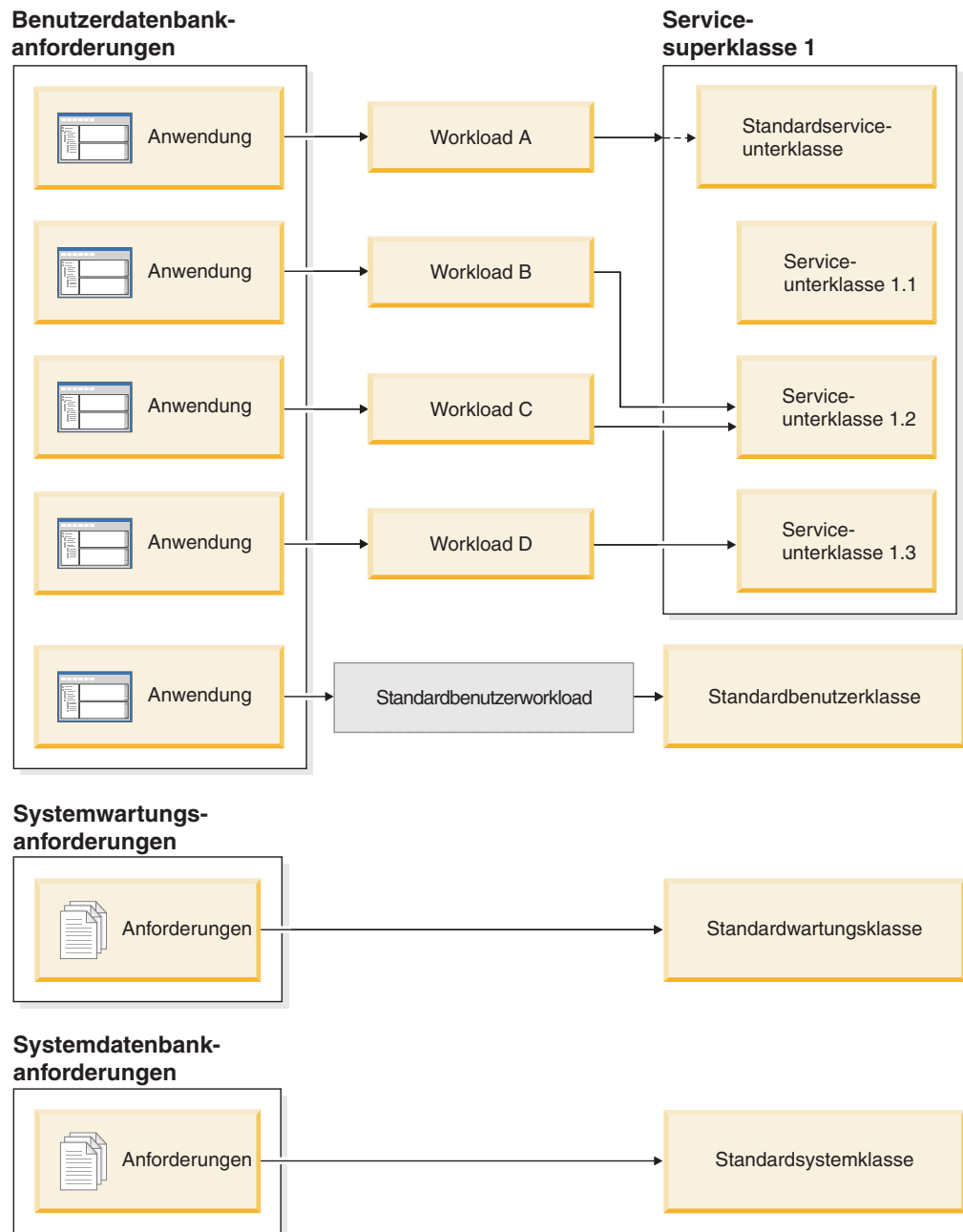


Abbildung 4. Serviceklassen und Workloads

Workloadzuordnung

Am Anfang der ersten UOW (Unit of Work - Arbeitseinheit) nach dem Herstellen der Datenbankverbindung ordnet der Datenserver die Verbindung einer Workload zu, indem er die Verbindungsattribute jeder aktivierten Workload bewertet.

Die Reihenfolge, in der die Workloads bewertet werden, wird vom Spaltenwert EVALUATIONORDER jeder Workload in der Tabelle SYSCAT.WORKLOADS bestimmt. Wenn eine Workload mit übereinstimmenden Verbindungsattributen gefunden wird, prüft der Datenserver, ob der aktuelle Sitzungsbenutzer das Zugriffsrecht USAGE für die Workload hat. Hat der Benutzer das Zugriffsrecht USAGE für

die übereinstimmende Workload, ist die Workloadzuordnung abgeschlossen, und die Verbindung wird dieser Workload zugeordnet. Hat der Benutzer das Zugriffsrecht USAGE für die übereinstimmende Workload nicht, wertet der Datenserver weitere Workloads aus, bis er eine übereinstimmende Workload findet, für die der Sitzungsbenutzer das Zugriffsrecht USAGE hat. Wenn keine übereinstimmende Workload gefunden wird, versucht der Datenserver, die Workload SYSDEFAULTUSERWORKLOAD zu verwenden. Wenn der aktuelle Sitzungsbenutzer nicht das Zugriffsrecht USAGE für diese Workload hat, wird SQL4707N zurückgegeben, und die UOW wird abgewiesen. Andernfalls wird die Verbindung der Workload SYSDEFAULTUSERWORKLOAD zugeordnet.

Sie können die Bewertungsreihenfolge mit dem Schlüsselwort POSITION der Anweisung CREATE WORKLOAD oder ALTER WORKLOAD wie folgt festlegen:

- Durch Angabe der absoluten Position der Workload in der Bewertungsreihenfolge (siehe folgendes Beispiel):

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

POSITION AT 2 bedeutet, dass die Workload an zweiter Stelle in der Bewertungsreihenfolge platziert wird. Eine übereinstimmende Workload, die in der Bewertungsreihenfolge höher platziert ist, wird zuerst bewertet. Wenn also die Workloads an Position 2 und Position 3 übereinstimmen, wird die Workload an Position 2 vor der Workload an Position 3 bewertet.

Wenn die in der Anweisung CREATE WORKLOAD oder ALTER WORKLOAD festgelegte Position größer als die Gesamtzahl vorhandener Workloads ist, wird die Workload als vorletzte vor der Workload SYSDEFAULTUSERWORKLOAD in der Bewertungsreihenfolge platziert. Dies hat dieselbe Auswirkung wie die Angabe von POSITION LAST in der Anweisung CREATE WORKLOAD oder ALTER WORKLOAD.

- Durch Verwendung des Schlüsselworts POSITION BEFORE *workloadname* oder POSITION AFTER *workloadname*, wobei *workloadname* eine vorhandene Workload ist. Dieses Schlüsselwort gibt die Position einer neuen oder geänderten Workload relativ zu einer anderen Workload in der Bewertungsreihenfolge an (siehe folgendes Beispiel):

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

Wenn Sie das Schlüsselwort POSITION nicht angeben, wird die neue Workload standardmäßig nach allen anderen definierten Workloads in der Bewertungsreihenfolge, aber noch vor der Workload SYSDEFAULTUSERWORKLOAD platziert, die immer als letzte berücksichtigt wird.

Erneute Workloadzuordnung

Eine Verbindung kann die ihr zugewiesene Workload potenziell an jeder UOW-Grenze ändern, wenn sich eines der Verbindungsattribute oder die Workloaddefinitionsgruppe ändert. Eine UOW-Grenze ist der Punkt, an dem eine Verbindung die Zuordnung zu ihrer aktuellen Transaktion aufhebt. Die folgenden Ereignisse haben eine UOW-Grenze zur Folge: ein Commit, ein Rollback, ein XA-Ende (Erfolg), ein XA-Commit und ein XA-Rollback.

Die Workloadzuordnung wird am Anfang einer neuen UOW neu bewertet, wenn der Datenserver erkennt, dass eines der folgenden Ereignisse aufgetreten ist:

- Ein relevantes Verbindungsattribut hat sich geändert. Eine Liste der Verbindungsattribute, die Sie in einer Workloaddefinition angeben können, finden Sie in der Tabelle in „Identifizieren der Arbeit nach Ursprung mithilfe von Workloads“ auf Seite 19. Die Neubewertung der Workload erfolgt auch, wenn

die aktuelle Sitzungsberechtigungs-ID sich ändert, weil die Datenbankverbindung aufgrund eines gesicherten Kontexts wechselt. Weitere Informationen finden Sie in Gesicherte Kontexte und gesicherte Verbindungen.

- Sie haben eine Workload erstellt oder geändert.
- Sie haben das Zugriffsrecht USAGE für eine Workload einem Benutzer, einer Gruppe oder einer Rolle erteilt oder das Zugriffsrecht USAGE für eine Workload einem Benutzer, einer Gruppe oder einer Rolle entzogen.

Eine Verbindung kann keiner anderen Workload neu zugeordnet werden, während eine Aktivität, die eine UOW-Grenze umfasst, noch aktiv ist. Eine Aktivität kann eine Operation sein, die Ressourcen über mehrere UOWs verwaltet, z. B. eine Ladeoperation, eine gespeicherte Prozedur, eine Tabellenfunktion oder ein WITH HOLD-Cursor. Das aktuelle Workloadvorkommen wird ausgeführt, bis alle Aktivitäten abgeschlossen sind. Die erneute Workloadzuordnung erfolgt dann am Anfang der nächsten UOW.

Eine versuchte (erneute) Workloadzuordnung führt zu einem Fehler SQL4707N, wenn einer der folgenden Fälle vorliegt:

- Der Datenserver versucht, die Verbindung einer Workload zuzuweisen, die keine Berechtigung für den Datenbankzugriff hat. Weitere Informationen finden Sie in „Verhindern des Datenbankzugriffs für Vorkommen einer Workload“ auf Seite 38.
- Der Datenserver versucht, die Verbindung der Workload SYSDEFAULTUSERWORKLOAD zuzuordnen, aber der aktuelle Sitzungsbenutzer hat nicht das Zugriffsrecht USAGE für diese Workload.

Wenn Sie die Berechtigung ACCESSCTRL, DATAACCESS, DBADM, SECADM oder WLMADM haben, können Sie Ihre Datenbankverbindung der Standardverwaltungsworkload (SYSDEFAULTADMWORKLOAD) zuordnen. Weitere Informationen finden Sie in „Maßnahmen zur Fehlerbehebung mit der Standardverwaltungsworkload“ auf Seite 29.

XA-Transaktionen und erneute Workloadzuordnung

XA-Aufrufe wie XA_END (Erfolg), XA-Commit und XA-Rollback setzen einen DB2-Befehl COMMIT oder ROLLBACK ab, wodurch das Ende einer UOW angegeben wird. Da die Workloadneubewertung am Anfang einer UOW erfolgen kann, können diese XA-Aufrufe die Workloadneubewertung einleiten, obwohl die Ursache für die Workloadneubewertung nicht direkt mit der XA-Transaktion selbst zusammenhängt.

Standardworkloads

Die Standardbenutzerworkload SYSDEFAULTUSERWORKLOAD stellt eine Workload für Ihren Datenserver bereit, der anfänglich alle Verbindungen zugeordnet werden. Die Standardverwaltungsworkload SYSDEFAULTADMWORKLOAD erlaubt es Ihnen, Verwaltungsmaßnahmen zur Fehlerbehebung zu ergreifen, die auf andere Weise nicht möglich sind. Beide Workloads werden bei der Datenbankerstellung erzeugt und können nicht gelöscht werden.

Standardbenutzerworkload (SYSDEFAULTUSERWORKLOAD)

Verbindungen, die der Standardworkload zugewiesen sind, werden der Standardbenutzerservicesuperklasse SYSDEFAULTUSERCLASS zugeordnet, die die Standardausführungsumgebung bereitstellt. Sie können den benutzerdefinierten Ser-

viceklassen Verbindungen zuordnen, indem Sie benutzerdefinierte Workloads erstellen. Darüber hinaus können Sie SYSDEFAULTUSERWORKLOAD so ändern, dass sie Verbindungen einer anderen Serviceklasse als SYSDEFAULTUSERCLASS zuordnet.

Sie können die Workload SYSDEFAULTUSERWORKLOAD durch Abfragen der Tabelle SYSCAT.WORKLOADS anzeigen.

Die folgende Tabelle zeigt die für die Workload SYSDEFAULTUSERWORKLOAD in der Sicht SYSCAT.WORKLOADS zurückgegebenen Spalten zusammen mit Werten und gibt an, ob diese Werte geändert werden können. Informationen zum Zuordnen einer Verbindung zur Workload SYSDEFAULTUSERWORKLOAD finden Sie in „Workloadzuordnung“ auf Seite 24.

Tabelle 3. Eintrag SYSDEFAULTUSERWORKLOAD in SYSCAT.WORKLOADS

Spalte	Wert	Modifizierbar mit der Anweisung ALTER WORKLOAD, wenn Sie die Berechtigung DBADM oder WLMADM haben (und SQLADM for COLLECT-Klauseln)
WORKLOADID	1	Nein
WORKLOADNAME	SYSDEFAULTUSERWORKLOAD	Nein
EVALUATIONORDER	Zweitletzte	Nein
CREATE_TIME	Zeitmarke der Datenbankerstellung	Nein
ALTER_TIME	Zeitmarke der letzten Aktualisierung der Workloaddefinition	Nein (aber der Datenserver ändert diese Spalte bei der Aktualisierung der Workloaddefinition)
ENABLED	Y	Nein
ALLOWACCESS	Y	Ja
SERVICECLASSNAME	SYSDEFAULTSUBCLASS	Ja
PARENTSERVICECLASSNAME	SYSDEFAULTUSERCLASS	Ja
COLLECTAGGACTDATA	N	Ja
COLLECTACTDATA	N	Ja
COLLECTACTPARTITION	C	Ja
COLLECTDEADLOCK	W	Ja
COLLECTLOCKTIMEOUT	W	Ja
COLLECTLOCKWAIT	N	Ja
LOCKWAITVALUE	0	Ja
COLLECTACTMETRICS	N	Ja
COLLECTUOWDATA	N	Ja
EXTERNALNAME	NULL	Nein
REMARKS	BLANK	Ja

Weitere Informationen finden Sie in SYSCAT.WORKLOADS.

Standardverwaltungsworkload (SYSDEFAULTADMWORKLOAD)

Diese Workload erlaubt es ACCESSCTRL-, DATAACCESS-, DBADM-, SECADM- oder WLMADM-Benutzern, jederzeit die Datenbank abzufragen und Verwaltungs- oder Überwachungstasks auszuführen, wird aber normalerweise in den folgenden Fällen verwendet:

- Die Workload, der der Administrator zugeordnet ist, darf nicht auf die Datenbank zugreifen (d. h., das Schlüsselwort DISALLOW DB ACCESS der Anweisung CREATE WORKLOAD oder ALTER WORKLOAD wurde für die Workload angegeben).
- Es wurde gegen einen Schwellenwert verstoßen, wodurch verhindert wurde, dass der Administrator an der Datenbank arbeiten konnte.

Die Workload SYSDEFAULTADMWORKLOAD unterscheidet sich folgendermaßen von anderen Workloads:

- Sie kann nicht gelöscht oder inaktiviert werden.
- Sie können für sie nicht DISALLOW DB ACCESS angeben.
- Für Vorkommen dieser Workload und die darin enthaltenen Aktivitäten gilt keiner der Schwellenwerte.
- Sie können diese Workload nur in der Servicesuperklasse SYSDEFAULTUSERCLASS ausführen. Weitere Informationen finden Sie in „Standardservicesuperklassen und -unterklassen“ auf Seite 74.
- Sie können eine Verbindung zu dieser Workload mit dem Befehl **SET WORKLOAD** über die CLP-Schnittstelle zuordnen oder durch Aufrufen der gespeicherten Prozedur WLM_SET_CLIENT_INFO (und Angabe von SYSDEFAULTADMWORKLOAD für den Parameter **client_workload**). Weitere Informationen finden Sie in „Maßnahmen zur Fehlerbehebung mit der Standardverwaltungsworkload“ auf Seite 29.

Sie können die Workload SYSDEFAULTADMWORKLOAD durch Abfragen der Tabelle SYSCAT.WORKLOADS anzeigen. Die folgende Tabelle zeigt die für die Workload SYSDEFAULTADMWORKLOAD in der Katalogsicht SYSCAT.WORKLOADS zurückgegebenen Spalten zusammen mit Werten und gibt an, ob diese Werte geändert werden können:

Tabelle 4. Eintrag SYSDEFAULTADMWORKLOAD in SYSCAT.WORKLOADS

Spalte	Wert	Modifizierbar mit der Anweisung ALTER WORKLOAD, wenn Sie die Berechtigung DBADM oder WLMADM haben (und SQLADM for COLLECT-Klauseln)
WORKLOADID	2	Nein
WORKLOADNAME	SYSDEFAULTADMWORKLOAD	Nein
EVALUATIONORDER	Letzte	Nein
CREATE_TIME	Zeitmarke der Datenbankerstellung	Nein
ALTER_TIME	Zeitmarke der letzten Aktualisierung der Workloaddefinition	Nein (aber der Datenserver ändert diese Spalte bei der Aktualisierung der Workloaddefinition)
ENABLED	Y	Nein
ALLOWACCESS	Y	Nein
SERVICECLASSNAME	SYSDEFAULTSUBCLASS	Nein
PARENTSERVICECLASSNAME	SYSDEFAULTUSERCLASS	Nein

Tabelle 4. Eintrag SYSDEFAULTADMWORKLOAD in SYSCAT.WORKLOADS (Forts.)

Spalte	Wert	Modifizierbar mit der Anweisung ALTER WORKLOAD, wenn Sie die Berechtigung DBADM oder WLMADM haben (und SQLADM for COLLECT-Klauseln)
COLLECTAGGACTDATA	N	Ja
COLLECTACTDATA	N	Ja
COLLECTACTPARTITION	C	Ja
COLLECTDEADLOCK	W	Ja
COLLECTLOCKTIMEOUT	W	Ja
COLLECTLOCKWAIT	N	Ja
LOCKWAITVALUE	0	Ja
COLLECTACTMETRICS	N	Ja
COLLECTUOWDATA	N	Ja
EXTERNALNAME	NULL	Nein
REMARKS	BLANK	Ja

Weitere Informationen finden Sie in SYSCAT.WORKLOADS.

Maßnahmen zur Fehlerbehebung mit der Standardverwaltungsworkload

Die Standardverwaltungsworkload SYSDEFAULTADMWORKLOAD ist eine spezielle von DB2 bereitgestellte Workloaddefinition, die keinem DB2-Schwellenwert unterliegt. Verwenden Sie diese Workload, um Maßnahmen zur Fehlerbehebung zu ergreifen, die nicht auf andere Weise durchgeführt werden können, wie z. B. das Ändern prohibitiver Schwellenwertdefinitionen, die die Ausführung aller Aktivitäten in einer Workload verhindern.

Vorbereitende Schritte

Mit dem Befehl **SET WORKLOAD** (oder der Prozedur WLM_SET_CLIENT_INFO) können Sie der Standardverwaltungsworkload SYSDEFAULTADMWORKLOAD eine Verbindung zuordnen.

Sie benötigen zwar keine Sonderberechtigung für die Verwendung des Befehls **SET WORKLOAD**, für die Zuordnung einer Verbindung zur Standardverwaltungsworkload ist jedoch die Berechtigung ACCESSCTRL, DATAACCESS, DBADM, SECADM oder WLMADM erforderlich. Andernfalls wird bei der Workloadzuordnung SQL0552N zurückgegeben.

Informationen zu diesem Vorgang

Da diese Workload nicht von Schwellenwerten betroffen ist, hat sie eingeschränkte Workload-Management-Steuerung und wird nicht für die Übergabe von alltäglicher Arbeit empfohlen.

Vorgehensweise

Setzen Sie den Befehl **SET WORKLOAD** wie folgt ab, um der Standardverwaltungsworkload eine Verbindung zuzuordnen:

SET WORKLOAD TO SYSDEFAULTADMWORKLOAD

Wann der Befehl wirksam wird, hängt vom Zeitpunkt ab, zu dem Sie ihn absetzen:

- Wenn Sie den Befehl SET WORKLOAD TO SYSDEFAULTADMWORKLOAD vor der Verbindung zur Datenbank absetzen, wird diese nach dem Herstellen der Verbindung am Anfang der ersten UOW (Unit of Work - Arbeitseinheit) der Workload SYSDEFAULTADMWORKLOAD zugeordnet.
- Wenn Sie den Befehl SET WORKLOAD TO SYSDEFAULTADMWORKLOAD am Anfang einer UOW absetzen, wird die Verbindung nach dem Herstellen der Verbindung zur Datenbank der Workload SYSDEFAULTADMWORKLOAD zugeordnet, wenn die erste Anforderung übergeben wird, die keine API sqlseti (Clientinformationen festlegen) ist.
- Wenn Sie den Befehl SET WORKLOAD TO SYSDEFAULTADMWORKLOAD in der Mitte einer UOW absetzen, nachdem eine Verbindung hergestellt wurde, wird die Verbindung am Anfang der nächsten UOW der Workload SYSDEFAULTADMWORKLOAD zugeordnet.

Wenn eine Verbindung der Workload SYSDEFAULTADMWORKLOAD zugeordnet ist, erfolgt die Neuuzuordnung der Workload am Anfang der nächsten UOW, wenn eine der folgenden Situationen auftritt:

- Sie entziehen dem Sitzungsbenutzer die Berechtigung SYSADM oder DBADM. In dieser Situation wird SQL0552N zurückgegeben.
- Sie setzen einen Befehl SET WORKLOAD TO AUTOMATIC ab. Dieser Befehl gibt an, dass die nächste UOW nicht der Workload SYSDEFAULTADMWORKLOAD zugeordnet werden soll und dass eine normale Workloadbewertung am Anfang der nächsten UOW ausgeführt werden soll. Weitere Informationen finden Sie in „Workloadzuordnung“ auf Seite 24.

Beispiel

Das folgende Beispiel zeigt, wie Sie mithilfe der Workload SYSDEFAULTADMWORKLOAD Maßnahmen zur Fehlerbehebung ergreifen können, wenn keine andere Maßnahme zur Fehlerbehebung möglich ist.

Wenn Sie einen stark prohibitiven Schwellenwert für gemeinsamen Zugriff erzeugen, sodass keine Aktivitäten ausgeführt werden können, da der Schwellenwert immer überschritten wird, kann derselbe Schwellenwert die Durchführung von Maßnahmen zur Fehlerbehebung verhindern. Damit Sie den prohibitiven Schwellenwert ändern können, müssen Sie zuerst die Workload so definieren, dass die Arbeit in der Standardverwaltungsworkload ausgeführt wird. Da in dieser Workload ausgeführte Aktivitäten keinen Schwellenwerten unterliegen, können Sie das Problem beheben und die Workload (für Ihre ID) wieder auf ihr Standardverhalten zurücksetzen.

Der Schwellenwert, der die Problemursache darstellt, wird versehentlich mit der folgenden Anweisung erzeugt. Der gemeinsame Zugriff sollte auf 100 eingestellt werden, wurde jedoch auf 0 eingestellt. Dieser Schwellenwert verhindert in der Praxis das Ausführen jeglicher Aktivität:

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

Anmerkung: Diese Anweisung soll nur darstellen, wie ein stark prohibitiver Schwellenwert erstellt werden könnte. Sie sollten diese Anweisung nicht absetzen.

Wenn Sie versuchen, auch nur eine einfache Anweisung SELECT auszuführen, wird ein Fehler zurückgegeben, da der gemeinsame Zugriff auf 0 gesetzt ist:

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

```
SQL4712N Der Schwellenwert "PROHIBITIVE" wurde überschritten. Ursachencode = "6".  
SQLSTATE=5U026
```

Bevor Sie Maßnahmen zur Fehlerbehebung ergreifen können, müssen Sie die Workload auf die Standardverwaltungsworkload einstellen:

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

Diese Anweisung kann nur von einem Benutzer mit der Berechtigung ACCESSCTRL, DATAACCESS, DBADM, SECADM oder WLMADM abgesetzt werden. Sie veranlasst, dass jede Verbindung der Workload SYSDEFAULTADMWORKLOAD zugeordnet wird, in der der prohibitive Schwellenwert nicht für Aktivitäten gilt.

Das Problem kann jetzt behoben werden, indem der Schwellenwert so geändert wird, dass Aktivitäten ausgeführt werden können:

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

Sobald das Problem behoben wurde, ändern Sie die Workload wieder zurück, so dass die Verbindung nicht mehr SYSDEFAULTADMWORKLOAD sondern der zuvor verwendeten Workload zugeordnet wird:

```
SET WORKLOAD TO AUTOMATIC
```

Dieselbe Anweisung SELECT, die Sie zuvor versucht haben, sollte jetzt erfolgreich ausgeführt werden:

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

...

```
DB20000I Der SQL-Befehl wurde erfolgreich ausgeführt.
```

Erstellen einer Workload

Mit der Anweisung CREATE WORKLOAD können Sie eine Workload zu den Katalogen hinzufügen.

Vorbereitende Schritte

Zum Erstellen einer Workload benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in den folgenden Abschnitten:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Anhang A, „Allgemeine Namenskonventionen“, auf Seite 541

Vorgehensweise

Gehen Sie wie folgt vor, um eine Workload zu erstellen:

1. Geben Sie mit der Anweisung CREATE WORKLOAD mindestens eine der folgenden Eigenschaften für die Workload an:
 - Den Namen der Workload.
 - Die Verbindungsattribute. Die eingehende Verbindungsanforderung muss Verbindungsattribute angeben, die den für die Workload definierten Attributen entsprechen.

ten entsprechen, damit eine Übereinstimmung erfolgen kann. Weitere Informationen finden Sie in „Identifizieren der Arbeit nach Ursprung mithilfe von Workloads“ auf Seite 19. Beachten Sie bei der Angabe der Verbindungsattribute, dass Werte mit OR und Attribute mit AND verknüpft werden: z. B. UserID (bob OR sue OR frank) AND Application (SAS).

- Einen Wert, der angibt, ob Vorkommen dieser Workload auf die Datenbank zugreifen dürfen. Standardmäßig ist der Datenbankzugriff für Vorkommen dieser Workload erlaubt.
- Einen Wert, der angibt, ob die Workload aktiviert oder inaktiviert ist. Standardmäßig ist die Workload aktiviert.
- Die Serviceklasse, in der Arbeit ausgeführt wird, die von Vorkommen dieser Workload übergeben werden. Standardeinstellung ist die Servicesuperklasse SYSDEFAULTUSERCLASS. Wenn Sie eine benutzerdefinierte Servicesuperklasse angeben, wird die von Workloadvorkommen übergebene Arbeit in der Serviceunterklasse SYSDEFAULTSUBCLASS der Servicesuperklasse ausgeführt.

Anmerkung: Sie können die Serviceunterklasse SYSDEFAULTSUBCLASS unter keiner Servicesuperklasse, auch nicht unter der Servicesuperklasse SYSDEFAULTUSERCLASS, angeben.

Wenn Sie eine Workload definiert haben, die auf eine Servicesuperklasse verweist, davon übergebene Arbeit jedoch nicht in der Standardserviceunterklasse SYSDEFAULTSUBCLASS ausgeführt werden soll, können Sie die Workloaddefinition so ändern, dass sie auf eine benutzerdefinierte Serviceunterklasse verweist, oder Sie können ein Arbeitsaktionsset verwenden, das für die Servicesuperklasse definiert wurde und mit dem einzelne Arbeitselemente einer anderen Serviceunterklasse zugeordnet werden. Weitere Informationen finden Sie in „Arbeitsaktionen und Arbeitsaktionssets“ auf Seite 102.

- Die Position der Workload relativ zu anderen Workloads bei der Zwischenspeicherung im Hauptspeicher. Die Position der neuen Workload bestimmt die Reihenfolge, in der sie bei der Workloadzuordnung bewertet wird. Standardmäßig wird die neue Workload an letzter Stelle positioniert, d. h., dass sie als letztes direkt vor der Standardbenutzerworkload bewertet wird. Weitere Informationen finden Sie in „Workloadzuordnung“ auf Seite 24.
- Die Stufe der Erfassung der Messdaten für die Überwachung von Aktivitäten, die von Verbindungen übergeben wurden, die dieser Workload zugeordnet sind. Die Standardeinstellung für die Erfassung der Aktivitätsmessdaten für eine Workload ist NONE. Beachten Sie, dass die effektive Erfassungseinstellung für Aktivitäten eine Kombination der Erfassungsstufe für Aktivitätsmessdaten und der Einstellung des Datenbankkonfigurationsparameters **mon_act_metrics** ist.
- Den Typ der zu erfassenden Aktivitätsinformationen. Standardmäßig werden keine Informationen zu Aktivitäten, die der Workload zugeordnet sind, an den Aktivitätsereignismonitor gesendet.
- Die zu erfassenden zusammengefassten Aktivitätsinformationen. Die zusammengefassten Aktivitätsinformationen für die Workload ändern sich erst nach dem Durchführen eines Commits für die Operation CREATE WORKLOAD.
- Die zu erfassenden Informationen zu Ereignissen mit Zeitlimitüberschreitung für Sperren. Standardmäßig werden Daten über ein Sperrenereignis an den Ereignismonitor für Sperren (sofern einer aktiv ist) gesendet, wenn das Sperrenereignis auftritt. Frühere Ereignisse mit Zeitlimitüberschreitung für Sperren werden jedoch nicht gesendet (WITHOUT HISTORY).

- Die zu erfassenden Deadlock-Informationen. Standardmäßig werden Daten über ein Deadlock-Ereignis an den Ereignismonitor für Sperren (sofern einer aktiv ist) gesendet, wenn das Deadlock-Ereignis auftritt. Frühere Deadlock-Ereignisse werden jedoch nicht gesendet (WITHOUT HISTORY).
 - Die zu erfassenden Informationen zum Wartestatus für Sperren. Wenn eine Sperre nicht innerhalb der festgelegten Wartezeit erhalten wird, werden standardmäßig keine Informationen zum Wartestatus für Sperren erfasst.
 - Die UOW-Informationen für jede dieser Workload zugeordnete Transaktion, die am Ende der UOW an den UOW-Ereignismonitor (sofern einer aktiv ist) gesendet werden. Standardmäßig werden keine UOW-Informationen gesendet.
 - Die Histogrammschablonen, die von der Workload als Schablone für ihre Histogramme verwenden werden sollen. Die angegebenen Histogrammschablonen werden in der Sicht SYSCAT.HISTOGRAMTEMPLATEUSE widergespiegelt. Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload zur Sicht SYSCAT.WORKLOADS hinzugefügt. Durch das Festschreiben der Änderung erfolgt die Workloadneubewertung am Anfang der nächsten UOW (Unit of Work - Arbeitseinheit) jeder Anwendung. Je nachdem, welche Workload ausgewählt wird, wird die Anwendung eventuell einer anderen Workload zugeordnet.

Nächste Schritte

Nachdem Sie eine Workload erstellt haben, müssen Sie unter Umständen mindestens einem Sitzungsbenutzer das Zugriffsrecht USAGE erteilen. (Sitzungsbenutzer mit der Berechtigung WLMADM oder DBADM haben ein implizites Zugriffsrecht für jede Workload.) Selbst wenn eine Verbindung eine exakte Übereinstimmung mit den Verbindungsattributen der Workload aufweist, berücksichtigt der Daten-server die Workload nicht bei der Workloadbewertung, wenn der Sitzungsbenutzer nicht das Zugriffsrecht USAGE für die Workload hat. Weitere Informationen finden Sie in „Erteilen des Zugriffsrechts USAGE für eine Workload“ auf Seite 39.

Ändern einer Workload

Eine Anweisung ALTER WORKLOAD ändert eine Workload in den Katalogen.

Vorbereitende Schritte

Zum Ändern einer Workload benötigen Sie die Berechtigung SQLADM, WLMADM oder DBADM. Soll eine andere Klausel als die Klausel COLLECT angegeben werden, muss die Berechtigungs-ID die Berechtigung WLMADM oder DBADM aufweisen.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Workload zu ändern:

1. Geben Sie mit der Anweisung ALTER WORKLOAD mindestens eine der folgenden Eigenschaften für die Workload an:
 - Die Verbindungsattribute. Sie können Verbindungsattribute zur Workloaddefinition hinzufügen oder daraus löschen, sofern es sich nicht um die Work-

load SYSDEFAULTUSERWORKLOAD oder SYSDEFAULTADMWORKLOAD handelt. Die eingehende Verbindungsanforderung muss Verbindungsattribute angeben, die den für die Workload definierten Attributen entsprechen, damit eine Übereinstimmung erfolgen kann. Weitere Informationen finden Sie in „Identifizieren der Arbeit nach Ursprung mithilfe von Workloads“ auf Seite 19. Zum Anzeigen der Verbindungsattribute für eine Workload fragen Sie die Sicht SYSCAT.WORKLOADCONNATTR ab.

- Einen Wert, der angibt, ob ein Vorkommen dieser Workload auf die Datenbank zugreifen darf. Standardmäßig ist der Datenbankzugriff für ein Vorkommen dieser Workload erlaubt. Sie können den Datenbankzugriff von der Workload SYSDEFAULTADMWORKLOAD nicht entfernen.
- Einen Wert, der angibt, ob die Workload aktiviert oder inaktiviert ist. Standardmäßig ist die Workload aktiviert. Sie können die Workload SYSDEFAULTUSERWORKLOAD oder SYSDEFAULTADMWORKLOAD nicht inaktivieren.
- Die Serviceklasse, unter der Vorkommen dieser Workload ausgeführt werden sollen. Standardeinstellung ist die Servicesuperklasse SYSDEFAULTUSERCLASS. Wenn Sie eine benutzerdefinierte Servicesuperklasse angeben, können Sie unter der Servicesuperklasse eine Serviceunterklasse angeben. Die Unterklasse SYSDEFAULTSUBCLASS kann unter keiner Servicesuperklasse, auch nicht unter der Servicesuperklasse SYSDEFAULTUSERCLASS, angegeben werden. Außerdem können Sie die Servicesuperklasse SYSDEFAULTSYSTEMCLASS oder SYSDEFAULTMAINTENANCECLASS nicht angeben.
- Die Position der Workload relativ zu anderen Workloads, womit die Reihenfolge festgelegt wird, in der die Workload bei der Workloadzuordnung bewertet wird. Die Position der Workload SYSDEFAULTUSERWORKLOAD oder SYSDEFAULTADMWORKLOAD kann nicht festgelegt werden. Weitere Informationen finden Sie in „Workloadzuordnung“ auf Seite 24.
- Den Typ der zu erfassenden Aktivitätsinformationen. Standardmäßig werden keine Informationen zu Aktivitäten, die der Workload zugeordnet sind, an den Aktivitätsereignismonitor gesendet.
- Die Stufe der Erfassung der Messdaten für die Überwachung von Aktivitäten, die von Verbindungen übergeben wurden, die dieser Workload zugeordnet sind. Beachten Sie, dass die effektive Erfassungseinstellung für Aktivitäten eine Kombination der Erfassungsstufe für Aktivitätsmessdaten und der Einstellung des Datenbankkonfigurationsparameters `mon_act_metrics` ist.
- Die zu erfassenden zusammengefassten Aktivitätsinformationen. Die zusammengefassten Aktivitätsinformationen für die Workload ändern sich erst nach dem Durchführen eines Commits für die Operation ALTER WORKLOAD.
- Die Informationen zu Ereignissen mit Zeitlimitüberschreitung für Sperren, die an den Ereignismonitor für Sperren (sofern einer aktiv ist) gesendet werden, wenn ein Sperrenereignis auftritt.
- Die Deadlock-Informationen, die an den Ereignismonitor für Sperren (sofern einer aktiv ist) gesendet werden, wenn ein Deadlock-Ereignis auftritt.
- Die zu erfassenden Informationen zum Wartestatus für Sperren.
- Die UOW-Informationen für jede dieser Workload zugeordnete Transaktion, die am Ende der UOW an den UOW-Ereignismonitor (sofern einer aktiv ist) gesendet werden.
- Die Histogrammschablonen, die von der Workload als Schablone für ihre Histogramme verwenden werden sollen. Die angegebenen Histogrammschablonen werden in der Sicht SYSCAT.HISTOGRAMTEMPLATEUSE widergespiegelt. Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht SYSCAT.WORKLOADS aktualisiert. Die festgeschriebene Änderung verursacht am Anfang der nächsten UOW (Unit of Work - Arbeitseinheit) jeder Anwendung eine Workloadneubewertung. Je nachdem, welche Workload ausgewählt wird, wird die Anwendung eventuell einer anderen Workload zugeordnet.

Nächste Schritte

Sie müssen unter Umständen mindestens einem Sitzungsbenutzer das Zugriffsrecht USAGE erteilen. (Sitzungsbenutzer mit der Berechtigung DBADM haben ein implizites Zugriffsrecht für jede Workload.) Selbst wenn eine Verbindung exakt mit den Verbindungsattributen der Workload übereinstimmt, ordnet der Datenserver die Verbindung der Workload nicht zu und erzeugt kein Vorkommen der Workload, wenn der Sitzungsbenutzer nicht das Zugriffsrecht USAGE für die Workload hat. Weitere Informationen finden Sie in „Erteilen des Zugriffsrechts USAGE für eine Workload“ auf Seite 39.

Aktivieren einer Workload

Der DB2-Datenserver vergleicht die angegebenen Verbindungsattribute für einen Workloadagenten mit den Verbindungsattributen der aktuellen Sitzung. Bei der Suche nach übereinstimmenden Workloads berücksichtigt der Datenserver keine inaktivierten Workloads.

Vorbereitende Schritte

Zum Ändern einer Workload benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Standardmäßig ist eine Workload bei der Erstellung aktiviert. Wenn Sie eine Workload als inaktiviert erstellen, müssen Sie sie aktivieren, damit der Datenserver sie bei der Workloadbewertung berücksichtigt.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Workload zu aktivieren:

1. Geben Sie die Workload an, die Sie aktivieren wollen. Sie können die Gruppe der inaktivierten Workloads durch Abfragen der Sicht SYSCAT.WORKLOADS anzeigen (siehe folgendes Beispiel):

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

2. Aktivieren Sie die inaktivierte Workload mit der Anweisung ALTER WORKLOAD:

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

Wenn die Anweisung ALTER WORKLOAD erfolgreich ist, wird die Definition für die Workload in den Datenbankkatalog geschrieben.

3. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht SYSCAT.WORKLOADS aktualisiert.

Ergebnisse

Die Aktivierung der Workload wird am Anfang der nächsten UOW (Unit of Work - Arbeitseinheit) wirksam. An diesem Punkt erfolgt die Workloadneubewertung und der Datenserver berücksichtigt die neu aktivierte Workload bei der Workloadneubewertung.

Inaktivieren einer Workload

Mit dieser Task können Sie verhindern, dass bestimmte Workloads bei der Workloadzuordnung berücksichtigt werden. Wenn Sie eine Workload inaktivieren, berücksichtigt der Datenserver diese bei der Suche nach einer übereinstimmenden Workload nicht. Stattdessen ordnet der Datenserver die UOW (Unit of Work - Arbeitseinheit) der nächsten übereinstimmenden Workload zu. Wenn keine benutzerdefinierte Workload übereinstimmt, wird die Arbeit der Standardworkload zugeordnet.

Vorbereitende Schritte

Zum Erstellen oder Ändern einer Workload benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Workload zu inaktivieren:

1. Verwenden Sie die Option `DISABLE` der Anweisung `ALTER WORKLOAD`, um die Workload zu inaktivieren:
`ALTER WORKLOAD...DISABLE`
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht `SYSCAT.WORKLOADS` aktualisiert.

Ergebnisse

Die Inaktivierung der Workload wird am Anfang der nächsten UOW wirksam. An diesem Punkt erfolgt die Workloadneubewertung und die Verbindung wird der nächsten Workload zugeordnet, die mit den Verbindungsattributen übereinstimmt und für die eine Berechtigung vorhanden ist.

Löschen einer Workload

Durch das Löschen einer Workload wird diese aus dem Datenbankkatalog entfernt.

Vorbereitende Schritte

Zum Löschen einer Workload benötigen Sie die Berechtigung `WLMADM` oder `DBADM`.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Workload zu löschen:

1. Inaktivieren Sie die Workload mit der Anweisung ALTER WORKLOAD. Weitere Informationen finden Sie in „Inaktivieren einer Workload“ auf Seite 36.
Durch die Inaktivierung der Workload wird verhindert, dass neue Vorkommen der Workload in der Datenbank ausgeführt werden können.
2. Stellen Sie mithilfe der Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES sicher, dass zurzeit keine Vorkommen dieser Workload ausgeführt werden. Weitere Informationen finden Sie in WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion).
Die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES gibt die Anwendungskennungen zurück, die den aktiven Workloadvorkommen entsprechen. Mit dem Befehl **FORCE APPLICATION** können Sie die Anwendungen anhand der Anwendungskennungen beenden.
3. Löschen Sie die Workload mit der Anweisung DROP WORKLOAD. Wenn Sie z. B. die Workload ACCTNG löschen wollen, geben Sie die folgende Anweisung an:

```
DROP WORKLOAD ACCTNG
```
4. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload aus der Sicht SYSCAT.WORKLOADS entfernt. Außerdem werden Berechtigungsinformationen für die Workload aus der Sicht SYSCAT.WORKLOADAUTH entfernt.

Erlauben des Datenbankzugriffs für Vorkommen einer Workload

Wenn Sie eine Workload haben, die nicht auf die Datenbank zugreifen darf, Sie jetzt aber die Ausführung von Vorkommen dieser Workload erlauben wollen, ändern Sie die Workload so, dass sie auf die Datenbank zugreifen darf. Nach dem Erstellen einer Workload ist diese standardmäßig berechtigt, auf die Datenbank zuzugreifen.

Vorbereitende Schritte

Sie benötigen die Berechtigung WLMADM oder DBADM, um eine Workload so zu ändern, dass sie auf eine Datenbank zugreifen darf.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Auch wenn Sie den Zugriff einer Workload auf die Datenbank verhindern, untersucht der Datenserver diese Workload bei der Workloadzuordnung. Alle Vorkommen dieser Workload werden jedoch mit einem Fehler zurückgewiesen.

Vorgehensweise

Gehen Sie wie folgt vor, um einer Workload den Zugriff auf die Datenbank zu erlauben:

1. Verwenden Sie die Option ALLOW DB ACCESS der Anweisung ALTER WORKLOAD, um der Workload den Zugriff auf die Datenbank zu erlauben.

Wenn Sie z. B. einer Workload namens WL1 den Datenbankzugriff erlauben wollen, geben Sie die folgende Anweisung an:

```
ALTER WORKLOAD WL1 ALLOW DB ACCESS
```

2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht SYSCAT.WORKLOADS aktualisiert.

Ergebnisse

Die Änderung einer Workload, mit der der Datenbankzugriff für ihre Vorkommen erlaubt wird, wird wirksam, wenn der Datenserver die nächste UOW (Unit of Work - Arbeitseinheit) für diese Workload analysiert. Wenn Sie z. B. DISALLOW DB ACCESS für Workload A angegeben haben und Sie nun die Workload mit ALLOW DB ACCESS ändern, dürfen neue Vorkommen von Workload A ausgeführt werden. Bisher wären alle Vorkommen von Workload A mit einem Fehler zurückgewiesen worden.

Verhindern des Datenbankzugriffs für Vorkommen einer Workload

Mit dieser Task können Sie steuern, welche Workloads auf die Datenbank zugreifen dürfen. Vor der Ausführung eines Workloadvorkommens prüft der Datenserver, ob die Workload berechtigt ist, auf die Datenbank zuzugreifen. Wenn Sie den Datenbankzugriff für das Workloadvorkommen verhindern, wird ein Fehler zurückgegeben, der angibt, dass das Workloadvorkommen zurückgewiesen wird.

Vorbereitende Schritte

Zum Verhindern des Datenbankzugriffs für eine Workload benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Das Verhindern des Datenbankzugriffs für ein Workloadvorkommen unterscheidet sich von der Inaktivierung einer Workload. Wenn Sie eine Workload inaktivieren, wird die Workloaddefinition nicht im Hauptspeicher zwischengespeichert und wird daher für die Workloadzuordnung nicht berücksichtigt.

Vorgehensweise

Gehen Sie wie folgt vor, um den Datenbankzugriff für eine Workload zu verhindern:

1. Verwenden Sie die Option DISALLOW DB ACCESS der Anweisung ALTER WORKLOAD (siehe folgendes Beispiel):

```
ALTER WORKLOAD workloadname DISALLOW DB ACCESS ...
```
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht SYSCAT.WORKLOADS aktualisiert.

Ergebnisse

Die Änderung einer Workload, mit der der Datenbankzugriff für ihre Vorkommen verhindert wird, wird bei bereits ausgeführten Workloadvorkommen am Anfang der nächsten UOW (Unit of Work - Arbeitseinheit) wirksam. Wenn Sie z. B. ALLOW DB ACCESS für die Workload A angeben und die Workload durch Angabe von DISALLOW DB ACCESS ändern, empfangen Vorkommen der Workload A, die bereits ausgeführt werden, am Anfang der nächsten UOW einen SQL-Fehler. Neue Vorkommen von Workload A werden zurückgewiesen.

Erteilen des Zugriffsrechts USAGE für eine Workload

Damit einer Workload eine Verbindung zugeordnet wird, muss der Sitzungsbenutzer das Zugriffsrecht USAGE für diese Workload haben. Benutzer mit der Berechtigung ACCESSCTRL, DATAACCESS, DBADM, SECADM oder WLMADM haben implizit das Zugriffsrecht USAGE für alle Workloads.

Vorbereitende Schritte

Sie benötigen die Berechtigung ACCESSCTRL, SECADM oder WLMADM, um die Anweisung GRANT USAGE ON WORKLOAD verwenden zu können.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Wenn der Datenserver eine Workload findet, die den Attributen einer eingehenden Verbindung entspricht, prüft der Datenserver, ob der Sitzungsbenutzer das Zugriffsrecht USAGE für diese Workload hat. Hat der Sitzungsbenutzer das Zugriffsrecht USAGE für diese Workload nicht, sucht der Datenserver nach der nächsten übereinstimmenden Workload. (Die Workloads, für die der Sitzungsbenutzer das Zugriffsrecht USAGE nicht hat, werden behandelt, als wären sie nicht vorhanden.) Über das Zugriffsrecht USAGE für eine Workload können Sie daher weiter steuern, welcher Workload unter den übereinstimmenden Workloads ein Benutzer, eine Gruppe oder eine Rolle zugeordnet werden sollte. Sie können z. B. mehrere Workloads mit denselben Verbindungsattributen definieren und das Zugriffsrecht USAGE für jede dieser Workloads nur bestimmten Benutzern, Gruppen oder Rollen erteilen. Weitere Informationen finden Sie in „Workloadzuordnung“ auf Seite 24.

Der Client kann die Clientbenutzer-ID, den Clientanwendungsnamen, den Clientworkstationnamen und die Clientabrechnungszeichenfolge (die einige der Verbindungsattribute darstellen, die für die Zuordnung einer Verbindung zu einer Workload verwendet werden) ohne Berechtigung festlegen. Daher können Sie über das Zugriffsrecht USAGE für Workloads auch steuern, welcher Sitzungsbenutzer die Berechtigung zur Verwendung einer Workload hat.

Sie können die Informationen zum Zugriffsrecht USAGE durch Abfragen der Sicht SYSCAT.WORKLOADAUTH anzeigen.

Wenn Sie eine Datenbank mit der Option RESTRICTIVE erstellen, wird PUBLIC das Zugriffsrecht USAGE für die Workload SYSDEFAULTUSERWORKLOAD bei der Datenbankerstellung nicht erteilt. Sie müssen Benutzern ohne die Berechtigung WLMADM oder DBADM das Zugriffsrecht USAGE für diese Workload explizit erteilen.

Wenn der Sitzungsbenutzer für keine der Workloads, einschließlich SYSDEFAULTUSERWORKLOAD, das Zugriffsrecht USAGE hat, wird SQL4707N zurückgegeben, wenn der Datenserver versucht, eine Workload mit der Datenbankverbindung zu verknüpfen.

Vorgehensweise

Gehen Sie wie folgt vor, um das Zugriffsrecht USAGE für eine Workload zu erteilen:

1. Verwenden Sie die Anweisung GRANT USAGE ON WORKLOAD. Sie können das Zugriffsrecht USAGE bestimmten Benutzern, Gruppen, Rollen oder dem Benutzer PUBLIC erteilen. Wenn Sie z. B. das Zugriffsrecht USAGE für die Workload ACCOUNTS der Gruppe CPA erteilen wollen, müssen Sie die folgende Anweisung absetzen:

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

Für die Workload SYSDEFAULTADMWORKLOAD können Sie das Zugriffsrecht USAGE nicht erteilen. Die Workload SYSDEFAULTADMWORKLOAD kann nur von ACCESSCTRL-, DATAACCESS-, DBADM-, SECADM- oder WLMADM-Benutzern verwendet werden, die den Befehl SET WORKLOAD TO SYSDEFAULTADMWORKLOAD absetzen.

2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Sicht SYSCAT.WORKLOADAUTH aktualisiert. Bis zur Festschreibung der Anweisung GRANT berücksichtigt der Datenserver die Workload nicht für die Zuordnung für die neu autorisierten Benutzer, Gruppen oder Rollen.

Widerrufen des Zugriffsrechts USAGE für eine Workload

Mit der Anweisung REVOKE USAGE ON WORKLOAD können Sie das Zugriffsrecht USAGE für eine Workload widerrufen.

Vorbereitende Schritte

Sie benötigen die Berechtigung ACCESSCTRL, SECADM oder WLMADM, um die Anweisung REVOKE USAGE ON WORKLOAD verwenden zu können.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Für die Workload SYSDEFAULTADMWORKLOAD können Sie das Zugriffsrecht USAGE nicht explizit widerrufen. Nur ACCESSCTRL-, DATAACCESS-, DBADM-, SECADM- oder WLMADM-Benutzer, die den Befehl SET WORKLOAD TO SYSDEFAULTADMWORKLOAD absetzen, können diese Workload verwenden. Daher funktioniert die Anweisung REVOKE USAGE ON WORKLOAD nicht für SYSDEFAULTADMWORKLOAD.

Vorgehensweise

Gehen Sie wie folgt vor, um das Zugriffsrecht USAGE für eine Workload zu widerrufen:

1. Verwenden Sie die Anweisung REVOKE USAGE ON WORKLOAD. Sie können das Zugriffsrecht USAGE für bestimmte Benutzer, Gruppen, Rollen oder den Benutzer PUBLIC widerrufen. Wenn Sie z. B. das Zugriffsrecht USAGE für die Workload ACCOUNTS des Benutzers PUBLIC widerrufen wollen, müssen Sie die folgende Anweisung absetzen:

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

2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Sicht SYSCAT.WORKLOADAUTH aktualisiert. Bis zum Durchführen eines Commits für die Anweisung REVOKE berücksichtigt der Datenserver die Workload bei der Workloadzuordnung.

Beispiel: Workloadzuordnung

Am Anfang der ersten UOW (Unit of Work - Arbeitseinheit) nach dem Herstellen der Datenbankverbindung ordnet der Datenserver die Verbindung einer Workload zu, indem er die Verbindungsattribute jeder aktivierten Workload bewertet.

Die Workloadneubewertung erfolgt am Anfang jeder UOW, falls der Wert eines Verbindungsattributs oder der Workloaddefinition selbst sich innerhalb der UOW ändert.

Die folgende Abbildung zeigt eine Workloadzuordnung. Benutzer in der Gruppe Marketing, die Abfragen über AppA übergeben, werden der Workload APPAQUERIES zugeordnet. Obwohl PAYROLL vor APPAQUERIES positioniert ist, werden sie nicht der Workload PAYROLL zugeordnet, da die Definition der Workload PAYROLL für den Schlüsselwert SESSION_USER GROUP den Wert Finance definiert. Benutzer in der Gruppe Finance, die Abfragen über AppA übergeben, werden der Workload FINANCE zugeordnet. Sie werden nicht der Workload PAYROLL zugeordnet, obwohl diese genauer ist und in ihrer Definition sowohl AppA als auch Finance angibt, da die Workload FINANCE vor der Workload PAYROLL positioniert ist. Benutzer in der Gruppe Marketing, die Abfragen über AppB übergeben, werden der Workload SYSDEFAULTUSERWORKLOAD zugeordnet, da keines der Verbindungsattribute, die in den Definitionen der Workloads FINANCE, PAYROLL oder APPAQUERIES angegeben sind, der Anwendung AppB oder der Gruppe Marketing entspricht.

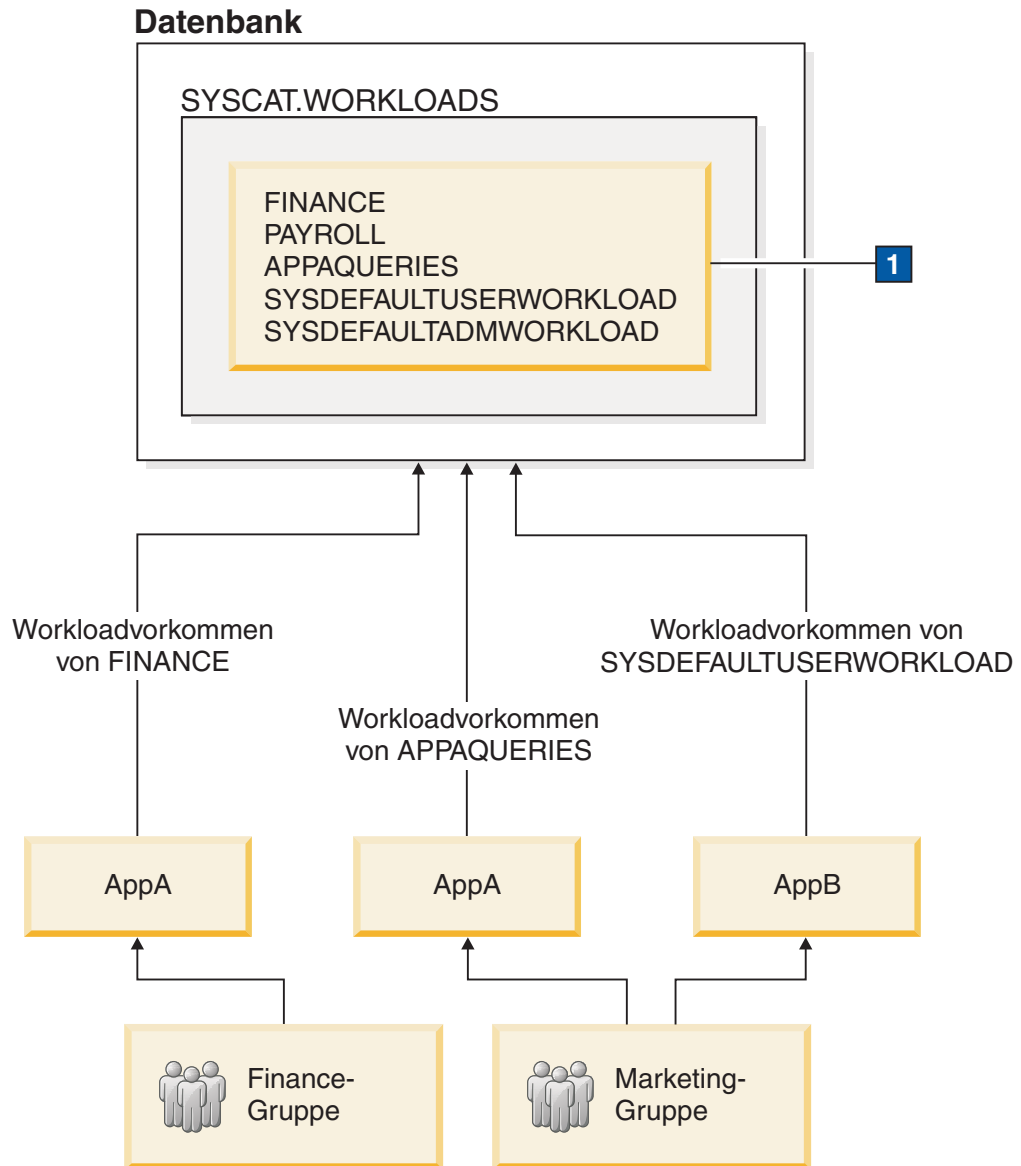


Abbildung 5. Beispiel einer Workloadzuordnung

1 In der Abbildung oben lauten die Anweisungen CREATE WORKLOAD folgendermaßen:

```
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
```

Viel Anwendungsserver stellen die Verbindungen für alle von ihnen verarbeiteten Clientanforderungen mit denselben Clientinformationen her und geben dieselben Clientinformationen weiter. Einige Produkte, beispielsweise WebSphere und Cognos, stellen eine Funktion für das Senden eindeutiger Informationen zu den einzelnen Anforderungen über die Clientinformationsfelder im Pushdown-Modus bereit.

Diese Informationen geben die Endbenutzeranforderung innerhalb von DB2 eindeutig an. Bei den meisten anderen Produkten kann der Anwendungsserver auf andere Weise so angepasst werden, dass vor dem Beginn der Verarbeitung einer Endbenutzeranforderung eindeutige Clientinformationen an DB2 gesendet werden.

Durch die Angabe eindeutiger Clientattribute über den Anwendungsserver wird die Sonderbehandlung von Anforderungen in DB2 und die Zuordnung von Anforderungen unterschiedlicher Clients zu unterschiedlichen Workloads (und zu unterschiedlichen Serviceklassen) aktiviert.

Die folgende Abbildung zeigt ein Beispiel einer dreischichtigen Umgebung, in der Abfragen von unterschiedlichen Benutzeranwendungen (marketing.exe, auditing.exe und reporting.exe) über einen Anwendungsserver übergeben werden, der eine Verbindung zur Datenbank mit dem Sitzungsbenutzer APPUSER herstellt. Drei Workloads sind definiert: eine für Abfragen, die von marketing.exe übergeben werden, eine für Abfragen, die von reporting.exe übergeben werden, und eine für die restlichen Abfragen. Wie in der Abbildung dargestellt, ruft der Anwendungsserver zum Zuordnen von Abfragen von marketing.exe die API sqleseti auf, um den Wert des Sonderregisters CURRENT CLIENT_APPLNAME auf marketing.exe einzustellen. Analog dazu ruft der Server zum Zuordnen von Abfragen von reporting.exe zur Workload REPORTING die API sqleseti auf, um den Wert des Sonderregisters CURRENT CLIENT_APPLNAME auf reporting.exe einzustellen. Beachten Sie in der Abbildung, dass keine erneute Workloadzuordnung erfolgt, wenn der Server die API sqleseti aufruft, um das Sonderregister CURRENT CLIENT_USERID auf Lidia einzustellen (ohne jegliche andere Änderungen, d. h., der Clientanwendungsname ist noch auf reporting.exe eingestellt). Dies liegt daran, dass es keine Workload gibt, die spezifisch mit der Einstellung Lidia für CURRENT CLIENT_USERID definiert ist.

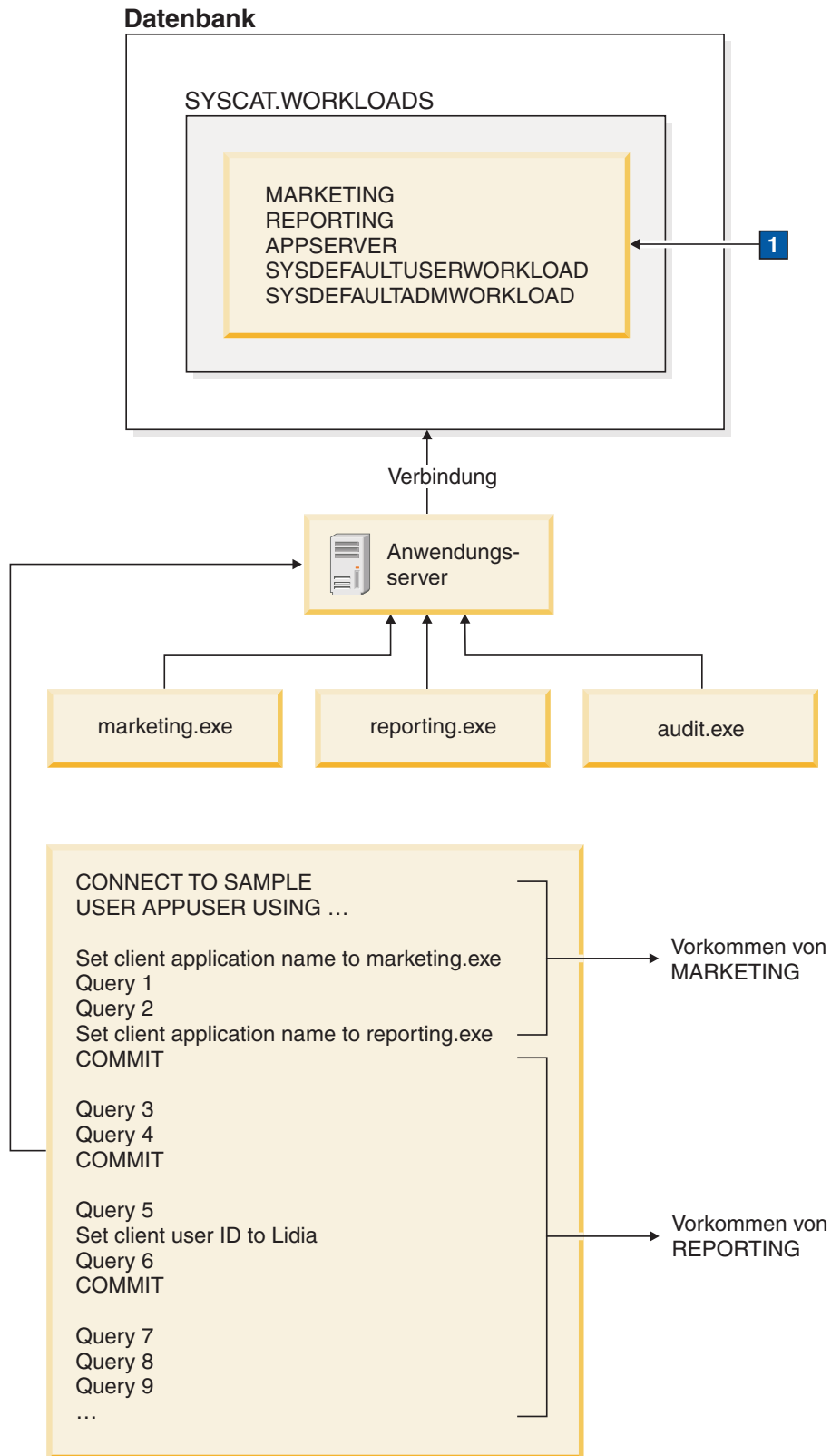


Abbildung 6. Beispiel einer Workloadzuordnung in einer dreischichtigen Umgebung

Die folgenden Anweisungen werden verwendet, um die im Feld **1** in der Abbildung oben angegebenen Workloads zu definieren:

```
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
```

Beispiel: Workloadzuordnung bei Workloadattributen mit Einzelwerten

Das Beispiel in diesem Abschnitt zeigt, wie der Datenserver Workloads zuordnet. In diesem Beispiel ist für jedes Workloadverbindungsattribut nur ein Wert angegeben.

Nehmen wir an, dass im Katalog die folgenden Workloads vorhanden sind:

Tabelle 5. Workloads im Katalog

Bewertungsreihenfolge	Workloadname	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			Audit Group

Nehmen wir an, dass eine Datenbankverbindung mit den folgenden Attributen hergestellt wurde:

Tabelle 6. Datenbankverbindungsattribute

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	Business account

Bei der Übergabe der ersten UOW (Unit of Work - Arbeitseinheit) prüft der Datenserver jede Workload im Katalog, beginnend mit der ersten Workload in der Liste, und verarbeitet die Workloads in aufsteigender Reihenfolge, bis er eine Workload mit übereinstimmenden Attributen findet. Wenn eine übereinstimmende Workload gefunden wurde, wird die UOW unter einem Vorkommen dieser Workload ausgeführt. Bei der Ermittlung, welcher Workload die Verbindung zugeordnet werden soll, vergleicht der Datenserver die Verbindungsattribute in einer definierten Reihenfolge.

Der Datenserver prüft zuerst die Workload REPORTS auf Übereinstimmung. Die Workload REPORTS ist die erste Workload in der Liste.

Tabelle 7. Workload REPORTS im Katalog

Bewertungsreihenfolge	Workloadname	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								

Der Datenserver prüft die Verbindungsattribute in der folgenden definierten Reihenfolge:

1. APPLNAME. Der Wert von APPLNAME, AppA, für die Datenbankverbindung entspricht dem Wert von APPLNAME für die Workload REPORTS.
2. SYSTEM_USER, das in der Workloaddefinition nicht angegeben ist. Jeder Wert (einschließlich eines Nullwerts) wird als Übereinstimmung betrachtet.
3. SESSION_USER, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.
4. SESSION_USER GROUP, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.
5. SESSION_USER ROLE, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.
6. CURRENT_CLIENT_USERID, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.
7. CURRENT_CLIENT_APPLNAME, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.
8. CURRENT_CLIENT_WRKSTNNAME, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.
9. CURRENT_CLIENT_ACCTNG, das in der Workloaddefinition nicht angegeben ist. Jeder Wert wird als Übereinstimmung betrachtet.

Aufgrund der expliziten und impliziten Übereinstimmungen zwischen den Verbindungsattributen der Workload REPORTS und den von der Verbindung übergebenen Informationen wählt der Datenserver in dieser Situation die Workload REPORTS als potenzielle Übereinstimmung aus. Nach der Auswahl einer Workload prüft der Datenserver dann, ob der Sitzungsbenutzer das Zugriffsrecht USAGE für die Workload hat. Nehmen wir an, dass der Sitzungsbenutzer TIM das Zugriffsrecht USAGE für die Workload REPORTS hat. Diese Workload wird dann für die Verbindung verwendet. Wenn jedoch TIM nicht das Zugriffsrecht USAGE für die Workload REPORTS hat, fährt der Datenserver fort, indem er die Workload INVENTORYREPORT auf eine Übereinstimmung überprüft.

Angenommen, TIM soll der Workload EXPENSEREPORT zugeordnet werden, da für diese Workload zusätzliche Verbindungsattribute angegeben sind. In dieser Situation würden Sie die Bewertungsreihenfolge der Workloads ändern, sodass EXPENSEREPORT vor REPORTS in der Workloadliste steht:

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

Mit der folgenden SQL-Anweisung könnten Sie dasselbe Ergebnis erzielen:

```
ALTER WORKLOAD EXPENSEREPORT BEFORE REPORTS
```

Damit die Anweisung ALTER WORKLOAD wirksam wird, müssen Sie sofort nach der Anweisung ALTER WORKLOAD eine Anweisung COMMIT absetzen. Die Anweisung ALTER WORKLOAD hat die folgende Auswirkung auf den Katalog:

Tabelle 8. Workloads im Katalog nach der Neupositionierung der Workload EXPENSEREPORT

Bewertungsreihenfolge	Workloadname	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			Audit Group

Wenn TIM noch nicht das Zugriffsrecht USAGE für die Workload EXPENSEREPORT hat, müssen Sie die folgenden Anweisungen absetzen (die Anweisung COMMIT stellt sicher, dass die Anweisung GRANT wirksam wird):

```
GRANT USAGE ON WORKLOAD EXPENSEREPORT TO USER TIM
COMMIT
```

Am Anfang der nächsten UOW erfolgt die erneute Workloadzuordnung und der Datenserver ordnet die Verbindung von TIM der Workload EXPENSEREPORT zu. Außerdem werden auch neue UOWs, die von anderen Verbindungen mit denselben Attributen übergeben werden, der Workload EXPENSEREPORT zugeordnet.

Beispiel: Workloadzuordnung für eine UOW, wenn mehrere Workloads vorhanden sind

Das Beispiel in diesem Abschnitt zeigt, wie der Datenserver die Workloadbewertung durchführt, um die Verbindung einer vorhandenen Workload zuzuordnen.

Nehmen wir an, dass die folgenden Workloads im Katalog definiert sind:

Tabelle 9. Workloads im Katalog

Bewertungsreihenfolge	Workloadname	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			Audit Group

Nehmen wir an, dass eine Datenbankverbindung mit den folgenden Attributen hergestellt wird:

Tabelle 10. Datenbankverbindungsattribute

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	Audit group

Bei der Übergabe der ersten UOW (Unit of Work - Arbeitseinheit) prüft der Datenserver jede Workload im Katalog in aufsteigender Bewertungsreihenfolge und stoppt, wenn er eine Workload findet, deren Verbindungsattribute den Verbindungsattributen entsprechen, die von der Verbindung angegeben werden. Beim Prüfen der Workloads vergleicht der Datenserver die Verbindungsattribute in einer definierten Reihenfolge.

Zuerst prüft der Datenserver die Workload EXPENSEREPORT:

Tabelle 11. Workload EXPENSEREPORT im Katalog

Bewertungsreihenfolge	Workloadname	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				

Da das Attribut APPLNAME in der Workloaddefinition AppB lautet, das von der Verbindung übergebene Attribut APPLNAME jedoch AppA lautet, ist keine Übereinstimmung möglich. Der Datenserver fährt mit der Workload REPORTS fort, dem zweiten Eintrag in der Liste:

Tabelle 12. Workload REPORTS im Katalog

Bewertungsreihenfolge	Workloadname	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
2	REPORTS	AppB								

Auch hier lautet das Attribut APPLNAME in der Workloaddefinition AppB und stimmt nicht mit AppA überein. Der Datenserver fährt mit der dritten Workload in der Liste fort, der Workload INVENTORYREPORT:

Tabelle 13. Workload INVENTORYREPORT im Katalog

Bewertungsreihenfolge	Workloadname	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
3	INVENTORYREPORT	AppA	LYNN		ACCOUNTING	TELEMKTR				

Der Datenserver prüft auf eine Übereinstimmung zwischen den übergebenen Verbindungsattributen und der Workload INVENTORYREPORT. Die Attribute werden in der folgenden Reihenfolge geprüft:

1. APPLNAME. Sowohl die Workloaddefinition als auch die Verbindung haben den Wert AppA. Es gibt also eine Übereinstimmung.
2. SYSTEM_USER. Sowohl die Workloaddefinition als auch die Verbindung haben den Wert LYNN. Es gibt also eine Übereinstimmung.
3. SESSION_USER. Die Verbindung übergab den Wert LYNN. Da das Attribut SESSION_USER für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert, einschließlich eines Nullwerts, als Übereinstimmung.
4. SESSION_USER GROUP. Sowohl die Workloaddefinition als auch die Verbindung haben den Wert ACCOUNTING. Es gibt also eine Übereinstimmung.
5. SESSION_USER ROLE. Die Workloaddefinition gibt den Wert TELEMKTR an. Die Verbindung übergab jedoch die Werte FINANALYST und SALESREP. Für dieses Attribut gibt es keine Übereinstimmung.

Der Datenserver versucht nicht weiter, die Workload INVENTORYREPORT mit den Verbindungsattributen abzugleichen, und fährt mit der vierten Workload in der Liste fort, der Workload SALESREPORT:

Tabelle 14. Workload SALESREPORT im Katalog

Bewertungsreihenfolge	Workloadname	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
4	SALESREPORT	AppC	KATE	KATE		SALESREP				

Da das Attribut APPLNAME der Workloaddefinition von SALESREPORT den Wert AppC hat, gibt es keine Übereinstimmung mit der Verbindung (die den Wert AppA für APPLNAME übergab). Der Datenserver fährt dann mit der fünften Workload in der Liste fort, der Workload AUDITREPORT:

Tabelle 15. Workload AUDITREPORT im Katalog

Bewertungsreihenfolge	Workloadname	APPLNAME	SYSTEM_USER	SESSION_USER	SESSION_USER GROUP	SESSION_USER ROLE	CURRENT_CLIENT_USERID	CURRENT_CLIENT_APPLNAME	CURRENT_CLIENT_WRKSTNNAME	CURRENT_CLIENT_ACCTNG
5	AUDITREPORT	AppA			ACCOUNTING	FINANALYST				

Der Datenserver vergleicht die Attribute der Workload AUDITREPORT und der Verbindung in der definierten Reihenfolge:

1. APPLNAME. Sowohl die Workloaddefinition als auch die Verbindung haben den Wert AppA. Es gibt also eine Übereinstimmung.

2. SYSTEM_USER. Die Verbindung übergab den Wert LYNN. Da das Attribut SYSTEM_USER für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
3. SESSION_USER. Die Verbindung übergab den Wert LYNN. Da das Attribut SESSION_USER für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
4. SESSION_USER GROUP. Sowohl die Workload als auch die Verbindung haben den Wert ACCOUNTING für dieses Attribut. Es gibt also eine Übereinstimmung.
5. SESSION_USER ROLE. Sowohl die Workload als auch die Verbindung haben den Wert FINANALYST für dieses Attribut. Es gibt also eine Übereinstimmung.
6. CURRENT CLIENT_USERID. Da das Attribut CURRENT CLIENT_USERID für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
7. CURRENT CLIENT_APPLNAME. Da das Attribut CURRENT CLIENT_APPLNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
8. CURRENT CLIENT_WRKSTNNAME. Da das Attribut CURRENT CLIENT_WRKSTNNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
9. CURRENT CLIENT_ACCTNG. Da das Attribut CURRENT CLIENT_ACCTNG für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.

Nachdem der Datenserver alle Verbindungsattribute verarbeitet und eine übereinstimmende Workload gefunden hat, prüft er, ob der Sitzungsbenutzer das Zugriffsrecht USAGE für die Workload hat. Nehmen wir an, dass LYNN das Zugriffsrecht USAGE für die Workload AUDITREPORT nicht besitzt. In dieser Situation wird diese Workload der Verbindung nicht zugeordnet, obwohl alle Verbindungsattribute übereinstimmen. Der Datenserver fährt mit der sechsten Workload in der Bewertungsliste fort, der Workload AUDITRESULT:

Tabelle 16. Workload AUDITRESULT im Katalog

Bewertungsreihenfolge	Workloadname	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			Audit Group

Der Datenserver vergleicht die Attribute der Workload AUDITRESULT und der Verbindung in der definierten Reihenfolge:

1. APPLNAME. Da das Attribut APPLNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
2. SYSTEM_USER. Da das Attribut SYSTEM_USER für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
3. SESSION_USER. Sowohl die Workload als auch die Verbindung haben den Wert LYNN für dieses Attribut. Es gibt also eine Übereinstimmung.
4. SESSION_USER GROUP. Da das Attribut SESSION_USER GROUP für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
5. SESSION_USER ROLE. Da das Attribut SESSION_USER ROLE für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
6. CURRENT CLIENT_USERID. Sowohl die Workload als auch die Verbindung haben den Wert LYNN für dieses Attribut. Es gibt also eine Übereinstimmung.

7. CURRENT CLIENT_APPLNAME. Da das Attribut CURRENT CLIENT_APPLNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
8. CURRENT CLIENT_WRKSTNNAME. Da das Attribut CURRENT CLIENT_WRKSTNNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
9. CURRENT CLIENT_ACCTNG. Sowohl die Workload als auch die Verbindung haben den Wert Audit Group für dieses Attribut. Es gibt also eine Übereinstimmung.

Nachdem der Datenserver alle Verbindungsattribute verarbeitet und eine übereinstimmende Workload gefunden hat, prüft er, ob der Sitzungsbenutzer das Zugriffsrecht USAGE für die Workload hat. Nehmen wir in dieser Situation an, dass der Sitzungsbenutzer LYNN das Zugriffsrecht USAGE für die Workload AUDITRESULT besitzt. Da alle Verbindungsattribute übereinstimmen und der Sitzungsbenutzer das Zugriffsrecht USAGE hat, wird die Verbindung der Workload AUDITRESULT zugeordnet.

Beispiel: Workloadzuordnung bei Workloadattributen mit mehreren Werten

Das Beispiel in diesem Abschnitt zeigt, wie der Datenserver Workloads zuordnet. In diesem Beispiel lassen einige der Workloaddefinitionen mehrere Werte für ein Verbindungsattribut zu.

Nehmen wir an, dass die folgenden Workloads im Katalog definiert sind:

Tabelle 17. Workloads im Katalog

Bewertungsreihenfolge	Workloadname	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				

Nehmen wir an, dass eine Datenbankverbindung mit den folgenden Attributen hergestellt wurde:

Tabelle 18. Datenbankverbindungsattribute

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	Business Account

Bei der Übergabe der ersten UOW (Unit of Work - Arbeitseinheit) prüft der Datenserver jede Workload im Katalog in aufsteigender Bewertungsreihenfolge und stoppt, wenn er eine Workload findet, deren Verbindungsattribute den Verbindungsattributen entsprechen, die von der Verbindung angegeben werden. Beim Prüfen der Workloads vergleicht der Datenserver die Verbindungsattribute in einer definierten Reihenfolge.

Zuerst prüft der Datenserver die Workload ITEMINQ:

Tabelle 19. Workload ITEMINQ im Katalog

Bewertungsreihenfolge	Workloadname	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					

Der Datenserver prüft auf eine Übereinstimmung zwischen den übergebenen Verbindungsattributen und der Workload ITEMINQ. Die Attribute werden in der folgenden Reihenfolge geprüft:

1. APPLNAME. Da das Attribut APPLNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert, einschließlich eines Nullwerts, als Übereinstimmung.
2. SYSTEM_USER. Die Verbindung übergab den Wert LINDA. Die Workload ITEMNO hat jedoch die Werte KYLE und GEORGE. Für dieses Attribut gibt es keine Übereinstimmung.

Der Datenserver versucht nicht weiter, die Workload ITEMNO mit der Verbindung abzugleichen, und fährt mit der zweiten Workload in der Liste fort, der Workload DAILYTRANSREPORT:

Tabelle 20. Workload DAILYTRANSREPORT im Katalog

Bewertungsreihenfolge	Workloadname	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					

Der Datenserver vergleicht die Attribute der Workload DAILYTRANSREPORT und der Verbindung in definierter Reihenfolge:

1. APPLNAME. Sowohl die Workloaddefinition als auch die Verbindung haben den Wert AppC. Es gibt also eine Übereinstimmung.
2. SYSTEM_USER. Da das Attribut SYSTEM_USER für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert, einschließlich eines Nullwerts, als Übereinstimmung.
3. SESSION_USER. Der von der Verbindung übergebene Wert SESSION_USER lautet KYLE. Dies ist eine Übereinstimmung mit einem der Werte der Workload SESSION_USER. Hätte die Verbindung CAROL übergeben, wäre auch dies eine Übereinstimmung, da sowohl KYLE als auch CAROL als Teil der Workloaddefinition von DAILYTRANSREPORT angegeben sind.
4. SESSION_USER GROUP. Der von der Verbindung übergebene Wert SESSION_USER GROUP lautet SALES. Dieser Wert entspricht dem Wert SALES, der für das Attribut der Workload SESSION_USER GROUP angegeben ist. Hätte die Verbindung ACCOUNTING übergeben, wäre auch dies eine Übereinstimmung, da sowohl SALES als auch ACCOUNTING in der Workloaddefinition angegeben sind.
5. SESSION_USER ROLE. Da das Attribut SESSION_USER ROLE für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
6. CURRENT CLIENT_USERID. Da das Attribut CURRENT CLIENT_USERID für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
7. CURRENT CLIENT_APPLNAME. Da das Attribut CURRENT CLIENT_APPLNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
8. CURRENT CLIENT_WRKSTNNAME. Da das Attribut CURRENT CLIENT_WRKSTNNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.
9. CURRENT CLIENT_ACCTNG. Da das Attribut CURRENT CLIENT_WRKSTNNAME für die Workload nicht festgelegt ist, gilt jeder von der Verbindung übergebene Wert als Übereinstimmung.

Nachdem der Datenserver alle Verbindungsattribute verarbeitet und eine übereinstimmende Workload für die Verbindung gefunden hat, prüft er, ob der Sitzungsbenuer das Zugriffsrecht USAGE für die Workload hat. Nehmen wir in dieser Situation an, dass der Sitzungsbenuer KYLE das Zugriffsrecht USAGE für die Workload DAILYTRANSREPORT hat. Da alle Verbindungsattribute übereinstimmen und der Sitzungsbenuer das Zugriffsrecht USAGE hat, wird die Verbindung der Workload DAILYTRANSREPORT zugeordnet.

Identifizieren der Arbeit nach Typ, Aufwand oder Datenzugriff mithilfe von Arbeitsklassen

Zusätzlich zur Verwendung von Verbindungsattributen, die sich auf den Ursprung der Aktivitäten mit Workloads beziehen, können Sie Aktivitäten basierend auf dem Typ, Aufwand oder Datenzugriff identifizieren, indem Sie ein Arbeitsklassenset mit einer Arbeitsklasse erstellen.

Eine Arbeitsklasse ist eine Methode zur Kategorisierung einzelner Datenbankaktivitäten basierend auf Attributen der Aktivitäten. Wenn für eine Arbeitsklasse eine Arbeitsaktion definiert ist, wird die Arbeitsaktion auf die Arbeitsklasse angewendet und bestimmt, wie die Aktivitäten in der Arbeitsklasse verwaltet werden. Weitere Informationen finden Sie in „Anwenden von Steuerelementen auf Typen von Aktivitäten mit Arbeitsaktionssets“ auf Seite 99.

Die folgende Tabelle zeigt die Typschlüsselwörter, die für Arbeitsklassen verfügbar sind, sowie die SQL-Anweisungen, die den verschiedenen Schlüsselwörtern entsprechen. Mit Ausnahme des Dienstprogramms LOAD werden alle in der folgenden Tabelle aufgeführten Anweisungen unmittelbar vor der Ausführung bei der Verarbeitung einer Anforderung EXECUTE, EXECUTE IMMEDIATE oder OPEN abgefangen. Wenn das Dienstprogramm LOAD von einem Client abgesetzt wird, setzt es eventuell Anforderungen vor dem Start der eigenen Ladeoperation auf dem Datenserver ab.

Tabelle 21. Arbeitstypen

Arbeitstypschlüsselwort	Gültige SQL-Anweisungen
READ, einschließlich SET-Anweisungen mit eingebettetem READ-SQL	<ul style="list-style-type: none"> Alle SELECT-Anweisungen (SELECT INTO, VALUES INTO, FULL SELECT) Ausnahme: SELECT-Anweisungen mit DELETE, INSERT oder UPDATE sind nicht eingeschlossen. Alle XQuery-Anweisungen
WRITE, einschließlich SET-Anweisungen mit eingebettetem WRITE-SQL	<ul style="list-style-type: none"> Alle UPDATE-Anweisungen (gesucht, positioniert) Alle DELETE-Anweisungen (gesucht, positioniert) Alle INSERT-Anweisungen (values, Subselect) Alle MERGE-Anweisungen Alle SELECT-Anweisungen mit einer Anweisung DELETE, INSERT oder UPDATE

Tabelle 21. Arbeitstypen (Forts.)

Arbeitstypschlüsselwort	Gültige SQL-Anweisungen
CALL	<p>Anweisung CALL</p> <p>Die Anweisung CALL wird nur unter den Arbeitsklassentypen CALL und ALL klassifiziert.</p> <p>Anmerkung: Sowohl anonyme Blöcke als auch autonome Routinen werden als CALL-Anweisungen klassifiziert.</p>
DML, einschließlich SET-Anweisungen mit eingebettetem READ- oder WRITE-SQL	<p>Alle Anweisungen, die unter den Arbeitsklassentypen READ und WRITE klassifiziert werden.</p>
DDL	<ul style="list-style-type: none"> • Alle ALTER-Anweisungen • Alle CREATE-Anweisungen • Anweisung COMMENT • Anweisung DECLARE GLOBAL TEMPORARY TABLE • Anweisung DROP • Anweisung FLUSH PACKAGE CACHE • Alle GRANT-Anweisungen • REFRESH TABLE • Alle RENAME-Anweisungen • Alle REVOKE-Anweisungen • Anweisung SET INTEGRITY
LOAD	<p>Dienstprogramm LOAD</p> <p>Das Dienstprogramm LOAD wird nur unter den Arbeitsklassentypen LOAD und ALL klassifiziert.</p>
ALL	<p>Arbeitstypen, die durch alle vorangehenden Schlüsselwörter dargestellt werden.</p> <p>Anmerkung: Wenn die Aktion ein Schwellenwert ist, hängt die Datenbankaktivität, auf die der Schwellenwert angewendet wird, vom Typ des Schwellenwerts ab. Wenn der Schwellenwerttyp z. B. ESTIMATEDSQLCOST ist, sind nur DML-Aktivitäten mit einem geschätzten Aufwand (in Timeron) vom Schwellenwert betroffen.</p> <p>Weitere Informationen finden Sie in „Beispiel: Verwenden einer Arbeitsklasse mit dem Schlüsselwort ALL“ auf Seite 68.</p>

Die folgende Abbildung zeigt eine hierarchische Ansicht der Arbeitstypschlüsselwörter:

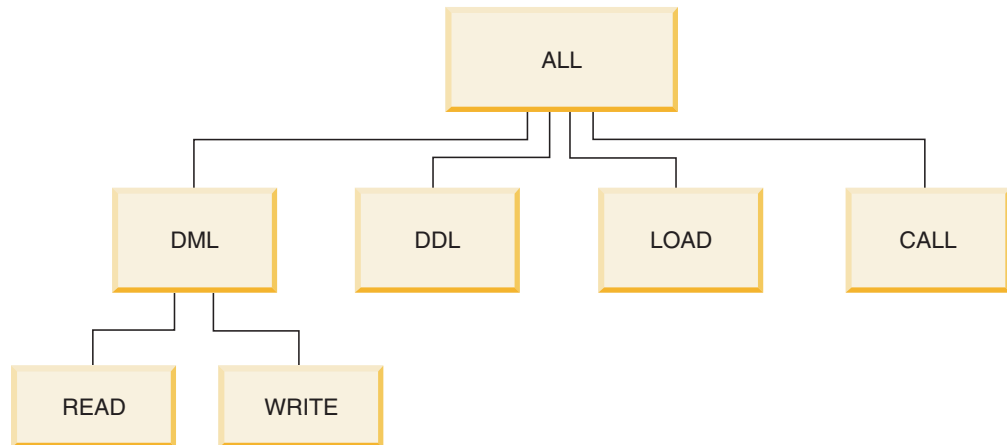


Abbildung 7. Arbeitstypschlüsselwörter

SQL-Anweisungen, die unter keines der verfügbaren Schlüsselwörter fallen, werden nicht klassifiziert und verhalten sich so, als wäre keine Arbeitsklasse und kein Arbeitsklassenset vorhanden. Wenn die Anweisung z. B. SET SCHEMA lautet und die einzige Arbeitsklasse im Arbeitsklassenset den Arbeitstyp DML hat, wird diese Anweisung nicht klassifiziert und keine Arbeitsaktion kann auf sie angewendet werden. Wenn die Aktion MAP ist, wird die Aktivität SET SCHEMA also in der Standardserviceunterklasse (SYSDEFAULTSUBCLASS) ausgeführt. Wenn die Aktion ein Schwellenwert ist, wird kein Schwellenwert auf die Aktivität angewendet.

Zusätzliche Identifikation

Arbeitsklassen erlauben es Ihnen auch, prädiktive Elemente bei der Identifikation für DML-Arbeit (oder READ- und WRITE-Anweisungen) zu verwenden. Prädiktive Elemente sind nützlich, da sie Informationen zu Datenbankaktivitäten bereitstellen, anhand derer Maßnahmen ergriffen werden können, bevor diese Aktivitäten beginnen, Ressourcen auf dem Datenserver zu verbrauchen. Die folgende Tabelle enthält Informationen zu prädiktiven Elementen, die von Arbeitsklassen unterstützt werden:

Tabelle 22. Merkmale für prädiktive Identifikation

Prädiktives Element	Beschreibung
Geschätzter Aufwand	Verwendet den verfügbaren geschätzten Aufwand vom DB2-Compiler, um den gesamten DML-Code in einem gegebenen Timerbereich einzuschließen (z. B. Erstellen einer Arbeitsklasse für alle großen Abfragen mit einem geschätzten Aufwand von über 1.000.000 Timeron)
Geschätzte Kardinalität	Verwendet die geschätzten zurückgegebenen Zeilen (Kardinalität) vom DB2-Compiler, um DML-Code in einem gegebenen Bereich zurückgegebener Zeilen einzuschließen (z. B. Erstellen einer Arbeitsklasse für große Abfragen, die geschätzt mehr als 500 000 Zeilen zurückgeben)

Tabelle 22. Merkmale für prädiktive Identifikation (Forts.)

Prädiktives Element	Beschreibung
Geschätzter Datenzugriff	Verwendet den geschätzten Datenzugriff vom DB2-Compiler, um DML-Code einzuschließen, der auf Daten zugreift, die einem bestimmten Datentag zugewiesen sind (z. B. Erstellen einer Arbeitsklasse für Abfragen, für die geschätzt wird, dass sie auf Daten in Tabellenbereichen mit dem Datentagwert 3 zugreifen) Anmerkung: Die Gruppe von Tabellenbereichen, deren Zugriff der DB2-Compiler vorhersagt, basiert auf der optimierten SQL-Anweisung, die sich von der benutzerdefinierten SQL-Anweisung unterscheiden kann. In einigen Fällen, z. B. beim Einfügen in bereichspartitionierte Tabellen, überschreitet die Anzahl Tabellenbereiche, deren Zugriff vorhergesagt wird, die erwarteten Tabellenbereiche.

Sie können auch Aktivitäten angeben, indem Sie den Schemanamen der Prozedur verwenden, den eine Anweisung CALL aufruft.

Basierend auf den Workloadattributen und Arbeitsklassentypen können Sie Arbeit identifizieren und für die nächste Phase, die Verwaltung der Arbeit, vorbereiten.

Weitere Informationen zum Arbeiten mit Arbeitsklassen und Arbeitsklassensets finden Sie in den folgenden Abschnitten:

Arbeitsklassen und Arbeitsklassensets

Eine Arbeitsklasse ist eine Methode zur Kategorisierung einzelner Datenbankaktivitäten basierend auf Attributen der Aktivitäten. Arbeitsklassen werden in Arbeitsklassensets gruppiert, die von unterschiedlichen Arbeitsaktionssets gemeinsam genutzt werden können.

Beispiele von Datenbankaktivitätsattributen, die bestimmen können, mit welcher Arbeitsklasse eine Aktivität verknüpft ist, sind unter anderem: Aktivitätstyp (DDL, DML, LOAD), geschätzter Aufwand (sofern verfügbar), geschätzte Kardinalität (sofern verfügbar), geschätzter Datentag und Schema (sofern verfügbar).

Arbeitsklassen

Eine Arbeitsklasse hat die folgenden Attribute:

- Den Arbeitsklassennamen, der im Arbeitsklassenset eindeutig sein muss.
- Die Datenbankaktivitätsattribute, die aus den folgenden Informationen bestehen:
 - Dem Typ von Datenbankaktivität, der unter diese Arbeitsklasse fällt. Mit vordefinierten Schlüsselwörtern (z. B. CALL, READ, WRITE, DML, DDL, LOAD oder ALL) können Sie Datenbankanforderungen in unterschiedliche Kategorien klassifizieren. Unterschiedliche Typen von Datenbankaktivitäten können je nach Arbeitstyp mit einer Arbeitsklasse verknüpft werden. Das Schlüsselwort WRITE z. B. umfasst Aktualisierungen, Löschvorgänge, Einfügungen, Zusammenführungen sowie Auswahlvorgänge, die einen Löschvorgang, eine Einfügung oder eine Aktualisierung enthalten. Weitere Informationen finden Sie in „Identifizieren der Arbeit nach Typ, Aufwand oder Datenzugriff mithilfe von Arbeitsklassen“ auf Seite 52.
 - Dem Bereich von Informationen, die Datenbankaktivitäten vom Typ DML oder XQuery weiter kategorisieren:

- Der Typ des anzugebenden Bereichs (entweder Timeron oder Kardinalität). Die Angabe eines Wertebereichs ist optional. Wenn Sie z. B. einen Bereich für eine Arbeitsklasse angeben, können Sie angeben, dass alle Abfragen mit einem geschätzten Aufwand von maximal 100 Timeron anders als die anderen Abfragen verarbeitet werden sollen.
- Die untere Bereichsgrenze.
- Die obere Bereichsgrenze.
- Dem aufzurufenden Routinenschema. Die Angabe des Schemas ist optional. Beim Definieren einer Arbeitsklasse können Sie das Schemaattribut verwenden, um CALL-Anweisungen basierend auf dem Schema der aufzurufenden Prozedur weiter zu klassifizieren. Wenn Sie z. B. SCHEMA1 als Schema einer Arbeitsklasse angeben und der Arbeitstyp CALL ist, werden alle CALL-Anweisungen, die eine SCHEMA1-Prozedur aufrufen, in dieser Arbeitsklasse klassifiziert. Wenn Sie das Schema für einen anderen Arbeitsklassentyp als CALL oder ALL angeben, wird der Fehler SQL0628N zurückgegeben.
- Der Identifikationstag für die Daten, die die Aktivität möglicherweise verarbeitet. Wenn Sie z. B. den Datentag 3 für eine Arbeitsklasse angeben, können Aktivitäten, die Daten in einem Tabellenbereich oder in einer Speichergruppe mit dem Datentag 3 verarbeiten, isoliert und anders behandelt werden.
- Die Bewertungsreihenfolge der Arbeitsklasse (oder Position der Arbeitsklasse im Arbeitsklassenset). Weitere Informationen finden Sie in „Bewertungsreihenfolge von Arbeitsklassen in einem Arbeitsklassenset“ auf Seite 63.
- Eine automatisch generierte Klassen-ID, die die Arbeitsklasse eindeutig bezeichnet.

Sie können Arbeitsklassen auf zwei Arten erstellen:

- Erstellen Sie mit dem Schlüsselwort `WORK CLASS` der Anweisung `CREATE WORK CLASS SET` ein neues Arbeitsklassenset, das die neue Arbeitsklasse enthalten soll.
- Fügen Sie mit dem Schlüsselwort `ADD` der Anweisung `ALTER WORK CLASS SET` die neue Arbeitsklasse einem vorhandenen Arbeitsklassenset hinzu.

Sie können Arbeitsklassen mit dem Schlüsselwort `ALTER WORK CLASS` der Anweisung `ALTER WORK CLASS SET` ändern.

Sie können Arbeitsklassen aus einem Arbeitsklassenset mit dem Schlüsselwort `DROP WORK CLASS` der Anweisung `ALTER WORK CLASS SET` löschen oder mit der Anweisung `DROP WORK CLASS` das Arbeitsklassenset selbst löschen.

Sie können Ihre Arbeitsklassen durch Abfragen der Sicht `SYSCAT.WORKCLASSES` anzeigen.

Arbeitsklassensets

Mit Arbeitsklassensets können Sie eine oder mehrere Arbeitsklassen gruppieren. Ein Arbeitsklassenset hat die folgenden Attribute:

- Einen eindeutigen beschreibenden Namen für das Arbeitsklassenset.
- Kommentare, die Sie für das Arbeitsklassenset angeben wollen.
- Null oder mehr Arbeitsklassen (obwohl eine Arbeitsklasse nur in einem Arbeitsklassenset vorhanden sein kann, muss ein Arbeitsklassenset keine Arbeitsklassen enthalten).
- Eine automatisch generierte ID, die die Arbeitsklasse eindeutig bezeichnet.

Sie können ein neues Arbeitsklassenset mit der Anweisung CREATE WORK CLASS SET erstellen. Sie können ein leeres Arbeitsklassenset erstellen und Arbeitsklassen später hinzufügen, oder Sie können ein Arbeitsklassenset erstellen, das bereits mindestens eine Arbeitsklasse enthält.

Sie können ein vorhandenes Arbeitsklassenset mit der Anweisung ALTER WORK CLASS SET auf die folgenden Arten ändern:

- Fügen Sie Arbeitsklassen zum Arbeitsklassenset hinzu.
- Ändern Sie Arbeitsklassenattribute für Arbeitsklassen im Arbeitsklassenset.
- Löschen Sie Arbeitsklassen aus dem Arbeitsklassenset.

Sie können keine Attribute des Arbeitsklassensets ändern.

Mit der Anweisung DROP WORK CLASS SET können Sie ein Arbeitsklassenset löschen.

Sie können Ihre Arbeitsklassensets durch Abfragen der Katalogsicht SYSCAT-.WORKCLASSETS anzeigen.

Die folgende Abbildung zeigt ein Beispiel von Arbeitsklassen in einem Arbeitsklassenset.

Arbeitsklassenset: Large activities

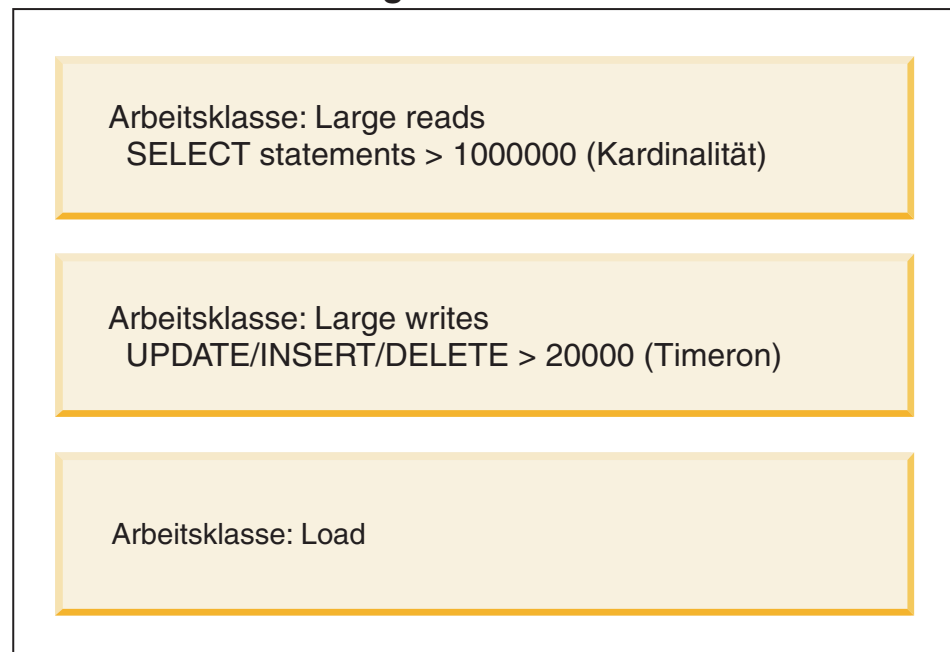


Abbildung 8. Beispiel von Arbeitsklassen in einem Arbeitsklassenset

Damit ein Arbeitsklassenset auf dem System wirksam wird, müssen Sie ein Arbeitsaktionsset definieren und es mit dem Arbeitsklassenset verknüpfen. Durch Verwendung eines Arbeitsaktionssets können Sie ein Arbeitsklassenset einer Service-Superklasse, Workload oder Datenbank zuordnen, um anzugeben, welche Aktion auf die Datenbankaktivitäten angewendet werden soll, die unter die Klassifizierung fallen. Wenn Sie kein Arbeitsaktionsset für das Arbeitsklassenset erstellen, ignoriert der Datenserver das Arbeitsklassenset.

Erstellen einer Arbeitsklasse

Mit der Anweisung `CREATE WORK CLASS SET` oder `ALTER WORK CLASS SET` können Sie eine Arbeitsklasse erstellen.

Vorbereitende Schritte

Zum Erstellen einer Arbeitsklasse benötigen Sie die Berechtigung `WLMADM` oder `DBADM`.

Weitere Voraussetzungen sind in den folgenden Abschnitten aufgeführt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Namenskonventionen

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsklasse zu erstellen:

1. Sie können eine Arbeitsklasse zur gleichen Zeit wie ein neues Arbeitsklassenset erstellen oder eine neue Arbeitsklasse einem vorhandenen Arbeitsklassenset hinzufügen:
 - Verwenden Sie das Schlüsselwort `WORK CLASS` der Anweisung `CREATE WORK CLASS SET`, um eine neue Arbeitsklasse zu erstellen, die einem neuen Arbeitsklassenset hinzugefügt wird.
 - Verwenden Sie das Schlüsselwort `ADD WORK CLASS` der Anweisung `ALTER WORK CLASS SET`, um eine neue Arbeitsklasse zu erstellen, die einem vorhandenen Arbeitsklassenset hinzugefügt wird.

Geben Sie mindestens eine der folgenden Eigenschaften für die neue Arbeitsklasse an:

- Einen Namen für die Arbeitsklasse. Dieser Name muss im Arbeitsklassenset eindeutig sein.
- Attribute für die Arbeitsklasse. Diese Attribute werden verwendet, um eine Aktivität mit der Arbeitsklasse zu verknüpfen:
 - Den Typ von Arbeit, für den die Arbeitsklasse verwendet wird. Verwenden Sie den Parameter `WORK TYPE`, um dieses Merkmal anzugeben.
 - `READ`, das nicht-aktualisierende `SELECT`-Aktivitäten sowie alle `XQuery`-Aktivitäten darstellt. Wenn Sie das Schlüsselwort `READ` angeben, können Sie auch ein optionales Argument für die Klausel `FOR-FROM-TO` oder `DATA TAG` angeben.
 - Mit dem optionalen Argument für die Klausel `FOR-FROM-TO` können Sie einen Bereich für den Aufwand der Anweisung in Timeron oder ihre Kardinalität (d. h. die Anzahl der zurückgegebenen Zeilen) angeben. Sie müssen einen numerischen Wert für den ersten Wert angeben. Für den zweiten Wert können Sie entweder einen numerischen Wert oder den Wert `UNBOUNDED` angeben, wenn Sie keine Obergrenze für den Aufwand oder die Kardinalität der Aktivität erzwingen wollen. Sie können dieses Argument auch für das Schlüsselwort `WRITE`, das Schlüsselwort `DML` und das Schlüsselwort `ALL` angeben. Wenn Sie z. B. `SELECT`-Aktivitäten mit einem Aufwand von mindestens 5.000 Timeron mit dieser Arbeitsklasse verknüpfen wollen, geben Sie Folgendes an:

```
WORK TYPE READ FOR TIMERONCOST FROM 5000 TO UNBOUNDED
```
 - Mit dem optionalen Argument für die Klausel `DATA TAG` können Sie einen Datentag für die Daten angeben, für die geschätzt wird, dass

die Aktivität auf sie zugreift. Sie können einen Wert zwischen 1 und 9 angeben. Wenn die Klausel DATA TAG nicht angegeben wird, kann eine Abfrage auf alle Datentypen zugreifen. Sie können dieses Argument auch für das Schlüsselwort WRITE, das Schlüsselwort DML und das Schlüsselwort ALL angeben.

Wenn Sie dieser Arbeitsklasse z. B. alle SELECT-Aktivitäten zuordnen wollen, für die geschätzt wird, dass sie auf Daten in Tabellenbereichen mit dem Datentagwert 1 zugreifen, geben Sie Folgendes an:

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

- WRITE, das SQL-Aktivitäten darstellt, die Daten in der Datenbank aktualisieren. Wenn Sie das Schlüsselwort WRITE angeben, können Sie auch ein optionales Argument für die Klausel FOR-FROM-TO oder DATA TAG angeben.

Wenn Sie dieser Arbeitsklasse z. B. alle WRITE-Aktivitäten zuordnen wollen, für die geschätzt wird, dass sie während der Kompilierung Daten mit dem Tagwert 5 verarbeiten, geben Sie Folgendes an:

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

- CALL, das CALL-Aktivitäten darstellt.

Wenn Sie das Schlüsselwort CALL angeben, können Sie auch das Schlüsselwort ROUTINES IN SCHEMA angeben, wenn nur CALL-Aktivitäten für Routinen in einem bestimmten Schema mit dieser Arbeitsklasse verknüpft werden sollen. Wenn Sie dieser Arbeitsklasse z. B. nur Aufrufe an Routinen im Schema ACCOUNTS zuordnen wollen, geben Sie Folgendes an:

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

- DML, das SQL-Aktivitäten darstellt, die von den Schlüsselwörtern READ und WRITE abgedeckt werden.

Wenn Sie z. B. alle DML-Aktivitäten mit einem geschätzten Aufwand von 500 bis 1.000 Timeron, für die geschätzt wird, dass sie während der Kompilierung Daten mit dem Tagwert 8 verarbeiten, zuordnen wollen, geben Sie Folgendes an:

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

- DDL, das die folgenden Aktivitäten darstellt:

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

Wenn Sie mit dieser Arbeitsklasse z. B. alle DDL-Aktivitäten verknüpfen wollen, geben Sie Folgendes an:

```
WORK TYPE DDL
```

- LOAD, das eine LOAD-Aktivität darstellt.

Wenn Sie mit dieser Arbeitsklasse z. B. alle LOAD-Aktivitäten verknüpfen wollen, geben Sie Folgendes an:

WORK TYPE LOAD

- ALL, das alle durch die oben aufgelisteten Schlüsselwörter bezeichneten Arbeitstypen darstellt.

Wenn Sie ALL als Arbeitsklassentyp angeben, können Sie auch das Schlüsselwort ROUTINES IN SCHEMA angeben, wenn nur CALL-Aktivitäten für Routinen in einem bestimmten Schema mit dieser Arbeitsklasse verknüpft werden sollen. Sie können auch die Klausel FOR-FROM-TO angeben, wenn alle DML-Aktivitäten, für die ein geschätzter Timeronafwand oder eine Kardinalität angegeben ist, unter diese Klasse fallen sollen. Wenn Sie z. B. dieser Arbeitsklasse sowohl DML-Aktivitäten mit einer Kardinalität von 300 bis 1500 Zeilen als auch Routinen, die vom Schema NEWHIRES aufgerufen werden, zuordnen wollen, geben Sie die folgende Anweisung an. Außerdem können Sie mit dem Argument für die Klausel DATA TAG alle DML-Aktivitäten angeben, die auf Daten in Tabellenbereichen mit einem Datentagwert zugreifen. Da diese Arbeitsklasse den Typ ALL hat, würde sie auch auf andere Aktivitäten angewendet, die kein Schema oder keine Kardinalität haben, z. B.: LOAD-Aktivitäten und DDL-Aktivitäten

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

- Optional. Die Position der Arbeitsklasse im Arbeitsklassenset. Die Position der Arbeitsklasse im Arbeitsklassenset bestimmt die Reihenfolge, in der die Arbeitsklasse beim Klassifizieren einer Aktivität in eine Arbeitsklasse bewertet wird. Wenn eine Arbeitsklassenzuordnung erfolgt, ermittelt der Datenserver zuerst das Arbeitsklassenset, das dem Objekt (entweder einer Servicesuperklasse oder der Datenbank) zugeordnet ist, und wählt dann die erste übereinstimmende Arbeitsklasse im Arbeitsklassenset aus, mit dem eine Arbeitsaktion verknüpft ist. Geben Sie mithilfe des Schlüsselworts POSITION eine der folgenden Optionen an:

- LAST. Die Arbeitsklasse wird am Ende der Liste der Arbeitsklassen im Arbeitsklassenset platziert. Beispiel:

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

- BEFORE *arbeitsklassenname*. Die Arbeitsklasse soll im Arbeitsklassenset erstellt und vor der angegebenen Arbeitsklasse platziert werden. Beispiel:

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

- AFTER *arbeitsklassenname*. Die Arbeitsklasse soll im Arbeitsklassenset erstellt und nach der angegebenen Arbeitsklasse platziert werden. Beispiel:

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

- AT *ganzzahl*. Die Arbeitsklasse soll im Arbeitsklassenset an der durch die ganze Zahl angegebenen Position platziert werden. Beispiel:

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

2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsklasse zur Sicht SYSCAT.WORKCLASSES hinzugefügt.

Ändern einer Arbeitsklasse

Wenn Sie eine Arbeitsklasse ändern müssen, verwenden Sie die Anweisung ALTER WORK CLASS SET.

Vorbereitende Schritte

Zum Ändern einer Arbeitsklasse benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Voraussetzungen sind in „DDL-Anweisungen für Workload-Management“ auf Seite 18 aufgeführt.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsklasse zu ändern:

1. Verwenden Sie das Schlüsselwort ALTER der Anweisung ALTER WORK CLASS SET, um eine oder mehrere der folgenden Eigenschaften zu ändern. Eine Erläuterung der unterstützten Werte für diese Eigenschaften finden Sie in „Erstellen einer Arbeitsklasse“ auf Seite 58.
 - Das Schlüsselwort FOR. Sie können z. B. den Wert für das Schlüsselwort FOR von CARDINALITY in TIMERONCOST ändern.
 - Das Argument FROM *anfangswert* TO *endwert*. Sie können z. B. das Argument von FROM 50 TO 100 in FROM 500 TO 1500 ändern.
 - Die Schlüsselwörter ROUTINES IN SCHEMA oder ROUTINES IN ALL für CALL-Aktivitäten. Wenn z. B. die Arbeitsklasse zurzeit kein Schema angibt, können Sie eines hinzufügen. Sie können auch das Schlüsselwort ALL angeben, sodass die Arbeitsklasse unabhängig vom Schema der Routine auf alle CALL-Anweisungen angewendet wird. ALL ist die Standardeinstellung.
 - Das Schlüsselwort DATA TAG LIST CONTAINS. Sie können z. B. den Wert für dieses Schlüsselwort von ANY in 8 ändern.
 - Das Schlüsselwort POSITION gefolgt von den Schlüsselwörtern LAST, BEFORE, AFTER oder AT. Wenn Sie POSITION BEFORE oder POSITION AFTER angeben, müssen Sie auch die Arbeitsklasse angeben, die Sie für die Positionierung Ihrer geänderten Arbeitsklasse verwenden wollen. Bei Angabe von POSITION AT müssen Sie auch die Positionsnummer angeben. Mit dem Schlüsselwort AT können Sie eine Arbeitsklasse z. B. von der letzten Position an jede andere Position versetzen oder Sie können das Schlüsselwort LAST verwenden, um eine Arbeitsklasse von einer beliebigen Position an die letzte Position zu versetzen.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsklasse in der Sicht SYSCAT.WORKCLASSES aktualisiert.

Löschen einer Arbeitsklasse

Wenn Sie eine Arbeitsklasse nicht mehr benötigen, können Sie sie aus dem Arbeitsklassenset löschen.

Vorbereitende Schritte

Zum Löschen einer Arbeitsklasse benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Voraussetzungen sind in „DDL-Anweisungen für Workload-Management“ auf Seite 18 aufgeführt.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsklasse zu löschen:

1. Verwenden Sie das Schlüsselwort DROP der Anweisung ALTER WORK CLASS SET. Sie können eine Arbeitsklasse nicht löschen, wenn eine Arbeitsaktion in einem beliebigen Arbeitsaktionsset, das mit dieser Arbeitsklasse verknüpft ist, von der Arbeitsklasse abhängt, die Sie löschen wollen. In dieser Situation müssen Sie zuerst alle abhängigen Arbeitsaktionen löschen, bevor Sie die Arbeitsklasse löschen können.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsklasse aus der Sicht SYSCAT.WORKCLASSES entfernt.

Erstellen eines Arbeitsklassensets

Mit der Anweisung CREATE WORK CLASS SET können Sie ein Arbeitsklassenset erstellen.

Vorbereitende Schritte

Zum Erstellen eines Arbeitsklassensets benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Voraussetzungen sind in den folgenden Abschnitten aufgeführt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18

Vorgehensweise

Gehen Sie wie folgt vor, um ein Arbeitsklassenset zu erstellen:

1. Geben Sie die folgenden Eigenschaften für das Arbeitsklassenset mit der Anweisung CREATE WORK CLASS SET an:
 - Einen Namen für das Arbeitsklassenset. Der angegebene Name muss in der Datenbank eindeutig sein.
 - Optional: Eine oder mehrere Arbeitsklassen für das Arbeitsklassenset. Weitere Informationen finden Sie in „Erstellen einer Arbeitsklasse“ auf Seite 58.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird das Arbeitsklassenset zur Sicht SYSCAT.WORKCLASSESETS hinzugefügt.

Ändern eines Arbeitsklassensets

Nach der Erstellung eines Arbeitsklassensets können die Attribute des Arbeitsklassensets nicht mehr geändert werden. Sie können jedoch Arbeitsklassen im Arbeitsklassenset mit der Anweisung ALTER WORK CLASS SET hinzufügen, ändern und löschen.

Vorbereitende Schritte

Zum Ändern eines Arbeitsklassensets benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Voraussetzungen sind in den folgenden Abschnitten aufgeführt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Namenskonventionen

Vorgehensweise

1. Wenn Sie eine neue Arbeitsklasse zum Arbeitsklassenset hinzufügen wollen, verwenden Sie das Schlüsselwort ADD. Informationen zu den Schlüsselwör-

tern, die Sie beim Hinzufügen einer Arbeitsklasse angeben können, finden Sie in „Erstellen einer Arbeitsklasse“ auf Seite 58.

2. Wenn Sie eine Arbeitsklasse ändern wollen, verwenden Sie das Schlüsselwort ALTER. Informationen zum Ändern einer Arbeitsklasse finden Sie in „Ändern einer Arbeitsklasse“ auf Seite 60.
3. Wenn Sie eine Arbeitsklasse löschen wollen, verwenden Sie das Schlüsselwort DROP. Informationen zum Löschen einer Arbeitsklasse aus einem Arbeitsklassenset finden Sie in „Löschen einer Arbeitsklasse“ auf Seite 61. Wenn Sie alle Arbeitsklassen aus dem Arbeitsklassenset löschen wollen, können Sie das Arbeitsklassenset selbst löschen. Weitere Informationen finden Sie in „Löschen eines Arbeitsklassensets“.
4. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Sicht SYSCAT.WORKCLASSES aktualisiert, um hinzugefügte, geänderte oder gelöschte Arbeitsklassen anzuzeigen.

Löschen eines Arbeitsklassensets

Mit der Anweisung DROP WORK CLASS SET können Sie ein Arbeitsklassenset löschen.

Vorbereitende Schritte

Zum Löschen eines Arbeitsklassensets benötigen Sie die Berechtigung WLMADM oder DBADM.

Informationen zu diesem Vorgang

Sie können ein Arbeitsklassenset nur löschen, wenn ihm keine Arbeitsaktionssets zugeordnet sind. Wenn Sie das Arbeitsklassenset löschen wollen, müssen Sie zuerst die abhängigen Arbeitsaktionssets löschen.

Vorgehensweise

Gehen Sie wie folgt vor, um ein Arbeitsklassenset zu löschen:

1. Verwenden Sie die Anweisung DROP WORK CLASS SET.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird das Arbeitsklassenset aus der Sicht SYSCAT.WORKCLASSETS entfernt. Darüber hinaus werden alle Arbeitsklassen, die Teil des Arbeitsklassensets waren, aus der Sicht SYSCAT.WORKCLASSES entfernt.

Bewertungsreihenfolge von Arbeitsklassen in einem Arbeitsklassenset

Ein Arbeitsklassenset kann mehrere Arbeitsklassen enthalten, die mit einer Datenbankaktivität übereinstimmen. Zur Auswahl, unter welche Arbeitsklasse aus einem Arbeitsklassenset eine Aktivität fallen soll, durchläuft der Datenserver die Arbeitsklassen entsprechend der IBM Formreihenfolge und stoppt an der ersten Arbeitsklasse, die mit der Aktivität übereinstimmt.

Wenn keine übereinstimmende Arbeitsklasse vorhanden ist, gehört die Datenbankaktivität zu keiner Arbeitsklasse und es wird keine Arbeitsaktion auf diese Aktivität angewendet.

Sie können die Bewertungsreihenfolge von Arbeitsklassen in einem Arbeitsklassenset beim Erstellen oder Ändern eines Arbeitsklassensets beeinflussen. Wenn Sie ein

Arbeitsklassenset erstellen oder ändern, können Sie mit einer der drei folgenden Methoden die Position festlegen, an der eine Arbeitsklasse im Arbeitsklassenset platziert wird:

- Geben Sie die absolute Position der Arbeitsklasse in der Liste an.
Beispiel: POSITION AT 2. In dieser Situation wird die Arbeitsklasse an der zweiten Position im Arbeitsklassenset platziert. Die bisher an Position 2 befindliche Arbeitsklasse steht jetzt an dritter Stelle, die bisher 3. Arbeitsklasse steht jetzt an vierter Stelle usw. Wenn die in der Anweisung CREATE WORK CLASS SET oder ALTER WORK CLASS SET angegebene Position für die Arbeitsklasse größer ist als die Gesamtzahl der Arbeitsklassen im Arbeitsklassenset, wird die Arbeitsklasse an letzter Stelle positioniert.
- Verwenden Sie das Schlüsselwort POSITION BEFORE oder POSITION AFTER, um die Position der Arbeitsklasse relativ zu anderen bereits im Arbeitsklassenset enthaltenen Arbeitsklassen anzugeben.
- Geben Sie beim Erstellen einer Arbeitsklasse keine Position an.
In dieser Situation wird die neue Arbeitsklasse am Ende der Liste platziert. Die für die Arbeitsklasse angegebene Position in der Liste der Arbeitsklassensets ist nicht notwendigerweise der tatsächliche Wert der Spalte EVALUATIONORDER in der Sicht SYSCAT.WORKCLASSES. Der Datenserver weist automatisch den Folgewert zu, um Lücken zu vermeiden.

Arbeitsklassen werden in der Reihenfolge ihres Empfangs verarbeitet, was sich auf die Bewertungsreihenfolge auswirken kann. Nehmen wir z. B. an, dass Sie die folgende Anweisung absetzen:

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

Als Ergebnis hat die Arbeitsklasse C1 eine Bewertungsreihenfolge von 2 und die Arbeitsklasse C2 hat eine Bewertungsreihenfolge von 1, da C2 die zuletzt verarbeitete Arbeitsklasse war.

Zuordnen von Aktivitäten zu Arbeitsklassen

Wenn ein Arbeitsklassenset über ein Arbeitsaktionsset mit einer Datenbank, Workload oder Servicesuperklasse verknüpft wird und dies direkt vor der Ausführung in der Verarbeitung einer Ausführungs-, Sofortausführungs- oder Öffnungsanforderung oder direkt vor der Ausführung des Dienstprogramms LOAD geschieht, wird die Datenbankaktivität geprüft, um festzustellen, ob sie einer der Bedingungen entspricht, die in den Arbeitsklassen im Arbeitsklassenset angegeben sind.

Die Arbeitsklassen werden im Arbeitsklassenset nach ihrer Bewertungsreihenfolge sortiert. Basierend auf dieser Bewertungsreihenfolge wird die Datenbankaktivität gegen jede Arbeitsklasse auf der Grundlage der Attribute der Datenbankaktivität (wie Aktivitätstyp und Kardinalität) geprüft, bis eine Übereinstimmung gefunden wird oder bis die Liste der Arbeitsklassen im Arbeitsklassenset abgearbeitet wurde.

Nehmen wir an, dass die folgenden Arbeitsklassen in einem Arbeitsklassenset enthalten sind:

- Bewertungsreihenfolge: 1; Arbeitsklassenname: MyLoad; Arbeitsklassentyp: LOAD
- Bewertungsreihenfolge: 2; Arbeitsklassenname: SmallRead; Arbeitsklassentyp: READ; weitere Attribute: geschätzter Aufwand < 300 Timeron
- Bewertungsreihenfolge: 3; Arbeitsklassenname: AllDML; Arbeitsklassentyp: DML
- Bewertungsreihenfolge: 4; Arbeitsklassenname: LargeRead; Arbeitsklassentyp: READ; weitere Attribute: geschätzter Aufwand > 301 Timeron
- Bewertungsreihenfolge: 5; Arbeitsklassenname: MyDDL; Arbeitsklassentyp: DDL

Wenn eine Anweisung SELECT mit einem geschätzten Aufwand von 200 Timeron empfangen wird, wird sie der Arbeitsklasse SmallRead zugeordnet. Wenn eine DDL-Aktivität (wie z. B. CREATE TABLE) empfangen wird, wird sie der Arbeitsklasse MyDDL zugeordnet. Wenn eine Anweisung SELECT mit einem geschätzten Aufwand von 500 Timeron empfangen wird, wird sie der Arbeitsklasse AllDML zugeordnet, da AllDML vor der Arbeitsklasse LargeRead positioniert ist. Weitere Informationen finden Sie in „Beispiel: Verwenden einer Arbeitsklasse mit dem Schlüsselwort ALL“ auf Seite 68.

Von Schwellenwerten unterstützte Arbeitsklassifikationen

Obwohl jeder der Schwellenwerttypen, die in Arbeitsaktionen verwendet werden können, mit einer beliebigen Arbeitsklasse verknüpft werden kann, werden nicht alle Typen von Datenbankaktivitäten für alle diese Schwellenwerttypen unterstützt.

Wenn Sie z. B. eine Arbeitsklasse für DDL erstellen und dann diese Arbeitsklasse mit einer Schwellenwertarbeitsaktion ESTIMATEDSQLCOST verknüpfen, wird dieser Schwellenwert auf keine der Anfragen angewendet, die unter DDL klassifiziert sind, da DDL-Anweisungen keinen geschätzten Aufwand haben. Wenn Sie eine Arbeitsklasse für ALL erstellen und dann diese Arbeitsklasse mit einer Schwellenwertarbeitsaktion ESTIMATEDSQLCOST verknüpfen, wird der Schwellenwert nur auf die Datenbankaktivitäten mit geschätztem Aufwand angewendet, obwohl alle Datenbankaktivitäten zur Arbeitsklasse ALL gehören.

Die folgenden Tabelle zeigen, welche Arbeitsklassenkategorien von welchen Schwellenwerttypen unterstützt werden:

Tabelle 23. Von Schwellenwerten unterstützte Arbeitsklassifikationen

	„ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143	„ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147	„CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153	„CPUTIME (Schwellenwert)“ auf Seite 144
READ, einschließlich SET-Anweisungen mit eingebettetem READ-SQL	Ja	Ja	Ja ¹	Ja
WRITE, einschließlich SET-Anweisungen mit eingebettetem WRITE-SQL	Ja	Ja	Ja ¹	Ja
CALL	Ja	Nein	Nein	Ja
DML, einschließlich SET-Anweisungen mit eingebettetem READ- oder WRITE-SQL	Ja	Ja	Ja ¹	Ja
DDL	Ja	Nein	Ja ¹	Nein
LOAD	Ja	Nein	Ja ¹	Nein
ALL	Ja	Einige	Ja ¹	Einige

Anmerkung:

1. Aktivitäten, die in benutzerdefinierten Funktionen (UDFs) ausgeführt werden und diese Arbeitsklassifikationen enthalten, sind vom Schwellenwert CONCURRENTDBCOORDACTIVITIES nicht betroffen.

Tabelle 24. Von Schwellenwerten unterstützte Arbeitsklassifikationen (Fortsetzung)

	„SQLROWSREAD (Schwellenwert)“ auf Seite 148	„SQLROWSRETURNED (Schwellenwert)“ auf Seite 150	„SQLTEMPSPACE (Schwellenwert)“ auf Seite 151
READ, einschließlich SET-Anweisungen mit eingebettetem READ-SQL	Ja	Ja	Ja

Tabelle 24. Von Schwellenwerten unterstützte Arbeitsklassifikationen (Fortsetzung) (Forts.)

	„SQLROWSREAD (Schwellenwert)“ auf Seite 148	„SQLROWSRETURNED (Schwellenwert)“ auf Seite 150	„SQLTEMPSPACE (Schwellenwert)“ auf Seite 151
WRITE, einschließlich SET-Anweisungen mit eingebettetem WRITE-SQL	Ja	Ja	Ja
CALL	Nein	Nein (siehe Anmerkung)	Nein
DML, einschließlich SET-Anweisungen mit eingebettetem READ- oder WRITE-SQL	Ja	Ja	Ja
DDL	Nein	Nein	Nein
LOAD	Nein	Nein	Nein
ALL	Einige	Einige	Einige

Anmerkung:

- Obwohl die Anweisungen in der aufgerufenen Prozedur eventuell Zeilen zurückgeben, werden diese nicht vom Schwellenwert SQLROWSRETURNED kontrolliert, da die Zeilen nicht als Ergebnis einer Anweisung CALL zurückgegeben werden.

Beispiel: Analysieren von Workloads nach Aktivitätstyp

Sie können mit DB2-Workload-Management-Tabellenfunktionen die Workloads in Ihrer Umgebung im Hinblick auf die Typen von ausgeführten Aktivitäten untersuchen.

In einigen Situationen sind Sie möglicherweise interessiert am Verhalten eines bestimmten Typs von Aktivitäten, z. B. LOAD-Aktivitäten. Sie können z. B. folgendermaßen beobachten, wie viele LOAD-Aktivitäten gleichzeitig auf dem System aktiv sind:

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

Sie können eine Anzahl der Aktivitäten eines bestimmten Typs, die seit der letzten Zurücksetzung der DB2-Workload-Management-Statistikdaten übergeben wurden, mit der Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS ermitteln (siehe folgendes Beispiel). Nehmen wir an, dass die Arbeitsklassen READCLASS und LOADCLASS für Aktivitäten vom Typ READ und vom Typ LOAD vorhanden sind. * stellt alle Aktivitäten dar, die nicht in die Arbeitsklasse READCLASS oder LOADCLASS fallen.

```
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

Sie können die durchschnittliche Laufzeit der LOAD-Aktivitäten anzeigen, indem Sie ein Arbeitsaktionsset für die Zuordnung der LOAD-Aktivitäten zu einer bestimmten Serviceunterklasse erstellen. Nehmen wir z. B. an, dass Sie LOAD-Aktivitäten der Serviceunterklasse LOADSERVICECLASS unter der Servicesuperklasse MYSUPERCLASS zuordnen wollen. Dann können Sie die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS abfragen:

```
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

Beispiel: Verwalten bestimmter Aktivitätstypen mithilfe von Arbeitsklassensets

Das folgende Beispiel zeigt, wie mithilfe eines Arbeitsklassensets DML-Aktivitäten verwaltet werden können.

Nehmen wir an, dass jeden Tag eine große Anzahl von Anwendungen auf Ihrer Datenbank NONAME ausgeführt werden und dass in letzter Zeit einige Leistungsprobleme aufgetreten sind. Im Hinblick auf einige dieser Probleme beschließen Sie, dass Sie in der Lage sein müssen, die Anzahl großer Abfragen (d. h. aller Abfragen mit einem geschätzten Aufwand von mehr als 9.999 Timeron oder einer geschätzten Kardinalität von mehr als 9.999 Zeilen), die gleichzeitig in der Datenbank ausgeführt werden können, zu steuern.

Zur Steuerung der Anzahl großer Abfragen, die in der Datenbank ausgeführt werden können, gehen Sie wie folgt vor:

1. Erstellen Sie das Arbeitsklassenset MYWORKCLASSSET, das zwei Arbeitsklassen enthält: eine für Abfragen mit einem hohen geschätzten Aufwand und eine für Abfragen mit einer hohen geschätzten Kardinalität. Beispiel:

```
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. Erstellen Sie ein Arbeitsaktionsset DATABASEACTIONS, das zwei Arbeitsaktionen enthält, die auf die Arbeitsklassen im Arbeitsklassenset MYWORKCLASSSET auf Datenbankebene angewendet werden sollen.

```
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)
```

Darüber hinaus werden täglich mehrere große Verwaltungsanwendungen auf der Datenbank ausgeführt. Diese Anwendungen sollen in einem einzelnen Ressourcenpool ausgeführt werden. Dazu erstellen Sie eine Servicesuperklasse namens ADMINAPPS für diese Anwendungen. Für jede Anwendung erstellen Sie eine Workload, um sie der Servicesuperklasse ADMINAPPS zuzuordnen.

Da es wichtig ist, dass die Abfragen (SELECT-Anweisungen) schnell ausgeführt werden, beschließen Sie, für diese Abfragen eine Serviceunterklasse namens SELECTS in der Servicesuperklasse ADMINAPPS zu erstellen.

Gehen Sie wie folgt vor, um die SELECT-Anweisungen der Serviceunterklasse SELECTS zuzuordnen:

1. Erstellen Sie ein Arbeitsklassenset SELECTDML, das eine Arbeitsklasse für alle SELECT-Anweisungen enthält, die die Datenbank nicht aktualisieren:
2. Erstellen Sie ein Arbeitsaktionsset ADMINAPPSACTIONS. Dieses Arbeitsaktionsset enthält eine Arbeitsaktion, die auf die Arbeitsklasse im Arbeitsklassenset SELECTDML auf der Ebene der Servicesuperklasse angewendet werden soll.

```
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)
```

Beispiel: Verwenden einer Arbeitsklasse mit dem Schlüsselwort ALL

Dieses Beispiel zeigt, wie Sie mit einer als ALL definierten Arbeitsklasse arbeiten können, die potenziell alle erkannten Aktivitäten in der Datenbank abdeckt.

In der Infrastruktur des DB2-Workload-Managements wird die folgende spezifische Teilmenge aller möglichen Datenbankaktivitäten erkannt: Ladeoperationen, die Anweisung CALL, alle DDL-Anweisungen und alle DML-Anweisungen. Das DB2-Workload-Management unterstützt die Überwachung und Steuerung dieser *erkannten Aktivitäten* komplett.

Wenn eine Arbeitsklasse mit dem Typ ALL in Verbindung mit einer Zuordnungsarbeitsaktion verwendet wird, werden alle erkannten Datenbankaktivitäten der in der Arbeitsaktion angegebenen Serviceunterklasse zugeordnet. Wenn eine Arbeitsklasse mit dem Typ ALL in Verbindung mit einer Schwellenwertarbeitsaktion verwendet wird, bestimmt der Schwellenwerttyp, auf welche Datenbankaktivitäten der Schwellenwert angewendet wird. Betrachten wir das folgende Beispiel.

Nehmen wir an, dass Sie ein Arbeitsklassenset namens Example mit den folgenden Arbeitsklassen erstellen. Die Arbeitsklassen werden in der folgenden Reihenfolge bewertet:

1. SMALLDML, für alle SQL-Anweisungen vom Typ DML mit einem geschätzten Aufwand von weniger als 1000 Timeron
2. LOADUTIL, für das Dienstprogramm LOAD
3. ALLACTIVITY, für die gesamte Datenbankaktivität

ALLACTIVITY wird als letzte Arbeitsklasse bewertet. Sie umfasst Datenbankaktivitäten, die keiner der ersten drei Arbeitsklassen entsprechen.

Der DDL-Code zum Erstellen dieses Arbeitsklassensets lautet:

```
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)
```

Nehmen wir an, dass Sie eine Servicesuperklasse namens EXAMPLESERVICECLASS haben und dass diese zwei Serviceunterklassen namens SMALLACTIVITY und OTHERACTIVITY enthält. Sie wollen das System so einrichten, dass alle klei-

nen Datenbankaktivitäten in der Serviceunterklasse SMALLACTIVITY und alle anderen erkannten Datenbankaktivitäten mit Ausnahme des Dienstprogramms LOAD in der Serviceunterklasse OTHERACTIVITY ausgeführt werden. Sie wollen das Dienstprogramm LOAD keiner anderen Serviceunterklasse neu zuordnen. Dieses soll stattdessen in der Standardserviceunterklasse ausgeführt werden.

Dazu richten Sie das Arbeitsaktionsset SERVICECLASSACTIONS für die Serviceunterklasse EXAMPLESERVICECLASS ein. Das Arbeitsaktionsset SERVICECLASSACTIONS enthält die folgenden Arbeitsaktionen.

Tabelle 25. Arbeitsaktionsset SERVICECLASSACTIONS

Arbeitsaktion	Angewendet auf Arbeitsklasse	Aktion
MAPDML	SMALLDML	Wird der Serviceunterklasse SMALLACTIVITY zugeordnet
COUNTLOAD	LOADUTIL	Zählt die Anzahl der LOAD-Aktivitäten
MAPOTHER	ALLACTIVITY	Wird der Serviceunterklasse OTHERACTIVITY zugeordnet

Der DDL-Code zum Erstellen dieses Arbeitsaktionssets lautet:

```
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)
```

Mit dieser Konfiguration werden alle kleinen DML-Anweisungen unter der Serviceunterklasse SMALLACTIVITY ausgeführt. Die Arbeitsaktion COUNTLOAD wird auf die Arbeitsklasse LOADUTIL angewendet, die unter der Standardserviceunterklasse ausgeführt wird. Alle anderen erkannten Datenbankaktivitäten werden unter der Serviceunterklasse OTHERACTIVITY ausgeführt.

Anmerkung: Befände sich die Arbeitsklasse ALLACTIVITY an erster Stelle in der Bewertungsreihenfolge, würden alle erkannten Aktivitäten der Serviceunterklasse OTHERACTIVITY zugeordnet.

Nehmen wir jetzt an, dass Sie ein Arbeitsaktionsset für die Datenbank definieren und Schwellenwerte anwenden wollen, die steuern, welche Aktivitäten gleichzeitig auf dem System ausgeführt werden dürfen. Sie könnten ein Arbeitsaktionsset namens DATABASEACTIONS mit den folgenden Arbeitsaktionen erstellen. Der DDL-Code zum Erstellen dieses Arbeitsaktionssets lautet:

```
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)
```


Table 26. Arbeitsaktionsset DATABASEACTIONS

Arbeitsaktion	Angewendet auf Arbeitsklasse	Schwellenwerttyp und -wert	Aktion
CONCURRENTSMALLDML	SMALLDML	Gemeinsamer Zugriff: bis zu 1000 Anweisungen; Warteschlange: bis zu 10.000 Anweisungen	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
CONCURRENTLOAD	LOADUTIL	Gemeinsamer Zugriff: bis zu 2 Vorkommen; Warteschlange: bis zu 10 Vorkommen	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
CONCURRENTOTHER	ALLACTIVITY	Gemeinsamer Zugriff: bis zu 100 Aktivitäten; Warteschlange: bis zu 100 Aktivitäten	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
MAXCOSTALLOWED	ALLACTIVITY	Geschätzter SQL-Aufwand: bis zu 1 000 000 Timeron	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten

Wenn diese Arbeitsaktionen angewendet werden, können bis zu 1000 kleine SQL-Anweisungen vom Typ DML (aufgrund der Arbeitsklasse SMALLDML) gleichzeitig ausgeführt werden und bis zu 10 000 dieser Anweisungen können in die Warteschlange gestellt werden. Nur zwei Vorkommen des Dienstprogramms LOAD können gleichzeitig ausgeführt werden und bis zu 10 Vorkommen können in die Warteschlange gestellt werden. Nur 100 Aktivitäten, die nicht den Typ LOAD haben und keine kleinen DML-Anweisungen sind, dürfen gleichzeitig ausgeführt werden und nur 100 dieser Aktivitäten können in die Warteschlange gestellt werden. In allen Situationen kann bei einem Verstoß gegen einen Warteschlangenschwellenwert die Datenbankaktivität nicht ausgeführt werden und es wird eine Fehlermeldung zurückgegeben.

Darüber hinaus wird die Arbeitsaktion MAXCOSTALLOWED auf die Arbeitsklasse ALLACTIVITY angewendet. Dies bedeutet, dass eine Datenbankaktivität mit einem geschätzten Aufwand (d. h. DML- und XQueries-Anweisungen) von mehr als 1 000 000 Timeron nicht ausgeführt werden darf. Obwohl die Arbeitsaktion MAXCOSTALLOWED auf die Arbeitsklasse ALLACTIVITY angewendet wird, wirkt sich diese Arbeitsaktion nur auf Datenbankaktivitäten mit einem geschätzten Aufwand von mehr als 1 000 000 Timeron aus. Diese Arbeitsaktion betrifft keine Aktivitäten, die keinen geschätzten Aufwand haben, z. B. DDL.

Kapitel 3. Aktivitätenverwaltung

Sobald Sie die auf Ihrem Datenserver ausgeführte Arbeit identifiziert haben, können Sie diese Arbeit aktiv verwalten, indem Sie Ressourcen zuweisen und die Steuerelemente festlegen.

Ressourcenzuordnung mit Serviceklassen

Eine Serviceklasse definiert eine Ausführungsumgebung, in der Arbeit ausgeführt werden kann. Diese Ausführungsumgebung weist verfügbare Ressourcen zu und kann Schwellenwerte umfassen, die bestimmen, wie Arbeit ausgeführt werden darf.

Die gesamte Arbeit wird in einer Serviceklasse ausgeführt. Sie verwenden Workloads, um Arbeit Serviceklassen zuzuordnen. Oder Sie weisen Arbeit Serviceunterklassen in einer Servicesuperklasse mithilfe von Workloads, der Schwellenwertaktion REMAP ACTIVITY oder der Arbeitsaktion MAP ACTIVITY zu. Beim Definieren einer Workload geben Sie die Serviceklasse an, in der die mit dieser Workload verknüpfte Arbeit ausgeführt wird. Standardmäßig ist auch eine Standardbenutzerworkload (SYSDEFAULTUSERWORKLOAD) vorhanden, die Arbeit der Standardbenutzerserviceklasse (SYSDEFAULTUSERCLASS) zuordnet, sodass jede Arbeit, die nicht explizit mit einer benutzerdefinierten Workload einer benutzerdefinierten Serviceklasse zugeordnet wird, in der Standardbenutzerserviceklasse ausgeführt wird.

Ohne Serviceklassen können Anforderungen nicht in erkennbaren logischen Gruppierungen organisiert werden (siehe hierzu folgende Abbildung).

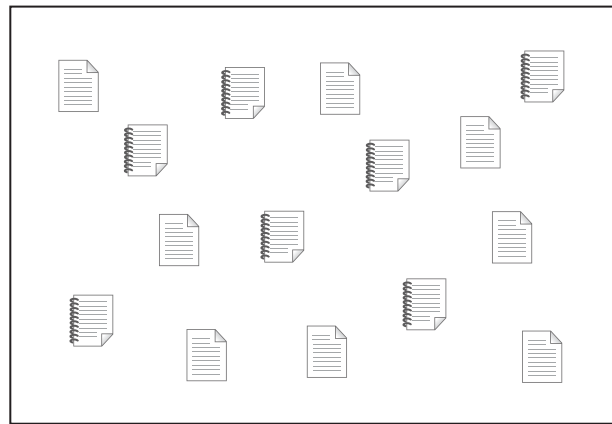


Abbildung 9. Nicht organisierte Arbeit

Sie können unterschiedliche Servicesuperklassen erstellen, um die Ausführungsumgebung für unterschiedliche Arten von Arbeit bereitzustellen, und dann den Servicesuperklassen die zutreffenden Anforderungen zuordnen. Nehmen wir einmal an, Sie haben Anwendungen aus zwei unterschiedlichen Geschäftsbereichen: Finance und Inventory. Dabei hat jeder Geschäftsbereich eigene Anwendungen, um die jeweiligen Aufgaben erfüllen zu können, für die der Geschäftsbereich in der Organisation zuständig ist. Sie können die Anforderungen in Kategorien aufteilen, die für Ihre Workload-Management-Ziele sinnvoll sind. In der folgenden Abbildung wer-

den unterschiedliche Servicesuperklassen den unterschiedlichen Geschäftsbereichen zugeordnet.

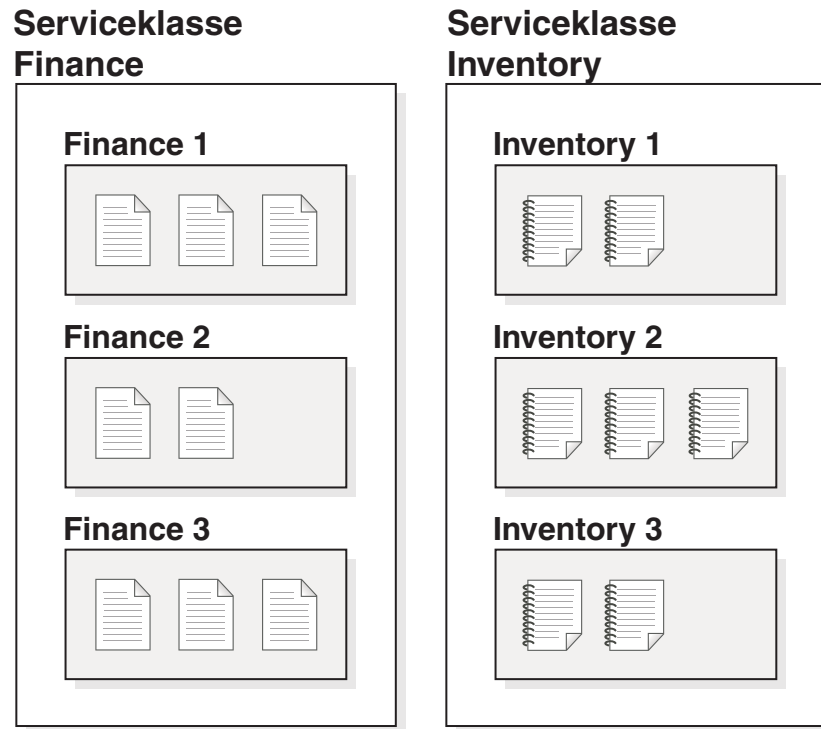


Abbildung 10. Nach Serviceklassen organisierte Arbeit

In der Abbildung oben sind die Aktivitäten in beiden Servicesuperklassen noch weiter unterteilt. Die Serviceklasse stellt eine zweischichtige Hierarchie bereit: eine Servicesuperklasse und darunter Serviceunterklassen. Diese Hierarchie lässt eine komplexere Unterteilung der Ausführungsumgebung und eine bessere Emulation eines realistischen Modells zu. Sofern nichts anderes angegeben ist, übernehmen Serviceunterklassen ihre Merkmale von der Servicesuperklasse. Mit Serviceunterklassen können Sie die Arbeit in der Servicesuperklasse weiter unterteilen.

Priorisierung und Ressourcensteuerung

Beim Erstellen oder Ändern eines Serviceklassenobjekts können Sie eine Reihe von Möglichkeiten zur Ressourcensteuerung definieren:

Tabelle 27. Ressourcensteuerung durch Serviceklassen

Steuerung	Beschreibung
Agentenpriorität	Diese Steuerung legt eine Prozessorprioritätsstufe für die Agententhreads in einer Serviceklasse fest. Diese Priorität wird als relative (Delta-)Priorität zu anderen Threads und Prozessen auf dem Datenserver an das Betriebssystem übergeben. Anmerkung: Diese Steuerung kann nicht festgelegt werden, wenn der abgehende Korrelator verwendet wird.
Vorablesezugriffspriorität	Diese Steuerung ordnet Vorablesezugriffsanforderungen eine Priorität zu, die sich auf die Reihenfolge auswirkt, in der der Datenserver sich mit ihnen befasst.

Tabelle 27. Ressourcensteuerung durch Serviceklassen (Forts.)

Steuerung	Beschreibung
Pufferpoolpriorität	Diese Steuerung weist Serviceklassen eine Pufferpoolpriorität zu. Dies wirkt sich auf die Wahrscheinlichkeit aus, mit der von Aktivitäten in einer Serviceklasse abgerufene Seiten ausgelagert werden.
Abgehender Korrelator	<p>Diese Steuerung erlaubt es einer Workload, mehrere ihrer Ressourcen von einem Betriebssystem-Workload-Manager wie AIX Workload Manager oder Linux-Workload-Management steuern zu lassen. Die Kennung wird durch den Agenten an den externen Workload-Manager übergeben und nimmt eine Zuordnung zu einer mit dem Manager definierten Ressourcengruppe vor.</p> <p>Wenn das DB2-Workload-Management in Verbindung mit einem Betriebssystem-Workload-Manager verwendet wird, stehen weitere Steuerelemente zur Verfügung. Mit AIX Workload Manager können Sie den Prozessorressourcenanteil steuern, der jeder Serviceklasse zugeordnet wird, indem Sie einen minimalen, maximalen oder relativen Anteil der Prozessorressourcen für jede Serviceklasse festlegen. Mit Linux-Workload-Management können Sie den Anteil der CPU-Ressourcen steuern, indem Sie Anteile für jede Serviceklasse relativ zur Linux-Standardklasse festlegen.</p> <p>Anmerkung: Diese Steuerung kann nicht festgelegt werden, wenn eine Agentenpriorität verwendet wird.</p>

Serviceunterklassen

Die Servicesuperklasse ist zwar die höchste Arbeitsebene, Aktivitäten werden jedoch nur in Serviceunterklassen ausgeführt. Jede Servicesuperklasse enthält eine Standardserviceunterklasse, in der die Aktivitäten ausgeführt werden, die nicht explizit einer definierten Unterklasse zugeordnet sind. Die Standardunterklasse wird beim Erstellen der Servicesuperklasse erstellt. Sie können weitere Unterklassen in einer Serviceklasse erstellen, wenn Sie diese zur weiteren Eingrenzung von Arbeit benötigen. Mit Ausnahme von Histogrammen und den Optionen COLLECT ACTIVITY DATA, COLLECT AGGREGATE ACTIVITY DATA und COLLECT AGGREGATE REQUEST DATA übernimmt eine Serviceunterklasse die Attribute ihrer Servicesuperklasse, sofern nichts anderes angegeben ist. Die Ressourcen in der Superklasse werden von allen darin enthaltenen Unterklassen gemeinsam genutzt.

Sie können nur eine einzige Ebene von Unterklassen definieren (d. h., Sie können keine Unterklasse in einer anderen Unterklasse, sondern nur in einer Servicesuperklasse definieren).

Angepasste DB2-Workload-Management-Konfiguration mit Workloads und Serviceklassen

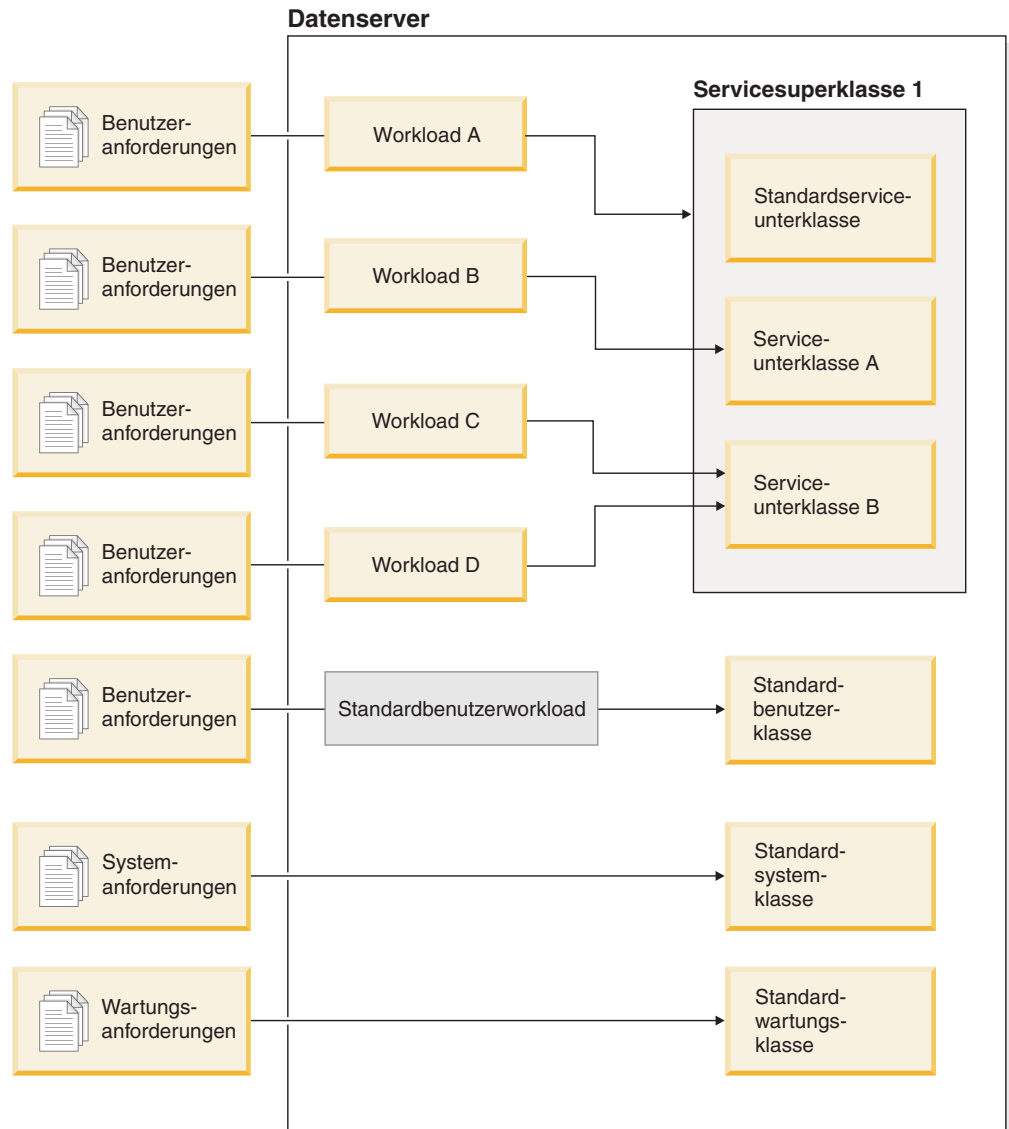


Abbildung 11. Angepasste DB2-Workload-Management-Konfiguration mit Workloads und Serviceklassen

Sobald Benutzeranforderungen in den Datenserver eingehen, werden sie als einer bestimmten Workload zugehörig identifiziert und einer Servicesuperklasse oder -unterklasse zugeordnet. Es gibt auch Systemanfragen (z. B. Vorablesezugriffe), die unter einer speziellen Standardsystems-serviceklasse (SYSDEFAULTSYSTEMCLASS) ausgeführt werden, und von DB2 gesteuerte Wartungsanforderungen (z. B. automatische RUNSTATS vom Diagnosemonitor), die unter einer Standardwartungs-serviceklasse (SYSDEFAULTMAINTENANCECLASS) ausgeführt werden.

Sie können Ihre Serviceklassen durch Abfragen der Katalogsicht SYSCAT.SERVICE-CLASSES anzeigen.

Standardservicesuperklassen und -unterklassen

Jede neue Datenbank bzw. jede Datenbank, für die ein Upgrade durchgeführt wurde, hat drei vordefinierte Standardservicesuperklassen: die Standardbenutzerklasse, die Standardverwaltungsklasse und die Standardsystemklasse.

Die Standardservicesuperklassen können nicht inaktiviert oder gelöscht werden.

Alle Standardservicesuperklassen werden mit einer Standardserviceunterklasse erstellt. Für die Standardservicesuperklassen können keine weiteren Serviceunterklassen erstellt werden. Die Standardserviceunterklasse hat immer den Namen SYSDEFAULTSUBCLASS und wird wie folgt erstellt:

SYSDEFAULTUSERCLASS

SYSDEFAULTSUBCLASS

SYSDEFAULTSYSTEMCLASS

SYSDEFAULTSUBCLASS

SYSDEFAULTMAINTENANCECLASS

SYSDEFAULTSUBCLASS

Abbildung 12. Zweischichtige Serviceklassenhierarchie

Die gesamte Arbeit, die von Verbindungen an eine Standardservicesuperklasse abgesetzt wird, wird in der Standardserviceunterklasse dieser Servicesuperklasse ausgeführt.

Standardservicesuperklassen und ihre Standardserviceunterklassen werden nur beim Löschen der Datenbank gelöscht. Sie können nicht mit der Anweisung DROP SERVICE CLASS gelöscht werden.

Standardbenutzerservicesuperklasse (SYSDEFAULTUSERCLASS)

Standardmäßig werden alle Benutzeraktivitäten in der Standardbenutzerklasse SYSDEFAULTUSERCLASS ausgeführt.

Standardwartungsservicesuperklasse (SYSDEFAULTMAINTENANCECLASS)

Die Standardwartungsservicesuperklasse verfolgt die internen DB2-Verbindungen, die Tasks zur Datenbankpflege und -verwaltung ausführen. Verbindungen von den DB2-ABP-Agenten (ABP - Asynchronous Background Processing - asynchrone Hintergrundverarbeitung) werden dieser Service-superklasse zugeordnet. ABP-Agenten sind interne Agenten, die Tasks zur Datenbankpflege ausführen. Die asynchrone Indexbereinigung (AIC - Asynchronous Index Cleanup) ist ein Beispiel für eine ABP-Task. ABP-Agenten verringern automatisch ihre Ressourcenauslastung und die Anzahl der Subagenten, wenn sich die Anzahl der Benutzerverbindungen auf dem Datenserver erhöht. Dienstprogramme, die von Benutzerverbindungen abgesetzt werden, werden mithilfe regulärer Serviceklassen zugeordnet. Auf SYSDEFAULTMAINTENANCECLASS können Sie keine Serviceklassenschwellenwerte implementieren.

Folgende interne Verbindungen werden von der Standardwartungsservicesuperklasse verfolgt:

- ABP-Verbindungen (einschließlich AIC)

- Vom Diagnosemonitor eingeleitetes BACKUP
- Vom Diagnosemonitor eingeleitete RUNSTATS
- Vom Diagnosemonitor eingeleitetes REORG

Standardsystemservicesuperklasse (SYSDEFAULTSYSTEMCLASS)

Die Standardsystemservicesuperklasse verfolgt interne DB2-Verbindungen und -Threads, die Tasks auf Systemebene ausführen. Sie können für diese Servicesuperklasse weder Serviceunterklassen definieren noch Workloads oder Arbeitsaktionen mit ihr verknüpfen. Außerdem können Sie bei SYSDEFAULTSYSTEMCLASS keine Serviceklassenschwellenwerte implementieren. Folgende DB2-Threads und -Verbindungen werden von der Standardsystemservicesuperklasse verfolgt:

- ABP-Dämon
- Manager für Speicher mit automatischer Leistungsoptimierung
- EDUs (Engine Dispatchable Units - von der Steuerkomponente zuteilbare Einheiten) (db2pfchr)
- Seitenlösch-EDUs (db2pclnr)
- Protokoll-Lese-EDUs (db2loggr)
- Protokollausgabe-EDUs (db2loggw)
- Protokolldateilese-EDUs (db2lfr)
- Deadlock-Detektor-EDUs (db2dlock)
- Ereignismonitore (db2evm)
- Ereignismonitor für schnelle Ausgabeprogramme (db2fw)
- Verbindungen, die Tasks auf Systemebene ausführen

Zuordnung von Aktivitäten zu Serviceklassen

Alle Datenbankverbindungen werden am Anfang der ersten UOW (Unit of Work - Arbeitseinheit) einer Workload zugeordnet. Beim Start eines Workloadvorkommens werden alle Aktivitäten, die unter diesem Workloadvorkommen ausgeführt werden, basierend auf dem in der Workloaddefinition festgelegten Serviceklassennamen den Serviceklassen zugeordnet.

Der Datenserver ordnet eine Verbindung einer Workloaddefinition zu, wenn die Verbindung die für diese Workloaddefinition definierten Bedingungen erfüllt. Sie können beispielsweise eine Workload-Management-Konfiguration einrichten, so dass alle Verbindungen von der Anwendung A zur Workloaddefinition Alpha gehören, während alle Verbindungen von der Anwendung B zur Workloaddefinition Beta gehören.

Wenn das Workloadvorkommen einer Servicesuperklasse zugeordnet wird, können Aktivitäten, die für dieses Workloadvorkommen übergeben werden, mithilfe eines Arbeitsaktionssets einer benutzerdefinierten Serviceunterklasse in dieser Servicesuperklasse zugeordnet werden.

Sie können mithilfe der Workload Aktivitäten von einer Verbindung einer Servicesuperklasse zuordnen, indem Sie das Schlüsselwort SERVICE CLASS der Anweisung CREATE WORKLOAD angeben. Wenn keine Arbeitsklasse oder Arbeitsaktion auf die Aktivität angewendet wird, wird die Aktivität in der Standardserviceunterklasse der Servicesuperklasse ausgeführt. Sie können mithilfe einer Workload auch Aktivitäten von einer Verbindung einer Serviceunterklasse in der Servicesuperklasse zuordnen, indem Sie das Schlüsselwort UNDER für das Schlüsselwort SERVICE CLASS der Anweisung CREATE WORKLOAD angeben. In diesem Fall gehört die Verbindung noch zur Servicesuperklasse, aber alle Aktivitäten, die von dieser Ver-

bindung abgesetzt werden, werden automatisch der Serviceunterklasse zugeordnet, die in der Workloaddefinition angegeben ist.

Nachdem eine Aktivität einer Serviceunterklasse zugeordnet wurde und ihre Ausführung begonnen hat, bleibt die Aktivität in dieser Serviceunterklasse, es sei denn, Sie ordnen sie (mithilfe eines Schwellenwerts) einer anderen Serviceunterklasse zu. Neuordnung ist der Prozess, mit dem Sie die Ressourcenzuordnungen für eine Aktivität ändern können, indem Sie die Aktivität einer anderen Serviceunterklasse zuordnen. Sowohl die Quellen- als auch die Zielserviceunterklasse müssen in derselben Superklasse enthalten sein. Nach der Neuordnung der Aktivität wird ihre Ausführung in der neuen Serviceunterklasse fortgesetzt.

Die folgende Abbildung zeigt die Beziehung zwischen Verbindungen, Workloads und Servicesuperklassen. Verbindungen, die die Definition von Workload A erfüllen, werden der Servicesuperklasse 1 zugeordnet; Verbindungen, die die Definition von Workload B oder C erfüllen, werden der Servicesuperklasse 2 zugeordnet; Verbindungen, die die Definition von Workload D erfüllen, werden der Servicesuperklasse SYSDEFAULTUSERCLASS zugeordnet.

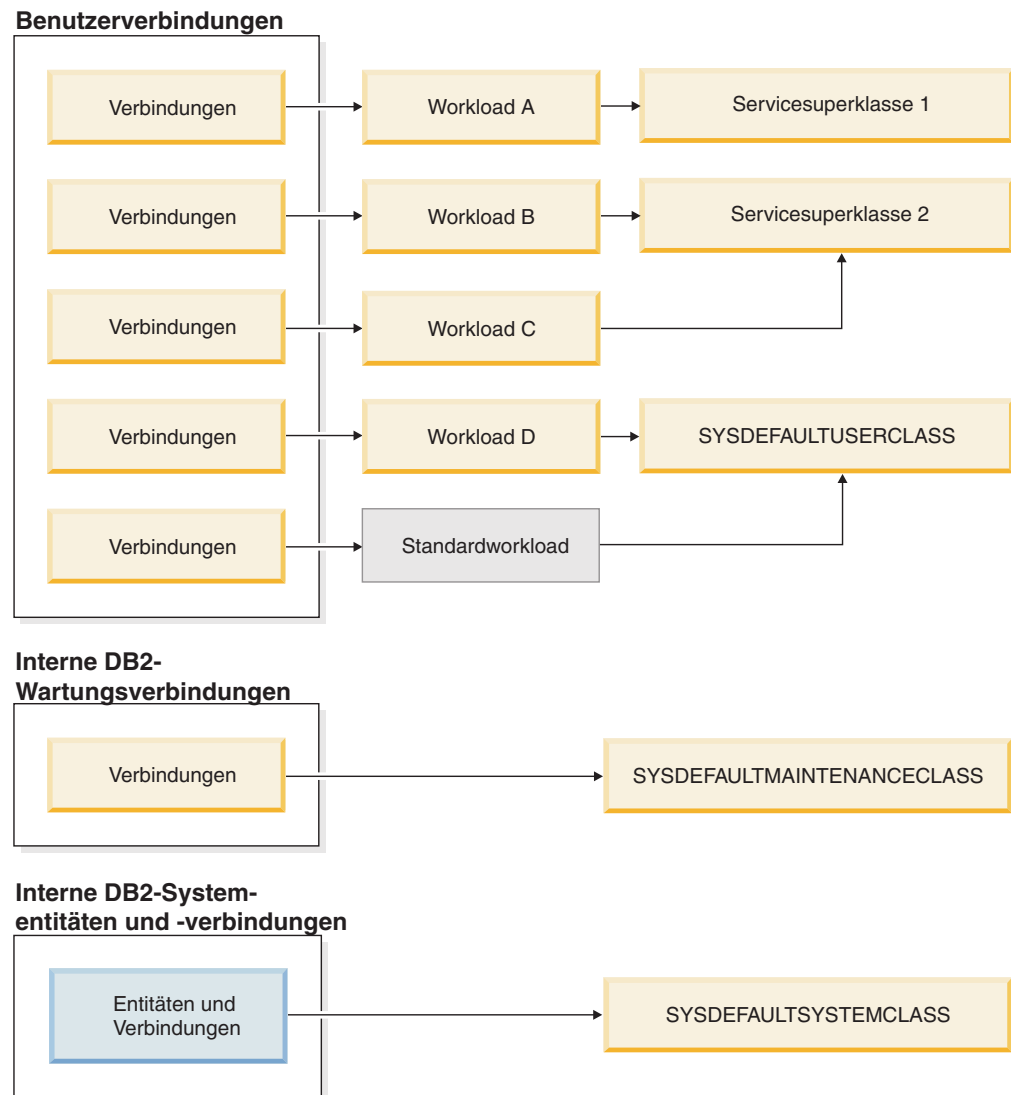


Abbildung 13. Zuordnung von Datenbankverbindungen zu einer Servicesuperklasse

Weitere Unterscheidung zwischen Aktivitäten

Wenn Sie eine komplexere DB2-Workload-Manager-Konfiguration haben, wollen Sie eventuell Aktivitäten basierend auf dem Aktivitätstyp oder einem anderen Aktivitätsattribut unterschiedlich behandeln. Beispielsweise wollen Sie eventuell eine der folgenden Aktionen ausführen:

- DML in eine andere Serviceunterklasse versetzen als DDL.
- Alle Leseabfragen mit einem geschätzten Aufwand von weniger als 100 Timeron in eine andere Serviceunterklasse versetzen als alle anderen Leseabfragen.

In einer komplexeren Konfiguration können Sie die Workload einrichten, um Aktivitäten von der Verbindung der Servicesuperklasse zuzuordnen. Dann können Sie mit Arbeitsaktionen (die in einem Arbeitsaktionsset enthalten sind, das auf die Servicesuperklasse angewendet wird) Aktivitäten basierend auf ihrem Typ oder Attribut bestimmten Serviceunterklassen in einer Servicesuperklasse neu zuordnen.

Insbesondere könnten Sie ein Arbeitsaktionsset, das eine Arbeitsaktion MAP ACTIVITY enthält, auf die Servicesuperklasse anwenden. Alle Aktivitäten, die der Servicesuperklasse zugeordnet sind und einer Arbeitsklasse entsprechen, mit der eine Arbeitsaktion MAP ACTIVITY verknüpft ist, werden der in der Arbeitsaktion angegebenen Serviceunterklasse zugeordnet.

Wenn eine Workloaddefinition sich selbst eine Serviceunterklasse zuordnet, ist keine der Aktivitäten, die durch diese Workload übergeben wird, von einem Arbeitsaktionsset betroffen, das auf die übergeordnete Servicesuperklasse angewendet wird.

- Eine Aktivität kann von einer Workload einer Serviceunterklasse in einer Servicesuperklasse zugeordnet werden.
- Eine Arbeitsaktion, die die Aktivität einer anderen Serviceunterklasse in derselben Servicesuperklasse zuordnet, wird ebenfalls auf die Aktivität angewendet.

Wenn eine Aktivität von einer Workload oder einer Arbeitsaktion nicht einer Serviceunterklasse zugeordnet wird, wird die Aktivität der Standardunterklasse (SYS-DEFAULTSUBCLASS) der Servicesuperklasse für diese Aktivität zugeordnet.

Wenn Datenbankaktivitäten ihren entsprechenden Servicesuperklassen und Serviceunterklassen zugeordnet wurden, können Sie Steuerelemente für alle Aktivitäten in einer bestimmten Serviceklasse implementieren. Auf der Serviceklassenebene sind Statistikdaten verfügbar, die Sie zum Überwachen der Datenbankaktivitäten in dieser Serviceklasse verwenden können.

Die folgende Abbildung zeigt Anforderungen an die Datenbank, die über Workloads einer Servicesuperklasse oder Serviceunterklasse zugeordnet werden. Informationen dazu, wie Arbeitsaktionen für die Zuordnung von Aktivitäten zu einer Serviceunterklasse verwendet werden, finden Sie in „Arbeitsaktionen und Arbeitsaktionssets“ auf Seite 102.

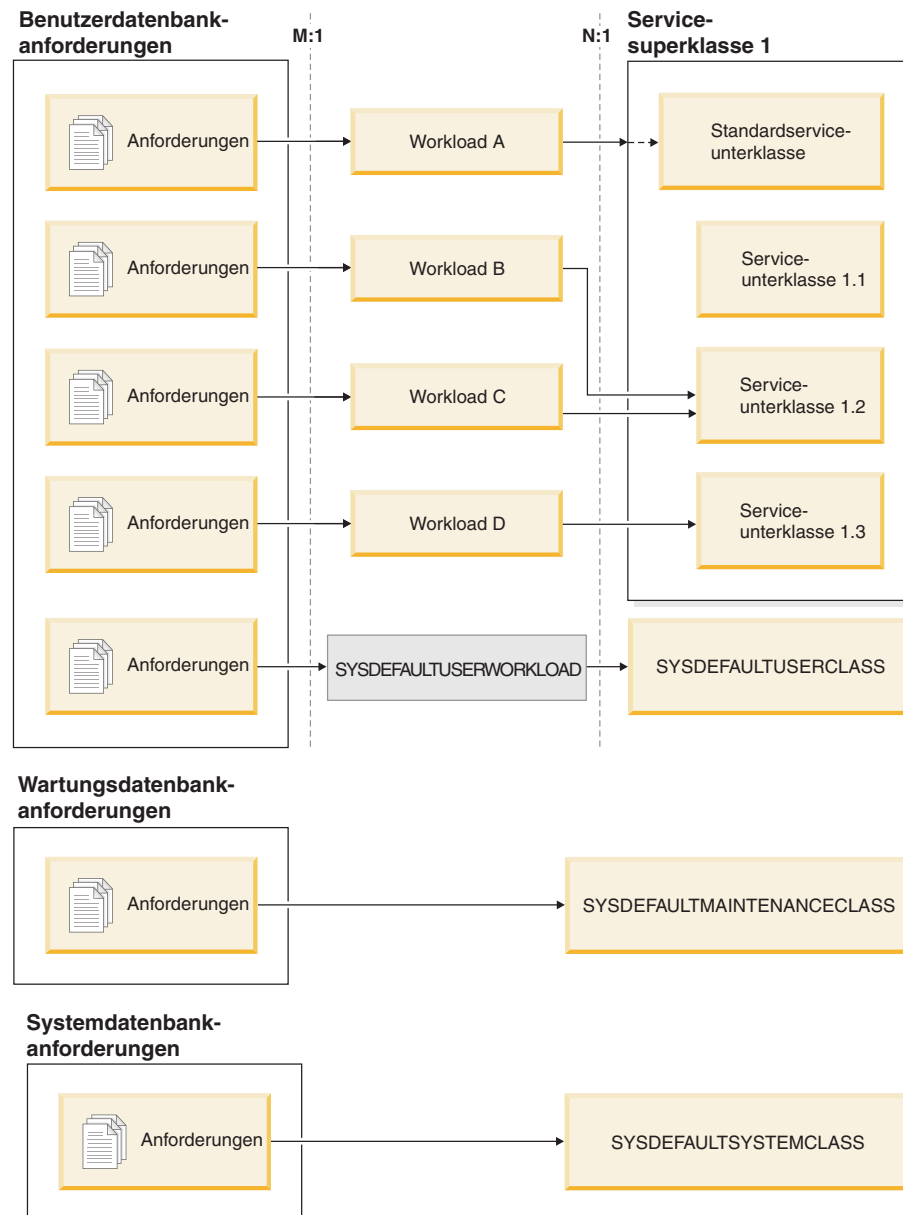


Abbildung 14. Datenbankverbindungen, die einer Servicesuperklasse zugeordnet sind

Verbindungen, die nicht einer benutzerdefinierten Workloaddefinition zugeordnet werden, werden der Standardbenutzerworkloaddefinition (SYSDEFAULTUSERWORKLOAD) zugeordnet. Standardmäßig werden Verbindungen von der Standardworkloaddefinition (SYSDEFAULTUSERWORKLOAD) der Servicesuperklasse SYSDEFAULTUSERCLASS zugeordnet. Dies ist die Standardservicesuperklasse für Benutzeranforderungen. Sie können die Workload SYSDEFAULTUSERWORKLOAD so ändern, dass sie einer anderen Serviceklasse zugeordnet wird. Interne DB2-Wartungsverbindungen werden der Standardservicesuperklasse für Verwaltungsanforderungen (SYSDEFAULTMAINTENANCECLASS) zugeordnet. Interne Systementitäten und Verbindungen werden der Servicesuperklasse SYSDEFAULTSYSTEMCLASS zugeordnet. Dies ist die Standardservicesuperklasse für interne DB2-Verbindungen und -Threads, die Tasks auf Systemebene ausführen.

Agentenpriorität von Serviceklassen

Sie können jeder DB2-Serviceklasse eine relative Agentenpriorität zuordnen, die die Prozessorpriorität auf Ihrem Datenserver steuert. Diese Priorität wird für alle Agenten festgelegt, die in einer Serviceklasse arbeiten, und ist relativ zur Agentenpriorität aller anderen DB2-Agenten.

Wenn Sie keine Agentenpriorität für eine Serviceklasse festlegen, haben alle Agenten in dieser Serviceklasse dieselbe Priorität wie alle anderen DB2-Agenten.

Durch das Festlegen der Agentenpriorität für eine DB2-Serviceklasse wird die Priorität von Agenten nur für neu in der Serviceklasse eintreffende Arbeit angepasst. Threads, die keine Agenten sind und in dieser Serviceklasse ausgeführt werden, verwenden den von Ihnen angegebenen Wert für die Agentenpriorität nicht. Das DB2-Workload-Management weist Arbeit, die in Prozessen im abgeschirmten Modus (FMP – Fenced Mode Process) ausgeführt wird, keine Serviceklassenagentenpriorität zu. Abgeschirmte Prozeduren führen ihre Logik nicht in einer Serviceklasse aus. Diese abgeschirmten Prozeduren werden in den Prozessen im abgeschirmten Modus von DB2 ausgeführt und diese Arbeit wird nicht von DB2-Agenten ausgeführt. Zur Erinnerung: DB2-WLM steuert DB2-Agenten.

Wenn Sie DB2-Serviceklassen in einen Betriebssystem-Workload-Manager wie AIX Workload Manager oder dem Linux-Workload-Management integrieren, können Sie mit dem Betriebssystem-Workload-Manager die Prozessorpriorität angeben, die für die Betriebssystemklasse verwendet werden soll (als Prozessoranteile). Die DB2-Serviceklasse übernimmt dann diesen Wert über den OUTBOUND CORRELATOR-Wert der DB2-Serviceklasse. Die Prozessorpriorität, die Sie über den Betriebssystem-Workload-Manager angeben, steuert die Priorität für Agenten, die in der DB2-Serviceklasse ausgeführt werden. Jede andere Einstellung der Serviceklassenagentenpriorität wird ignoriert.

Anmerkung: Sie können den von einer Teilmenge der Arbeit verwendeten CPU-Anteil auf dem System nicht mithilfe der Agentenpriorität begrenzen. Die Agentenpriorität gibt der CPU-Verarbeitung zwischen konkurrierenden Workloads den Vorzug, wenn diese CPU-Ressourcen erfordern, aber die Agentenpriorität ordnet CPU-Ressourcen nicht explizit zwischen konkurrierenden Workloads zu oder verwaltet sie. Daher ist in Situationen, in denen konkurrierende Workloads in unterschiedlichem Maß CPU-Ressourcen erfordern, die Agentenpriorität in den Fällen ineffektiv, in denen die Arbeit mit niedrigerer Priorität mehr Anforderungen zur Belegung von CPU-Ressourcen stellt als die Arbeit mit höherer Priorität. Die Agentenpriorität ist jedoch effektiv, wenn Sie konkurrierende Workloads haben, die im gleichen Maß CPU-Ressourcen anfordern und Sie der CPU-Belegung nur den Vorzug geben wollen, sie jedoch nicht explizit drosseln oder steuern wollen. Wenn Sie die CPU-Belegung explizit mehr steuern wollen, sind die Linux-WLM-Integration oder die Steuerung des gemeinsamen Zugriffs effektivere Optionen.

Wichtig: Verwenden Sie den veralteten Konfigurationsparameter **agentpri** des Datenbankmanagers nicht in Verbindung mit dem DB2-Workload-Management. Sie können diesen Konfigurationsparameter verwenden, um die absolute Prozessorpriorität aller Agenten in einer DB2-Instanz auf einen festen Wert einzustellen. Wenn Sie jedoch die absolute Priorität für einen Agenten mithilfe von **agentpri** festlegen, können Sie die relative Priorität des Agenten nicht über die Einstellung der Agentenpriorität der DB2-Serviceklasse oder über einen Betriebssystem-Workload-Manager ändern. Wenn Sie **agentpri** festlegen, haben die Agentenpriorität der Serviceklasse und der Betriebssystem-Workload-Manager keinerlei Auswirkung auf die Priorität von Agenten.

Unter UNIX-Betriebssystemen und Linux sind die gültigen Werte DEFAULT und -20 bis 20 (SQLSTATE 42615). Negative Werte kennzeichnen eine höhere relative Priorität. Positive Werte kennzeichnen eine niedrigere relative Priorität.

Unter Windows-Betriebssystemen sind die gültigen Werte DEFAULT und -6 bis 6 (SQLSTATE 42615). Negative Werte kennzeichnen eine niedrigere relative Priorität. Positive Werte kennzeichnen eine höhere relative Priorität.

Unter AIX-Betriebssystemen muss der Instanzeigner über die Funktionen CAP_NUMA_ATTACH und CAP_PROPAGATE verfügen, um mit AGENT PRIORITY eine höhere relative Priorität für Agenten in einer Serviceklasse festzulegen. Melden Sie sich zur Erteilung dieser Funktionen als Root an und führen Sie den folgenden Befehl aus:

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

Unter Solaris 10 oder höher muss der Instanzeigner über das Zugriffsrecht 'proc_prioctl' verfügen, um mit AGENT PRIORITY eine höhere relative Priorität für Agenten in einer Serviceklasse festzulegen. Melden Sie sich zur Erteilung dieses Zugriffsrechts als root an und führen Sie den folgenden Befehl aus:

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

In diesem Beispiel wird 'proc_prioctl' zur Standardgruppe der Zugriffsrechte des Benutzers 'db2user' hinzugefügt.

Wenn DB2 in einer nichtglobalen Zone von Solaris ausgeführt wird, muss das Zugriffsrecht 'proc_prioctl' darüber hinaus der Zugriffsrechtsgruppe 'Limit' (L) für die Zone hinzugefügt werden. Melden Sie sich zur Erteilung dieses Zugriffsrechts an der Zone als root an und führen Sie den folgenden Befehl aus:

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

In diesem Beispiel wird 'proc_prioctl' zur Zugriffsrechtsgruppe 'Limit' (L) für die Zone 'db2zone' hinzugefügt.

Unter Solaris 9 gibt es keine Funktion für DB2, mit der die relative Priorität von Agenten erhöht werden kann. Aktualisieren Sie auf Solaris 10 oder höher, um die Agentenpriorität der Serviceklasse zu verwenden.

Vorablesezugriffspriorität von Serviceklassen

Vorablesefunktionen rufen Daten vom Datenträger ab und speichern diese Daten in Pufferpools, sodass Agenten schnell darauf zugreifen können. Jeder Servicesuperklasse und -unterklasse kann eine andere Vorablesezugriffspriorität zugeordnet werden.

Agenten senden Vorableseanforderungen an die Vorablesewarteschlange der Datenbank. Die Vorablesefunktionen nehmen diese Vorableseanforderungen aus der Warteschlange und legen dann die Daten in den Pufferpools ab. Wenn ein Agent bestimmte Daten benötigt, prüft er zuerst die Pufferpools, um festzustellen, ob die Daten dort verfügbar sind. Ist dies nicht der Fall, ruft der Agent die Daten vom Datenträger ab. Vorablesefunktionen führen aufwendige Platten-E/A-Operationen aus, sodass die Agenten parallel für die Ausführung von Rechenarbeit frei werden.

Für jede Verbindung, die an eine Serviceklasse weitergeleitet wird, werden die Vorableseanforderungen entsprechend der Vorablesezugriffspriorität verarbeitet, die der Serviceklasse zugeordnet ist. Jeder Serviceklasse kann eine von drei Vorablese-

zugriffsprioritäten zugeordnet werden: HIGH, MEDIUM und LOW. Sie legen die Vorablesezugriffspriorität einer Serviceklasse mit dem Schlüsselwort PREFETCH PRIORITY in der Anweisung CREATE SERVICE CLASS oder ALTER SERVICE CLASS fest.

Durch Angabe von DEFAULT für eine Servicesuperklasse wird für diese Servicesuperklasse als Vorablesezugriffspriorität MEDIUM (mittel) festgelegt. Sie können eine andere Vorablesezugriffspriorität für jede Serviceunterklasse in der Servicesuperklasse angeben. Wenn Sie jedoch die Standardpriorität für den Vorablesezugriff für die Serviceunterklasse verwenden, übernimmt die Serviceunterklasse die Vorablesezugriffspriorität von ihrer Servicesuperklasse.

Vorablesezugriffsanforderungen mit hoher Priorität werden vor Vorablesezugriffsanforderungen mit mittlerer Priorität verarbeitet und diese wiederum vor Vorablesezugriffsanforderungen mit niedriger Priorität. Die Vorablesezugriffspriorität beeinflusst die Reihenfolge, in der Vorablesezugriffsanforderungen verarbeitet werden. Sie wirkt sich jedoch nicht auf die Geschwindigkeit ihrer Verarbeitung aus.

Pufferpoolpriorität von Serviceklassen

Durch das Festlegen der Pufferpoolpriorität von Serviceklassen können Sie den Anteil von Seiten im Pufferpool beeinflussen, der von Aktivitäten in einer bestimmten Serviceklasse belegt sein kann. Hierdurch kann der Durchsatz und die Leistung von Aktivitäten in dieser Serviceklasse verbessert werden.

Sie können jeder DB2-Serviceklasse eine relative Pufferpoolpriorität zuweisen, die die Wahrscheinlichkeit steuert, mit der Seiten ausgelagert werden, die von Aktivitäten in einer Serviceklasse in den Pufferpool abgerufen wurden. Die Erhöhung der Pufferpoolpriorität vergrößert möglicherweise den Anteil der Seiten, die von Agenten einer bestimmten Serviceklasse verwendet werden.

Wenn Sie keine Pufferpoolpriorität angeben oder wenn Sie BUFFERPOOL PRIORITY DEFAULT angeben, wird der Serviceklasse die Standard-Pufferpoolpriorität zugewiesen. Bei Servicesuperklassen entspricht DEFAULT dem Wert LOW, bei Serviceunterklassen dem Wert der Pufferpoolpriorität der übergeordneten Servicesuperklasse. Alle Standardunterklassen haben eine Pufferpoolpriorität von DEFAULT, die nicht geändert werden kann.

Wenn Sie ein Upgrade für eine frühere DB2-Version durchführen, wird die Pufferpoolpriorität vorhandener Serviceklassen auf DEFAULT gesetzt.

Nutzen der Vorteile der Festlegung der Pufferpoolpriorität

Sie erreichen eher eine Leistungssteigerung durch das Festlegen der Pufferpoolpriorität für eine Serviceklasse, wenn im Pufferpool eine gewisse Anzahl von Konkurrenzsituationen besteht. Der größte Vorteil ergibt sich wahrscheinlich bei Konkurrenzsituationen im Pufferpool mit einer Gesamttrefferquote von max. 85 %. Wenn die Gesamttrefferquote 90 % übersteigt, bestehen wahrscheinlich keine nennenswerten Konkurrenzsituationen im Pufferpool, sodass das Festlegen der Pufferpoolpriorität in den meisten Fällen weniger oder nur geringen Nutzen bringt. Welchen Vorteil Sie erzielen, hängt vom Typ der Workload ab, die Ihr Datenserver ausführt.

Bei einigen Workloads ist das Festlegen der Pufferpoolpriorität effektiver, wenn Sie auch die proaktive Seitenbereinigung aktivieren. Dies kommt daher, dass die Ein-

stellungen für die Pufferpoolpriorität nur für nicht genutzte Seiten effektiv ist und die proaktive Seitenbereinigung hinsichtlich des Auslagerns von genutzten Seiten auf Datenträger aggressiver ist. Beachten Sie, dass Sie die proaktive Seitenbereinigung nur aktivieren sollten, wenn dies eine Leistungssteigerung bewirkt.

Wenn Sie asynchrone Seitenbereinigung (auch als klassische Seitenbereinigung bezeichnet) verwenden und einen niedrigeren Wert für den Datenbankkonfigurationsparameter **chngpgs_thresh** festlegen, können Ihre Einstellungen für die Pufferpoolpriorität ebenfalls einen positiven Effekt auf die Leistung haben, da ein niedriger Wert für diesen Parameter auch sicherstellt, dass ausreichend ungenutzte Seiten im Pufferpool vorhanden sind.

Es ist möglich, dass die positiven Auswirkungen der Festlegung der Pufferpoolpriorität von den Auswirkungen des Vorablesezugriffs - mit oder ohne Festlegung der Vorablesezugriffspriorität - übertroffen werden, wenn eine nennenswerte Anzahl von Vorablesezugriffen stattfindet. Wenn Sie z. B. bei nur wenig Vorablesezugriffen eine Serviceklasse mit einer hohen Pufferpoolpriorität definieren, kann der tatsächliche Vorteil dieser Einstellung für die Pufferpoolpriorität gering sein im Vergleich zu einer Serviceklasse mit einer niedrigen Pufferpoolpriorität, bei der Aktivitäten jedoch eine nennenswerte Anzahl von Vorablesezugriffen durchführen. Durch die Vorteile des Vorablesezugriffs kann die Leistung der Aktivitäten in der Serviceklasse mit der niedrigen Pufferpoolpriorität sogar noch besser sein als die der Aktivitäten in der Serviceklasse mit der hohen Pufferpoolpriorität. Das Festlegen der Pufferpoolpriorität kann Ihre Workload-Management-Strategie unter den genannten Umständen dennoch ergänzen und sollte daher auch eingesetzt werden.

Status von Verbindungen und Aktivitäten in einer Serviceklasse

Serviceklassen erfassen Verbindungsstatistikdaten für jede Serviceklasse. Sie können sehen, welche Verbindungen und Aktivitäten in einer Serviceklasse enthalten sind, und sich den Status der Verbindung oder Aktivität ansehen.

Status einer Verbindung

Eine Verbindung in einer Serviceklasse kann die folgenden Status haben:

CONNECTED

Die Verbindung wurde erfolgreich mit der Datenbank verbunden, ist aber noch nicht mit der Workload und der Servicesuperklasse verknüpft.

DECOUPLED

Der Verbindung ist kein Koordinatoragent zugewiesen (Konzentratorfall).

DISCONNECTPEND

Die Verbindung wird von der Datenbank getrennt.

FORCED

Die Verbindung wurde erzwungen.

INTERRUPTED

Die Verbindung wurde unterbrochen.

MAPPED

Die Verbindung ist einer Workload zugeordnet und ist mit einer Servicesuperklasse verknüpft. Die Verbindung kann jetzt Aktivitäten zur Ausführung übergeben.

QUEUED

Der Verbindungsordinatoragent wurde aufgrund eines DB2-Schwellen-

werts für Verbindungen oder Aktivitäten (z. B. CONCURRENTDBCOORDACTIVITIES), der die Warteschlangensteuerung ermöglicht, in die Warteschlange gestellt. In einer Datenbankumgebung mit mehreren Membern kann dieser Status darauf hindeuten, dass der Koordinatoragent einen Prozedurfernaufruf an ein anderes Member abgesetzt hat, um Schwellenwerttickets zu erhalten, und noch keine Antwort erhalten hat.

TRANSIENT

Die Verbindung versucht, eine Serviceklasse zu verknüpfen, für die der Verbindungsschwellenwert erreicht ist. Die Verbindung wird für die Verknüpfung mit der Serviceklasse in die Warteschlange gestellt. Wenn die Serviceklasse nicht gegen ihren Verbindungsschwellenwert verstößt, wird die Verbindung mit der Serviceklasse verknüpft. Eine Verbindung mit dem Status TRANSIENT kann keine Aktivitäten zur Ausführung übergeben.

TERMINATING

Die Verbindung hat eine Verbindungsrücksetzung vom Client empfangen oder wird zwangsweise oder aufgrund einer Fehlerbedingung beendet.

UOWEXEC

Die Verbindung verarbeitet eine Anforderung.

UOWWAIT

Die Verbindung wartet auf eine Anforderung vom Client.

Status einer Aktivität

Eine Aktivität in einer Serviceklasse kann die folgenden Status haben:

CANCEL_PENDING

Wenn Sie eine Aktivität abbrechen, die keinen Agenten hat, der aktiv an einer Anforderung für die Aktivität arbeitet, erhält die Aktivität den Status CANCEL_PENDING und wird beim Empfang der nächsten Anforderung abgebrochen.

EXECUTING

Die Aktivität wird ausgeführt.

IDLE Kein Agent verarbeitet aktiv eine Anforderung für die Aktivität.

INITIALIZING

Die Aktivität wurde erstellt und wird zur Ausführung vorbereitet.

QUEUED

Die Aktivität kann aufgrund eines Schwellenwerts für gemeinsamen Zugriff auf Datenbank- oder Serviceklassenebene nicht ausgeführt werden. Die Aktivität wird in die Warteschlange gestellt, bis sie ausgeführt werden kann.

TERMINATING

Die Aktivität wird beendet.

UNKNOWN

Der Status der Aktivität ist unbekannt.

Nicht von Serviceklassen verwaltete Entitäten auf Systemebene

Serviceklassen werden zur Überwachung und Steuerung von Objekten auf Datenbankebene verwendet. Nicht alle DB2-Entitäten arbeiten jedoch direkt in einer Datenbank.

Da Serviceklassen in einer Datenbank arbeiten und in den Katalogtabellen der Datenbank gespeichert werden, können Entitäten, die nicht in einer Datenbank arbeiten, nicht von Serviceklassen verwaltet werden. Entitäten auf Instanzebene, z. B. die Systemcontroller- und Diagnoseмонитordämonen, arbeiten auf Instanzebene und sind mit keiner Datenbank direkt verknüpft. Agenten, die Instanzzuordnungen und Gatewayverbindungen durchführen, werden ebenfalls nicht von Serviceklassen verwaltet. Da Instanzzuordnungsagenten und Gatewayagenten nicht in einer Datenbank arbeiten, werden sie nicht von Serviceklassen verwaltet.

Die folgende Liste ist eine nicht vollständige Liste von Entitäten, die nicht innerhalb einer Datenbank arbeiten und daher nicht von Serviceklassen verwaltet werden:

- DB2-Systemcontroller (db2sysc)
- IPC-Listener (db2ipccm)
- TCP-Listener (db2tcpcom)
- FCM-Dämonen (db2fcms, db2fcmr)
- DB2-Resynchronisationsagenten (db2resync)
- Inaktive Agenten (Agenten ohne Datenbankzuordnung)
- Instanzzuordnungsagenten
- Gatewayagenten
- Alle anderen EDUs (Engine Dispatchable Units - von der Steuerkomponente zuteilbare Einheiten) auf Instanzebene

Erstellen einer Serviceklasse

Mit der DDL-Anweisung `CREATE SERVICE CLASS` können Sie Servicesuperklassen und darunter Serviceunterklassen erstellen.

Vorbereitende Schritte

Zum Erstellen einer Serviceklasse benötigen Sie die Berechtigung `WLMADM` oder `DBADM`.

Weitere Voraussetzungen werden auch in den folgenden Abschnitten behandelt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Anhang A, „Allgemeine Namenskonventionen“, auf Seite 541

Vorgehensweise

Gehen Sie wie folgt vor, um eine Serviceklasse zu erstellen:

1. Geben Sie mindestens eine der folgenden Eigenschaften für die Serviceklasse in der Anweisung `CREATE SERVICE CLASS` an:

- Geben Sie den Namen der Serviceklasse an:

Anmerkung: Sobald der Name einer Serviceklasse festgelegt ist, kann er nicht mehr geändert werden.

- Wenn Sie eine Servicesuperklasse erstellen, muss der Name unter allen Servicesuperklassen in der Datenbank eindeutig sein.

Beim Erstellen einer Servicesuperklasse wird die zugehörige Standardserviceunterklasse automatisch erstellt. Erst nach dem Erstellen einer Servicesuperklasse können Sie darunter andere Serviceunterklassen erstellen.

- Wenn Sie eine Serviceunterklasse erstellen, muss der Name unter allen Serviceunterklassen in der Servicesuperklasse eindeutig sein. Eine Serviceunterklasse kann nicht denselben Namen haben wie die zugehörige Servicesuperklasse.
- Wenn Sie eine Serviceunterklasse erstellen, geben Sie den Namen der übergeordneten Servicesuperklasse an. Nachdem eine Serviceunterklasse unter einer Servicesuperklasse erstellt wurde, kann sie nicht mehr einer anderen Servicesuperklasse zugeordnet werden.
- Geben Sie die Pufferpoolpriorität für die Serviceklasse an. Dies wirkt sich auf die Wahrscheinlichkeit aus, mit der von Aktivitäten in der Serviceklasse abgefangene Seiten ausgelagert werden. Bei Servicesuperklassen wird der Wert DEFAULT intern dem LOW zugeordnet. Auf DEFAULT gesetzte Serviceunterklassen übernehmen die Pufferpoolpriorität von ihren übergeordneten Superklassen.
- Geben Sie die Priorität für den Vorablesezugriff an. Sie können die Priorität angeben, mit der Agenten in der Serviceklasse ihre Vorablesezugriffsanforderungen übergeben. Je nach angegebenem Wert werden die Vorablesezugriffsanforderungen an die Vorablesewarteschlange mit hoher, mittlerer oder niedriger Priorität weitergeleitet. Standardmäßig ist die Vorablesezugriffspriorität auf MEDIUM eingestellt.
- Geben Sie die abgehende Korrelatorzeichenfolge an, wenn Sie die DB2-Serviceklasse einer AIX- oder Linux-Klasse zuordnen wollen. Ein Nullwert gibt an, dass es keine Zuordnung zum Workload-Manager des Betriebssystems gibt.
Wenn der abgehende Korrelator festgelegt ist, werden alle Threads in der DB2-Serviceklasse dem Betriebssystem-Workload-Manager mithilfe des abgehenden Korrelators zu Beginn der nächsten Aktivität zugeordnet.
Wenn der abgehende Korrelator für eine Serviceunterklasse auf NONE gesetzt ist und der abgehende Korrelator für die zugehörige Servicesuperklasse festgelegt ist, übernimmt die Serviceunterklasse den für die Servicesuperklasse angegebenen abgehenden Korrelator.
- Geben Sie die zu erfassenden Aktivitätsdaten an. Wenn die Erfassung von Aktivitätsdaten aktiviert ist, werden am Ende der Aktivität Aktivitätsinformationen von dem Koordinatormember an den entsprechenden Ereignismonitor gesendet. Sie können Daten in den Ereignismonitor schreiben, einschließlich Informationen zur ausgeführten Anweisung, ihrer Kompilierungsumgebung sowie alle zutreffenden Eingabedatenwerte. Sie können auch angeben, dass keine Aktivitätsdaten erfasst werden sollen. Standardmäßig werden keine Aktivitätsdaten erfasst.
- Geben Sie die zu erfassenden zusammengefassten Aktivitätsinformationen an. Die zusammengefassten Aktivitätsinformationen für die Serviceklasse ändern sich erst nach dem Durchführen eines Commits für die Operation CREATE SERVICE CLASS.
- Der Typ von Anforderungsmessdaten, die für Anforderungen erfasst werden sollen, die von einer Verbindung übergeben wurden, die der von Ihnen angegebenen Servicesuperklasse zugeordnet wurde. Standardmäßig werden immer Basismessdaten für Aktivitäten erfasst, die der Workload zugeordnet wurden.
- Geben Sie die Histogrammschablonen an, die die Serviceunterklasse als Schablonen für ihre Histogramme verwenden soll. Die angegebenen Histogrammschablonen werden in der Sicht SYSCAT.HISTOGRAMTEMPLATEUSE wi-

dergespiegelt. Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

- Geben Sie an, ob die Serviceklasse aktiviert oder inaktiviert ist.
 - Wenn eine Serviceklasse als aktiviert erstellt wird (Standard), können der Serviceklasse Verbindungen und Aktivitäten zugeordnet werden. Wenn eine Serviceklasse als inaktiviert erstellt wird, werden neue Verbindungen und Aktivitäten, die ihr zugeordnet werden sollen, zurückgewiesen.
 - Wenn Sie eine Servicesuperklasse als inaktiviert erstellen, verhalten sich alle Serviceunterklassen, die Sie mit dieser Servicesuperklasse verknüpfen, als wären sie inaktiviert, selbst wenn sie bei der Abfrage der Sicht SYSCAT.SERVICECLASSES als aktiviert angezeigt werden.
- 2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Serviceklasse zur Sicht SYSCAT.SERVICECLASSES hinzugefügt.

Ändern einer Serviceklasse

Verwenden Sie die Anweisung ALTER SERVICE CLASS, wenn Sie die Definition einer Serviceklasse ändern wollen.

Vorbereitende Schritte

Zum Ändern einer Serviceklasse benötigen Sie die Berechtigung SQLADM, WLMADM oder DBADM. Soll eine andere Klausel als die Klausel COLLECT angegeben werden, muss die Berechtigungs-ID die Berechtigung WLMADM oder DBADM aufweisen.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Aktivitäten, die bereits Ressourcen bezogen haben und ausgeführt werden, sind in der Regel von der Anweisung ALTER SERVICE CLASS nicht betroffen. Diese Aktivitäten halten ihre Ressourcen und werden bis zu ihrer Fertigstellung ausgeführt. (Beachten Sie, dass Änderungen an den CPU-Anteilen oder CPU-Grenzwerten einer Serviceklasse, die durch die Anweisung ALTER SERVICE CLASS implementiert werden, sofort aktiv sind. Dies gilt auch für Aktivitäten, die bereits ausgeführt werden.) Wenn jedoch während einer Operation ALTER SERVICE CLASS eine Subagentenanforderung an ein fernes Member gesendet wird, ist es möglich, dass der Koordinatoragent und der Subagent unterschiedliche Serviceklassendefinitionen sehen. Betrachten wir das folgende Beispiel, bei dem die Vorablesezugriffspriorität für die Serviceklasse anfänglich auf MEDIUM eingestellt ist:

Tabelle 28. Unterschiede zwischen den Sichten eines Koordinatoragenten und eines Subagenten für eine geänderte Serviceklasse

Ereignisreihenfolge	Verbindung 1	Verbindung 2
1	Der koordinierende Agent sendet eine Anforderung an das ferne Member. (Die Vorablesezugriffspriorität der Serviceklasse wurde zuvor auf MEDIUM gesetzt.)	

Tabelle 28. Unterschiede zwischen den Sichten eines Koordinatoragenten und eines Subagenten für eine geänderte Serviceklasse (Forts.)

Ereignisreihenfolge	Verbindung 1	Verbindung 2
2		ALTER SERVICE CLASS wird abgesetzt. Die Vorablesezugriffspriorität wird auf HIGH gesetzt.
3		COMMIT wird abgesetzt. (Die geänderte Serviceklasseneigenschaft wird im Katalogmember festgeschrieben und in allen Datenbankmembern in den Hauptspeicher geladen.)
4	Der ferne Subagent empfängt die Anforderung. Jetzt legt der Subagent die neue Vorablesezugriffspriorität HIGH für die Serviceklassendefinition fest.	

Die in der Tabelle oben beschriebene Situation ist temporär und betrifft nur Verbindungen, die während der Operation ALTER SERVICE CLASS Subagentenanforderungen absetzen. Alle neue Verbindungen sehen die aktualisierte Serviceklassendefinition mit der Vorablesezugriffspriorität HIGH.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Serviceklasse zu ändern:

1. Geben Sie mindestens eines der folgenden Eigenschaften für die Serviceklasse in der Anweisung ALTER SERVICE CLASS an:
 - Geben Sie an, ob die Serviceklasse aktiviert oder inaktiviert ist. Wenn Sie eine Serviceklasse von aktiviert in inaktiviert ändern, bleiben alle bestehenden Verbindungen oder Aktivitäten in der Serviceklasse und verwenden bis zu ihrer Fertigstellung weiterhin die zuvor zugeordneten Ressourcen. Sie können eine Serviceklasse inaktivieren, wenn die Arbeit, die in der Serviceklasse eingeht, das System überlastet oder wenn Sie alle in der Serviceklasse eingehende Arbeit zurückweisen wollen.

Beim Inaktivieren einer Servicesuperklasse geschieht Folgendes:

- a. Die Servicesuperklasse wird inaktiviert.
- b. Die zugehörigen Serviceunterklassen werden inaktiviert.

Die Serviceunterklassen sind nur inaktiviert, während ihre Servicesuperklasse inaktiviert ist. Wenn die Servicesuperklasse aktiviert wird, werden die Serviceunterklassen auf ihren vorherigen Zustand gemäß der Definition in der Katalogtabelle zurückgesetzt.

Wenn eine Serviceunterklasse inaktiviert wird, sind davon weder die zugehörige Servicesuperklasse noch andere Serviceunterklassen mit dieser Servicesuperklasse betroffen.

Eine Standard-Serviceunterklasse kann nicht explizit inaktiviert werden. Sie müssen die zugehörige Servicesuperklasse inaktivieren, um zu verhindern, dass neue Anforderungen unter einer Standard-Serviceunterklasse ausgeführt werden.

- Geben Sie die Agentenpriorität für die Serviceklasse an. Wenn die Agentenpriorität auf DEFAULT eingestellt ist, erhalten Agenten in der Serviceklasse die Priorität, die das Betriebssystem allen DB2-Threads zuordnet. Wenn Sie den Parameter AGENT PRIORITY auf einen anderen Wert als DEFAULT setzen, werden die Agententhreads auf eine Priorität entsprechend der Standardpriorität plus dem Wert gesetzt, der am Anfang der nächsten Aktivität festgelegt wird. Wenn die Standardpriorität z. B. den Wert 20 hat und Sie die Agentenpriorität auf -10 setzen, ergibt sich eine Agentenpriorität von $20 + (-10) = 10$.

In der Katalogsicht SYSCAT.SERVICECLASSES wird die Standardagentenpriorität als -32768 dargestellt.

Unter Linux und UNIX sind die Werte -20 bis 20 zulässig. (Ein negativer Wert bedeutet eine höhere relative Priorität.) Unter Windows-Betriebssystemen sind die Werte -6 bis 6 zulässig. (Ein negativer Wert bedeutet eine niedrigere relative Priorität.)

- Geben Sie die Priorität für den Vorablesezugriff an. Sie können die Priorität angeben, mit der Agenten in der Serviceklasse ihre Vorablesezugriffsanforderungen übergeben. Je nach angegebenem Wert werden die Vorablesezugriffsanforderungen an die Vorablesewarteschlange mit hoher, mittlerer oder niedriger Priorität weitergeleitet. Standardmäßig ist die Vorablesezugriffspriorität auf MEDIUM eingestellt. Wenn die Vorablesezugriffspriorität nach einer Vorablesezugriffsanforderung geändert wird, ändert die Anforderung ihre Priorität nicht.
- Geben Sie die Pufferpoolpriorität für die Serviceklasse an. Dies wirkt sich auf die Wahrscheinlichkeit aus, mit der von Aktivitäten in der Serviceklasse abgefangene Seiten ausgelagert werden. Bei Servicesuperklassen wird der Wert DEFAULT intern dem LOW zugeordnet. Auf DEFAULT gesetzte Serviceunterklassen übernehmen die Pufferpoolpriorität von ihren übergeordneten Superklassen.
- Geben Sie die abgehende Korrelatorzeichenfolge an, wenn Sie die DB2-Serviceklasse einer AIX- oder Linux-Klasse zuordnen wollen. Ein Nullwert gibt an, dass es keine Zuordnung zum Workload-Manager des Betriebssystems gibt.

Wenn der abgehende Korrelator in einen Nullwert geändert wird, heben alle Threads in der DB2-Serviceklasse am Anfang der nächsten Aktivität ihre Zuordnung zum Betriebssystem-Workload-Manager auf.

Wenn der abgehende Korrelator für eine Serviceunterklasse auf NONE gesetzt ist und der abgehende Korrelator für die zugehörige Servicesuperklasse festgelegt ist, übernimmt die Serviceunterklasse den für die Servicesuperklasse angegebenen abgehenden Korrelator.

Wenn eine Servicesuperklasse einen abgehenden Korrelator verwendet, muss die Agentenpriorität der Servicesuperklasse auf DEFAULT gesetzt werden.

Wenn eine Serviceunterklasse einen abgehenden Korrelator verwendet (entweder explizit als Teil der Serviceunterklassendefinition oder implizit durch Übernahme von der Servicesuperklasse), muss die Agentenpriorität der Serviceunterklasse auf DEFAULT gesetzt werden.

- Geben Sie das Hinzufügen bzw. Löschen eines Tabellenbereichs für temporäre Systemtabellen zu bzw. aus der Liste bevorzugter Tabellenbereiche für temporäre Tabellen für die angegebene Serviceunterklasse an.
- Geben Sie die zu erfassenden Aktivitätsdaten an. Wenn die Erfassung von Aktivitätsdaten aktiviert ist, werden am Ende der Aktivität Aktivitätsinformationen von dem Koordinatormember an den entsprechenden Ereignismonitor gesendet. Sie können Daten in den Ereignismonitor schreiben, ein-

schließlich Informationen zur ausgeführten Anweisung, ihrer Kompilierungsumgebung sowie alle zutreffenden Eingabedatenwerte. Sie können auch angeben, dass keine Aktivitätsdaten erfasst werden sollen. Standardmäßig werden keine Aktivitätsdaten erfasst.

- Geben Sie die zu erfassenden zusammengefassten Aktivitätsinformationen an. Die zusammengefassten Aktivitätsinformationen für die Serviceklasse ändern sich erst nach dem Durchführen eines Commits für die Operation ALTER SERVICE CLASS.
 - Geben Sie die Stufe der Erfassung von Messdaten für die Überwachung von Anforderungen an, die von Verbindungen übergeben wurden, die einer Unterklasse unter der angegebenen Servicesuperklasse zugeordnet sind. Beachten Sie, dass die effektive Erfassungseinstellung für unter einer Servicesuperklasse ausgeführte Anforderungen eine Kombination der Erfassungsstufe für Serviceklassen und die Einstellung des Datenbankkonfigurationsparameters **mon_req_metrics** ist.
 - Geben Sie an, ob die Histogramme geändert werden sollen, die von einer Serviceunterklasse verwendet werden, für die die Erfassung von zusammengefassten Aktivitätsdaten über COLLECT AGGREGATE ACTIVITY DATA oder die Erfassung von zusammengefassten Anforderungsdaten über COLLECT AGGREGATE REQUEST DATA aktiviert ist. Durch die Aktualisierung der von einer Serviceunterklasse verwendeten Histogrammschablonen werden die entsprechenden Zeilen in der Sicht SYSCAT.HISTOGRAMTEMPLATEUSE aktualisiert, die die Histogrammschablonen anzeigt, auf die eine Serviceklasse oder eine Arbeitsaktion verweist. Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Serviceklasse in der Sicht SYSCAT.SERVICECLASSES aktualisiert.

Löschen einer Serviceklasse

Mit der DDL-Anweisung DROP SERVICE CLASS können Sie Serviceklassen löschen.

Vorbereitende Schritte

Zum Löschen einer Serviceklasse benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Die Standard-servicesuperklassen (SYSDEFAULTUSERCLASS, SYSDEFAULTMAINTENANCECLASS und SYSDEFAULTSYSTEMCLASS) und ihre zugehörigen Serviceunterklassen können nicht gelöscht werden. Standard-servicesuperklassen und ihre zugehörigen Serviceunterklassen können nur durch das Löschen der Datenbank gelöscht werden.

Eine von Ihnen definierte Serviceklasse kann nicht gelöscht werden, wenn eine der folgenden Bedingungen zutrifft:

- Sie ist aktiviert.
- Sie enthält benutzerdefinierte Serviceunterklassen.
- Eine Workload, eine Arbeitsaktion oder ein Schwellenwert verweist auf sie.

- Ein Workloadvorkommen verweist noch auf sie.
- Der Serviceklasse ist zurzeit eine Verbindung oder Aktivität zugeordnet.
- Die Serviceklasse ist als Ziel der Aktion REMAP ACTIVITY definiert.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Serviceklasse zu löschen:

1. Ändern Sie Workloads so, dass sie nicht auf die Serviceklasse verweisen, oder inaktivieren Sie die Workloaddefinitionen. Alternativ können Sie mit der Anweisung DROP WORKLOAD alle Workloads löschen, die der Serviceklasse zugeordnet sind. Setzen Sie nach dem Löschen jeder Workload eine Anweisung COMMIT ab.
Aktivitäten, die bereits unter der Serviceklasse ausgeführt werden, werden weiter ausgeführt. Mit der Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS können Sie Agenten auflisten, die zurzeit einer Serviceklasse zugeordnet sind. Wenn diese Aktivitäten nicht abgeschlossen werden sollen, können Sie anhand der von der Tabellenfunktion zurückgegebenen Anwendungs-ID und mit dem Befehl **FORCE APPLICATION** die Trennung dieser Anwendungen von der Datenbank erzwingen.
2. Löschen Sie alle zutreffenden Arbeitsaktionen, die der zu löschenden Serviceklasse zugeordnet sind:
 - Wenn Sie eine Servicesuperklasse löschen und dieser ein Arbeitsaktionsset zugeordnet ist, inaktivieren und löschen Sie dieses Arbeitsaktionsset. Mit der Anweisung ALTER WORK ACTION SET können Sie ein Arbeitsaktionsset inaktivieren und mit der Anweisung DROP WORK ACTION SET können Sie das Arbeitsaktionsset löschen. Setzen Sie nach dem Löschen des Arbeitsaktionssets eine Anweisung COMMIT ab.
 - Wenn Sie eine Serviceunterklasse löschen und dieser eine Arbeitsaktion zugeordnet ist, löschen Sie die Arbeitsaktion mit der Klausel DROP WORK ACTION der Anweisung ALTER WORK ACTION SET. Alternativ können Sie das Arbeitsaktionsset, das die der Serviceunterklasse zugeordnete Arbeitsaktion enthält, mit der Anweisung DROP WORK ACTION SET löschen. Setzen Sie nach dem Löschen jeder Arbeitsaktion oder nach dem Löschen des Arbeitsaktionssets eine Anweisung COMMIT ab.
3. Inaktivieren und löschen Sie alle Schwellenwerte, die der zu löschenden Serviceklasse zugeordnet sind. Setzen Sie nach dem Löschen jedes Schwellenwerts eine Anweisung COMMIT ab. Inaktivieren Sie einen Schwellenwert mit der Anweisung ALTER THRESHOLD und löschen Sie den Schwellenwert mit der Anweisung DROP THRESHOLD.
4. Gehen Sie je nach dem zu löschenden Objekt wie folgt vor:
 - Wenn Sie eine Serviceunterklasse löschen wollen, verwenden Sie die Anweisung DROP SERVICE CLASS zum Löschen der Serviceunterklasse.
 - Wenn Sie eine Servicesuperklasse löschen wollen, verwenden Sie die Anweisung DROP SERVICE CLASS, um alle ihr zugeordneten Serviceunterklassen zu löschen, und setzen Sie nach dem Löschen jeder Serviceunterklasse eine Anweisung COMMIT ab. Setzen Sie dann die Anweisung DROP SERVICE CLASS ab, um die Servicesuperklasse zu löschen.

Anmerkung: Die Standardserviceunterklasse für die Servicesuperklasse kann nicht manuell gelöscht werden. Die Standardserviceunterklasse für eine Servicesuperklasse wird beim Löschen der Servicesuperklasse gelöscht.
5. Inaktivieren Sie die Serviceklasse mit der Anweisung ALTER SERVICE CLASS. Wenn Sie eine Servicesuperklasse löschen, werden mit dieser Aktion alle Ser-

viceunterklassen inaktiviert, die mit der Servicesuperklasse verknüpft sind. Durch das Inaktivieren einer Serviceklasse wird verhindert, dass weitere Aktivitäten mit der Serviceklasse verknüpft werden. Setzen Sie nach dem Inaktivieren der Serviceklasse eine Anweisung COMMIT ab.

6. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Serviceklasse aus der Sicht SYSCAT.SERVICECLASSES entfernt.

Beispiel: Verwenden von Serviceklassen

Das folgende Beispiel zeigt, wie die Datenbankworkload mithilfe von Serviceklassen gesteuert werden kann.

Dieses Beispiel stammt aus dem fiktiven International Beer Emporium. International Beer Emporium ist ein mittelständisches Unternehmen mit im Wesentlichen fünf Abteilungen: Vertrieb, Buchhaltung, Entwicklung, Test und Produktion. Alle fünf Abteilungen greifen gemeinsam auf dieselbe Produktkatalogdatenbank zu.

Erstimplementierung einer DB2-Workload-Management-Lösung

Die Produktkatalogdatenbank läuft meistens zufrieden stellend. Manchmal beklagen sich die Benutzer jedoch, dass ihre Anwendungen keine Verbindung zur Datenbank herstellen können, da die maximale Anzahl der Verbindungen überschritten wurde. Nachdem er ein Upgrade auf DB2 Version 9.7 durchgeführt hat, beschließt der Datenbankadministrator Bob, Serviceklassen auszuprobieren. Bob will die Verwendungsmuster der Produktkatalogdatenbank aller fünf Abteilungen ermitteln und herausfinden, warum seiner Datenbank gelegentlich die Verbindungen ausgehen. Im Folgenden sind die Schritte aufgeführt, die Bob zum Einrichten von Serviceklassen ausführt:

1. Zuerst erstellt er Servicesuperklassen für jede der Abteilungen (für jede Servicesuperklasse wird automatisch auch die Standardserviceunterklasse erstellt):
 - SALES wird für die Vertriebsabteilung erstellt:
CREATE SERVICE CLASS SALES
 - ACCOUNTING wird für die Buchhaltungsabteilung erstellt:
CREATE SERVICE CLASS ACCOUNTING
 - ENGINEERING wird für die Entwicklungsabteilung erstellt:
CREATE SERVICE CLASS ENGINEERING
 - TESTING wird für die Testabteilung erstellt:
CREATE SERVICE CLASS TESTING
 - PRODUCTION wird für die Produktionsabteilung erstellt:
CREATE SERVICE CLASS PRODUCTION
2. Bob erstellt Sitzungsbenutzergruppen mit entsprechenden Berechtigungs-IDs für jede der Abteilungen:
 - Eine Sitzungsbenutzergruppe mit der Berechtigungs-ID SALESGRP wird erstellt. Diese Gruppe umfasst die Berechtigungs-IDs aller Benutzer in der Vertriebsabteilung.
 - Eine Sitzungsbenutzergruppe mit der Berechtigungs-ID ACCTNGRP wird erstellt. Diese Gruppe umfasst die Berechtigungs-IDs aller Benutzer in der Buchhaltungsabteilung.
 - Eine Sitzungsbenutzergruppe mit der Berechtigungs-ID ENGINGRP wird erstellt. Diese Gruppe umfasst die Berechtigungs-IDs aller Benutzer in der Entwicklungsabteilung.

- Eine Sitzungsbenutzergruppe mit der Berechtigungs-ID TESTGRP wird erstellt. Diese Gruppe umfasst die Berechtigungs-IDs aller Benutzer in der Testabteilung.
 - Eine Sitzungsbenutzergruppe mit der Berechtigungs-ID PRODGRP wird erstellt. Diese Gruppe umfasst die Berechtigungs-IDs aller Benutzer in der Produktionsabteilung.
3. Bob erstellt Workloads, um Verbindungen von jeder Gruppe der zugehörigen Serviceklasse zuzuordnen:
- Workload WL_SALES wird erstellt und ihre Sitzungsbenutzergruppe wird auf SALESGRP eingestellt. WL_SALES ordnet ihre Verbindungen der Service- superklasse SALES zu:

```
CREATE WORKLOAD WL_SALES SESSION_USER GROUP ('SALESGRP') SERVICE CLASS SALES
```
 - Workload WL_ACCOUNTING wird erstellt und ihre Sitzungsbenutzergruppe wird auf ACCTNGRP eingestellt. WL_ACCOUNTING ordnet ihre Verbindungen der Servicesuperklasse ACCOUNTING zu:

```
CREATE WORKLOAD WL_ACCOUNTING SESSION_USER GROUP ('ACCTNGRP') SERVICE CLASS ACCOUNTING
```
 - Workload WL_ENGINEERING wird erstellt und ihre Sitzungsbenutzergruppe wird auf ENGINGRP eingestellt. WL_ENGINEERING ordnet ihre Verbindungen der Serviceklasse ENGINEERING zu:

```
CREATE WORKLOAD WL_ENGINEERING SESSION_USER GROUP ('ENGINGRP') SERVICE CLASS ENGINEERING
```
 - Workload WL_TEST wird erstellt und ihre Sitzungsbenutzergruppe wird auf TESTGRP eingestellt. WL_TEST ordnet ihre Verbindungen der Serviceklasse TESTING zu:

```
CREATE WORKLOAD WL_TEST SESSION_USER GROUP ('TESTGRP') SERVICE CLASS TESTING
```
 - Workload WL_PRODUCTION wird erstellt und ihre Sitzungsbenutzergruppe wird auf PRODGRP eingestellt. WL_PRODUCTION ordnet ihre Verbindungen der Serviceklasse PRODUCTION zu:

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

Bob nutzt die Standardeinstellungen für Serviceklassen und Workloads. Er will die Datenbanknutzungsmuster beobachten, bevor er eine Steuerung auf die Serviceklassen anwendet. Daraus ergeben sich die folgenden Definitionen für die Service- superklassen:

Tabelle 29. Serviceklassendefinitionen

Serviceklasse
SALES
ACCOUNTING
ENGINEERING
TESTING
PRODUCTION
SYSDEFAULTUSERCLASS
SYSDEFAULTMAINTENANCECLASS
SYSDEFAULTSYSTEMCLASS

Wenn eine DB2-Workload-Management-Lösung wie oben beschrieben implementiert ist, wird Arbeit von jeder Abteilung an die eigene Servicesuperklasse weiter-

geleitet. Arbeit von Abteilungen, für die keine speziellen Vorgaben gemacht wurden, wird der Standardservicesuperklasse SYSDEFAULTUSERCLASS zugeordnet. Mit dieser Konfiguration kann Bob die Arbeit in den einzelnen Serviceklassen überwachen, um das Datenbanknutzungsmuster der Abteilungen zu ermitteln.

Erste Verbesserung der DB2-Workload-Management-Implementierung

Nach der letzten Verbindungsspitze fragt Bob die Servicesuperklassenstatistik mit der Tabellenfunktion WLM_GET_SERVICE_SUPERCLASS_STATS ab und untersucht den oberen Grenzwert für jede Servicesuperklasse. Bob erkennt, dass die obere Grenze für Verbindungen für alle Abteilungen mit Ausnahme der Testabteilung nahe 100 liegt. Die Statistikdaten für die Testabteilung zeigen jedoch, dass das Testteam einmal über 800 Verbindungen herstellte.

Einmal pro Monat führt die Testabteilung ihre allmonatliche intensive Produktprüfung durch. Zu dieser Zeit stellt die Abteilung bis zu 1000 gleichzeitig bestehende Verbindungen her. Da der Konfigurationsparameter **max_connections** des Datenbankmanagers auf 1000 eingestellt ist, nutzt die Testabteilung die meisten der verfügbaren Verbindungen zur Datenbank. Wenn das System 1000 Verbindungen erreicht hat, werden alle weiteren Verbindungsversuche abgewiesen.

Aufgrund von Speichereinschränkungen auf dem System können die Konfigurationenwerte **max_connections** und **maxagents** auf dem Datenserver nicht erhöht werden, um weitere Verbindungen zuzulassen. Um zu verhindern, dass die Testabteilung alle Verbindungen nutzt, beschließt Bob, die Anzahl der Verbindungen von der Testabteilung zu beschränken und sicherzustellen, dass jede der anderen vier Abteilungen ausreichend viele Verbindungen zur Datenbank erhalten kann, um ihre Geschäftsziele erfüllen zu können.

Die anderen vier Abteilungen benötigen in der Regel jeweils nicht mehr als 150 gleichzeitig bestehende Verbindungen. Außerdem bemerkt Bob auch, dass die Standardbenutzer-, die Standardverwaltungs- und die Standardssystemservicesuperklasse nur selten Verbindungen enthalten. Er beschließt deshalb, dass 100 Verbindungen für diese Standardservicesuperklassen ausreichen sollten. Nachdem 700 Verbindungen (600 für die vier Abteilungen und 100 für die Standardklassen) aus dem Pool **max_connections** mit 1 000 verfügbaren Verbindungen zugeordnet wurden, stehen noch 300 Verbindungen für die Testabteilung zur Verfügung. Durch die Beschränkung der Testabteilung auf maximal 300 Verbindungen, sollten Verbindungen von Benutzern anderer Abteilungen nicht abgewiesen werden.

Zur Beschränkung der Testgruppe auf maximal 300 gleichzeitig bestehende Verbindungen erstellt Bob einen Schwellenwert MAXSERVICECLASSCONNECTIONS von 300 für die Serviceklasse TESTING.

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

Nach der Implementierung dieser Änderung sieht die DB2-Workload-Management-Konfiguration wie folgt aus:

Tabelle 30. Konfiguration nach dem Hinzufügen des Schwellenwerts für die Servicesuperklasse TESTING

Serviceklasse	Schwellenwert MAXSERVICECLASSCONNECTIONS
SALES	Nicht zutreffend

Tabelle 30. Konfiguration nach dem Hinzufügen des Schwellenwerts für die Servicesuperklasse TESTING (Forts.)

Serviceklasse	Schwellenwert MAXSERVICECLASSCONNECTIONS
ACCOUNTING	Nicht zutreffend
ENGINEERING	Nicht zutreffend
TESTING	300
PRODUCTION	Nicht zutreffend
SYSDEFAULTUSERCLASS	Nicht zutreffend
SYSDEFAULTMAINTENANCECLASS	Nicht zutreffend

Da die Serviceklasse TESTING nur maximal 300 gleichzeitig bestehende Verbindungen enthalten kann, werden alle Verbindungsanforderungen über diesen Schwellenwert hinaus abgewiesen. Auf die anderen Serviceklassen wird kein Schwellenwert MAXSERVICECLASSCONNECTIONS angewendet, sodass diese Serviceklassen die verbleibenden 700 verfügbaren Verbindungen zum Datenserver gemeinsam nutzen können. Da es keine Konkurrenzsituation für Verbindungen zwischen diesen Serviceklassen gibt, legt Bob keine Verbindungsschwellenwerte für diese fest.

Zweite Verbesserung der DB2-Workload-Management-Implementierung

Obwohl Verbindungen von der Vertriebs-, Buchhaltungs-, Entwicklungs- und Produktionsabteilung nicht mehr abgewiesen werden, beklagen sich Benutzer aus diesen Abteilungen immer noch über die schlechte Leistung, wenn die Testabteilung intensive Produkttests durchführt. Bob untersucht die Abfragen der Testabteilung während des Produkttestzyklus und erkennt, dass die Abfragen komplizierte Joins mit großen Datenmengen enthalten. Diese Abfragen generieren eine beträchtliche Vorablesezugriffsaktivität, wodurch die Verarbeitung von Vorablesezugriffsanforderungen von Verbindungen anderer Abteilungen verhindert wird. Bob beschließt, die Vorablesezugriffspriorität der Verbindungen von der Testabteilung zu verringern, und ändert die Vorablesezugriffspriorität der Serviceklasse TESTING auf LOW:

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

Die DB2-Workload-Management-Konfiguration sieht nun wie folgt aus:

Tabelle 31. Konfiguration nach dem Ändern der Vorablesezugriffspriorität für die Servicesuperklasse TESTING

Serviceklasse	Schwellenwert MAXSERVICECLASSCONNECTIONS	Vorablesezugriffspriorität
SALES	Nicht zutreffend	DEFAULT
ACCOUNTING	Nicht zutreffend	DEFAULT
ENGINEERING	Nicht zutreffend	DEFAULT
TESTING	300	LOW
PRODUCTION	Nicht zutreffend	DEFAULT
SYSDEFAULTUSERCLASS	Nicht zutreffend	DEFAULT
SYSDEFAULTMAINTENANCECLASS	Nicht zutreffend	DEFAULT

Indem die Vorablesezugriffspriorität der Serviceklasse TESTING auf LOW gesetzt wird, werden Vorablesezugriffsanforderungen von Verbindungen von der Testabteilung erst bedient, nachdem alle anderen Vorablesezugriffsanforderungen von den anderen Abteilungen verarbeitet wurden. Diese Änderung erhöht den Abfragedurchsatz der anderen Abteilungen und verringert den Durchsatz der Testabteilung in ihrer Produkttestphase.

Dritte Verbesserung der DB2-Workload-Management-Implementierung

Nachdem das Problem mit dem Vorablesezugriff behoben ist, teilt die Entwicklungsabteilung Bob mit, dass sie einige Verbindungen für eine experimentelle Anwendung namens Brewmeister benötigt. Da es sich um eine experimentelle Anwendung handelt, will Bob sicherstellen, dass sie nicht zu viele Datenbankverbindungen verbraucht und dass Abfragen von der Anwendung nicht um Vorablesefunktionen konkurrieren, wenn das System ausgelastet ist. Um diese Ziele zu erreichen, erstellt er eine neue Serviceunterklasse unter der Servicesuperklasse ENGINEERING für die experimentelle Anwendung sowie eine Workload, um Verbindungen von der Anwendung der neuen Serviceunterklasse zuzuordnen. Bob aktualisiert die Serviceklasse und die Workloads wie folgt:

- Die Serviceunterklasse EXPERIMENT wird unter der Servicesuperklasse ENGINEERING erstellt:
CREATE SERVICE CLASS EXPERIMENT UNDER ENGINEERING
- Der Schwellenwert MAXSERVICECLASSCONNECTIONS von 50 wird für die Serviceunterklasse EXPERIMENT erstellt:
CREATE THRESHOLD MAXSERVICECLASSCONNECTIONS FOR SERVICE CLASS EXPERIMENT UNDER ENGINEERING ACTIVITIES ENFORCEMENT DATABASE WHEN TOTALMEMBERCONNECTIONS > 50 STOP EXECUTION
- Die Workload WL_EXPERIMENT wird erstellt, um Verbindungen von der Anwendung BREWMEISTER der Serviceunterklasse EXPERIMENT zuzuordnen:
CREATE WORKLOAD WL_EXPERIMENT APPLNAME ('BREWMEISTER') SERVICE CLASS EXPERIMENT UNDER ENGINEERING
- Die Vorablesezugriffspriorität für die Serviceunterklasse EXPERIMENT wird auf LOW gesetzt:
ALTER SERVICE CLASS EXPERIMENT UNDER ENGINEERING PREFETCH PRIORITY LOW

Die DB2-Workload-Management-Konfiguration sieht nun wie folgt aus:

Tabelle 32. Konfiguration mit Serviceunterklasse EXPERIMENT

Serviceklasse	Schwellenwert MAXSERVICECLASSCONNECTIONS	Vorablesezugriffspriorität
SALES	Nicht zutreffend	DEFAULT
ACCOUNTING	Nicht zutreffend	DEFAULT
ENGINEERING	Nicht zutreffend	DEFAULT
EXPERIMENT	50	LOW
TESTING	300	LOW
PRODUCTION	Nicht zutreffend	DEFAULT
SYSDEFAULTUSERCLASS	Nicht zutreffend	DEFAULT
SYSDEFAULTMAINTENANCECLASS	Nicht zutreffend	DEFAULT

Mit dieser Konfiguration kann die Anwendung BREWMEISTER nur 50 gleichzeitig bestehende Verbindungen zur Datenbank halten. Darüber hinaus werden Vorablesezugriffsanforderungen von dieser Anwendung an die Vorablesewarteschlange mit niedriger Priorität gesendet. Die Entwicklungsabteilung kann jetzt auf sichere Weise mit der Anwendung experimentieren und kann davon ausgehen, dass nicht unbeabsichtigt das gesamte Datenbanksystem überlastet wird.

Beispiel: Analysieren einer serviceklassenbezogenen Leistungsminderung des Systems

Wenn Sie eine Leistungsminderung des Systems bemerken (z. B. dauert die Ausführung einiger Anwendungen länger als erwartet) und Sie nicht sicher sind, ob das Problem mit der Konfiguration der Serviceklassen zusammenhängt, können Sie mit Tabellenfunktionsdaten das Problem untersuchen und erforderlichenfalls beheben.

Zuerst erhalten Sie eine komplexe Übersicht, was in den Serviceklassen abläuft. Diese Übersicht sollte die durchschnittliche Aktivitätslaufzeit, die Anzahl der normal abgeschlossenen Aktivitäten und die obere Grenze für gleichzeitig ablaufende Koordinatoraktivitäten auf dem System enthalten. Dazu können Sie unter Verwendung der mithilfe der Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS abgefragten Daten eine allgemeine Abfrage erstellen, bei der die Zusammenfassung serviceklassen- und datenbankmemberübergreifend erfolgt. Definieren Sie für das erste und das zweite Argument leere Zeichenfolgen und geben Sie für das dritte Argument -2 (ein Platzhalterzeichen) ein, um anzugeben, dass Daten für alle Serviceklassen in allen Datenbankpartitionen zusammengestellt werden sollen. Ihre Abfrage könnte der folgenden ähneln:

```
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
```

Nehmen wir an, dass bei früheren Gelegenheiten die Abfrage die folgenden Ergebnisse zurückgegeben hat:

SUPERCLASS_NAME	SUBCLASS_NAME	ACTSCOMPLETED	ACTSABORTED	ACTSHW	ACTAVGLIFETIME
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	8	0	1	3.750
BI_APPS	SYSDEFAULTSUBCLASS	4	0	1	5.230
BATCH	SYSDEFAULTSUBCLASS	1	0	1	25.600

Die von dieser Abfrage zurückgegebenen Daten können ausreichen, um zu zeigen, dass die Leistungsminderung in der Serviceklasse BI_APPS auftritt, da ihre durchschnittliche Aktivitätslaufzeit signifikant höher als normal ist. Diese Situation könnte darauf hindeuten, dass die verfügbaren Ressourcen für diese Serviceklasse erschöpft sind.

Wenn anhand der Durchschnittswerte für die Serviceklassen für alle Datenbankmember das Problem nicht eingegrenzt werden kann, sollten Sie Durchschnittswerte für jedes Member einzeln analysieren. Durch Zusammenfassung der Durchschnittswerte für jedes Member zu einem globalen Durchschnitt können große Abweichungen zwischen Datenbankmembers verdeckt werden. In dieser Situation

wird angenommen, dass jedes Member als Koordinatormember verwendet wird. Wenn diese Annahme nicht stimmt, beträgt die berechnete durchschnittliche Laufzeit in Nicht-Koordinatormemberrn null.

```
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

In diesem Beispiel empfängt das Member 2 möglicherweise mehr Arbeit als gewöhnlich, da ihre durchschnittliche Aktivitätslaufzeit viel höher ist als die der anderen Datenbankmember.

Viele unterschiedliche Situationen können die Ursache einer Leistungsminderung sein. Verwenden Sie die folgenden Prinzipien, um die von den DB2-Workload-Management-Tabellenfunktionen bereitgestellten Informationen bestmöglich zu nutzen:

- Behandeln Sie große Mengen von Sperrkonflikten auf der Ebene der Anwendungslogik und -umgebung (Isolationsstufe usw.).
- Wenn sich die Serviceklasse ihren Schwellenwerten nähert (Anzahl gleichzeitiger Anforderungen usw.), müssen Sie eventuell die Schwellenwerte erhöhen.
- Wenn die einer Serviceklasse zugeordneten Ressourcen nahezu erschöpft sind und OUTBOUND CORRELATOR festgelegt ist, kann die Zuordnung zu den Betriebssystemserviceklassen die Ursache des Problems sein (d. h., die der Serviceklasse entsprechende Betriebssystemserviceklasse erhält nicht ausreichend Prozessorressourcen).
- Möglicherweise werden mehr Aktivitäten als erwartet in der Serviceklasse ausgeführt, wodurch mehr Ressourcen als normal verbraucht werden. Prüfen Sie die Anzahl abgeschlossener Aktivitäten, um festzustellen, ob die Menge der ausgeführten Arbeit in der Serviceklasse angemessen ist.
- Aktivitäten verbringen möglicherweise mehr Zeit in Warteschlangen, wenn mehr Aktivitäten als erwartet übergeben werden und Schwellenwerte für gemeinsamen Zugriff definiert sind. Prüfen Sie, ob die durchschnittliche Wartezeit sich um denselben Betrag erhöht hat wie die durchschnittliche Laufzeit. Wenn dies der Fall ist, verhalten sich die Warteschlangen wie erwartet. Wenn jedoch die Laufzeit inakzeptabel ist, sollten Sie der Serviceklasse mehr Ressourcen zuweisen und den Schwellenwert für gemeinsamen Zugriff verringern.

Beispiel: Untersuchen der Agentenauslastung nach Serviceklasse

Das DB2-Workload-Management stellt die Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS zur Verfügung, mit der Sie die relative Verteilung von Agenten auf die Serviceklassen ermitteln können.

Es können Situationen auftreten, in denen eine Datenserverressource, z. B. ein Agent, von einer Gruppe von Benutzern oder von einer Anwendung zu stark ver-

wendet wird. Nehmen wir z. B. an, dass eine Benutzergruppe fast alle verfügbaren Agenten nutzt und dass ein Benutzer, der nicht zu dieser Gruppe gehört, sich bei Ihnen darüber beklagt.

Als Erstes muss ermittelt werden, wie viele Agenten für jede Serviceklasse arbeiten. Sie können eine Abfrage wie die folgende verwenden:

```
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

Wenn Sie zu dem Schluss kommen, dass eine bestimmte Serviceklasse mehr als den ihr zustehenden Anteil von Agenten nutzt, können Sie Maßnahmen ergreifen, um die Anzahl der zulässigen Aktivitäten für eine Workload oder eine Serviceklasse zu beschränken. Alternativ können Sie die Anzahl der Verbindungen für eine Serviceklasse beschränken.

Anwenden von Steuerelementen auf Typen von Aktivitäten mit Arbeitsaktionssets

Arbeitsaktionssets enthalten Arbeitsaktionen, die Steuerelemente auf Aktivitäten eines bestimmten Typs in einer bestimmten Servicesuperklasse, in einer bestimmten Workload oder in der Datenbank als Ganzes anwenden.

Eine Arbeitsaktion stellt eine Aktion zur Verfügung, die auf eine Arbeitsklasse angewendet werden kann, die Aktivitäten eines bestimmten Typs wie LOAD- oder READ-Aktivitäten darstellt. Arbeitsaktionen werden auf die Aktivitäten angewendet, die zu der Arbeitsklasse gehören, auf die die Arbeitsaktion vor der Ausführung der Aktivität angewendet wurde. Wenn die Arbeitsaktion jedoch PREVENT EXECUTION lautet, wird verhindert, dass die Aktivität ausgeführt wird, und es wird keine weitere Arbeitsaktion angewendet.

Wenn Sie ein Arbeitsaktionsset auf eine Datenbank anwenden, gibt es mehrere Arten von Aktionen, die Sie auf Aktivitäten anwenden können, die unter eine Arbeitsklasse fallen (z. B. Schwellenwertdefinitionen, Stoppen der Ausführung, Erfassen von Aktivitätsdaten und Zählen der Aktivitäten). Das Definieren eines Schwellenwerts für eine Arbeitsaktion ist die leistungsfähigste Datenbankarbeitsaktion. Vielleicht wollen Sie verhindern, dass SQL-Code mehr als 100.000 Zeilen liest und zurückgibt. Sie können eine einzige Arbeitsklasse für ein Arbeitsaktionsset definieren, das SQL-READ-Anweisungen erkennt, und eine Arbeitsaktion mit einem Schwellenwert, der die Ausführung stoppen würde, wenn mehr als 100.000 Zeilen zurückgegeben würden. Weitere Informationen zu möglichen Aktionen finden Sie in „Arbeitsaktionen und die Domäne des Arbeitsaktionssets“ auf Seite 114.

Wenn Sie ein Arbeitsaktionsset auf eine Workload anwenden, können Sie unterschiedliche Arten von Aktionen auf Aktivitäten anwenden, darunter das Definieren von Schwellenwerten, das Verhindern der Ausführung, das Erfassen von Aktivitäts- oder zusammengefassten Aktivitätsdaten und das Zählen der Aktivitäten.

Wenn Sie das Arbeitsaktionsset für eine Servicesuperklasse definieren, können Sie unterschiedliche Arten von Aktionen auf Aktivitäten anwenden, darunter die Zuordnung von Aktivitäten zu einer Serviceunterklasse, das Verhindern der Ausführung, das Erfassen von Aktivitäts- oder zusammengefassten Aktivitätsdaten und das Zählen der Aktivitäten. Typischerweise ordnet die Arbeitsaktion eine Aktivität einer Serviceunterklasse zu und hat für die Unterklasse definierte Schwellenwerte, um die Aktivität zu verwalten.

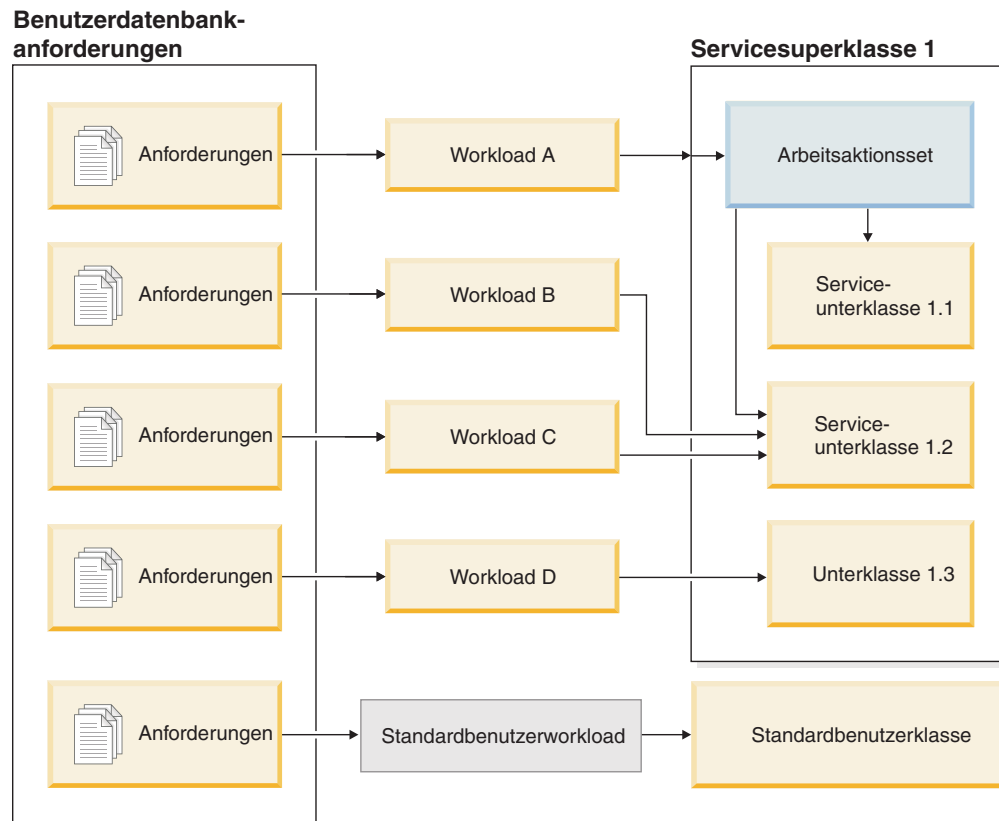


Abbildung 15. Zuordnung eines Arbeitsaktionssets für eine Servicesuperklasse

Beziehung zwischen Arbeitsklassen, Arbeitsklassensets, Arbeitsaktionen und Arbeitsaktionssets und deren Zuordnung zu anderen DB2-Objekten

Arbeitsklassen und Arbeitsaktionen arbeiten zusammen, um bestimmte Aktionen bestimmten Aktivitätstypen zuzuordnen. Dies wird am besten anhand eines Beispiels erläutert.

Das folgende Diagramm zeigt eine abstrakte Sicht der Beziehung zwischen Arbeitsklassen, Arbeitsklassensets, Arbeitsaktionen und Arbeitsaktionssets und deren Zuordnung zu anderen DB2-Objekten.

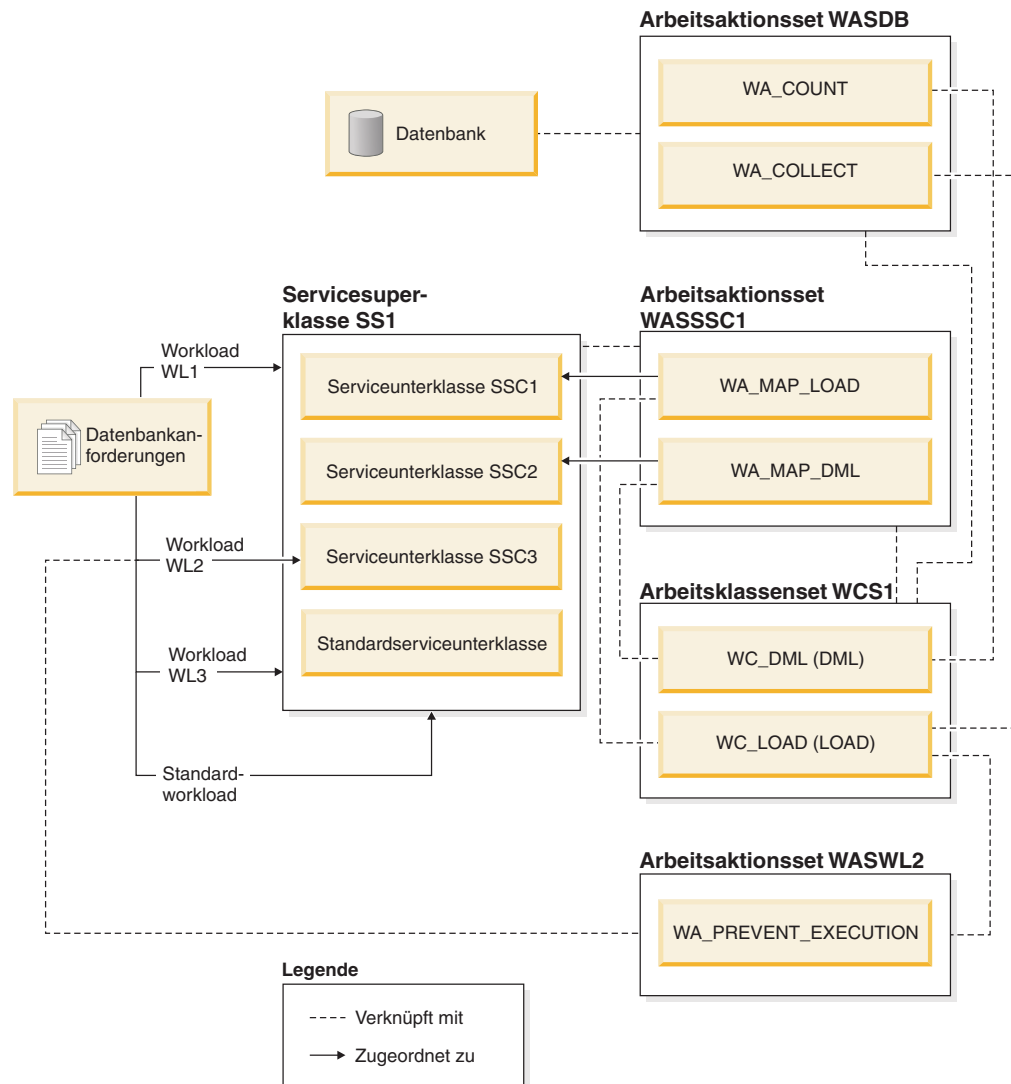


Abbildung 16. Überblick über Arbeitsaktionssets und Arbeitsklassensets

Im Diagramm werden einige Datenbankaktivitäten über die Workload WL1, die Workload WL3 und die Standardbenutzerworkload (SYSDEFAULTUSERWORKLOAD) der Servicesuperklasse SS1 zugeordnet. Da das Arbeitsaktionsset WASDB auf die Datenbank angewendet wird, werden die Arbeitsaktionen im Arbeitsaktionsset WASDB auf alle Aktivitäten angewendet, die (unabhängig von der Workload) in die Datenbank eingetragen werden und unter die Arbeitsklassen WC_DML oder WC_LOAD fallen. Das heißt, Aktivitäten mit dem Arbeitstyp DML werden gezählt und für Aktivitäten mit dem Arbeitstyp LOAD werden Aktivitätsdaten erfasst und in einen aktiven Ereignismonitor (sofern vorhanden) geschrieben.

Das Arbeitsaktionsset WASSSC1 wird auf die Servicesuperklasse SS1 angewendet. Die Arbeitsaktionen WA_MAP_DML und WA_MAP_LOAD werden auch auf alle Aktivitäten angewendet, die der Standardbenutzerworkload, der Workload WL1 oder der Workload WL3 zugeordnet sind und unter die Arbeitsklasse WC_DML oder WC_LOAD fallen, weil diese Workloads Arbeit an die Servicesuperklasse SS1 übertragen. Das heißt, Aktivitäten mit dem Arbeitstyp LOAD werden von der Arbeitsaktion WA_MAP_LOAD der Serviceunterklasse SSC1 zugeordnet und Aktivitäten mit dem Arbeitstyp DML werden von der Arbeitsaktion WA_MAP_DML der Serviceunterklasse SSC2 zugeordnet.

Der Workload WL2 zugeordnete Aktivitäten werden direkt einer Serviceunterklasse (SSC3) zugeordnet. Wenn eine Workload Aktivitäten direkt einer Serviceunterklasse zuordnet, werden keine Arbeitsaktionen aus dem Arbeitsaktionsset WASSSC1 auf diese Aktivitäten angewendet. Da WASWL2 jedoch auf WL2 angewendet wird, werden auf Aktivitäten, die WL2 zugeordnet sind und unter WL_LOAD fallen, Arbeitsaktionen im Arbeitsaktionsset WASWL2 angewendet. Dies bedeutet, dass die Ausführung von LOAD-Aktivitäten aufgrund der Arbeitsaktion PREVENT EXECUTION nicht zulässig ist.

Arbeitsaktionen und Arbeitsaktionssets

Eine Arbeitsaktion, die in Verbindung mit einer Arbeitsklasse verwendet wird, kann zur Steuerung bestimmter Aktivitätstypen eingesetzt werden. Sie können beispielsweise unterschiedliche Arbeitsaktionen auf LOAD-Aktivitäten anwenden, so dass sie anders als DML verarbeitet werden. Arbeitsaktionen werden in Arbeitsaktionssets gruppiert.

Arbeitsaktionen

Eine Arbeitsaktion hat die folgenden Attribute:

- Einen benutzerdefinierten Arbeitsaktionsnamen, der im Arbeitsaktionsset eindeutig sein muss.
- Die Arbeitsklassen-ID, auf die die Arbeitsaktion angewendet wird. Sie können für eine Arbeitsklasse mehrere Arbeitsaktionen definieren, aber jede Arbeitsaktion muss in dieser Arbeitsklasse eine andere Aktion ausführen.
- Die Aktion, die auf die Datenbankaktivität angewendet werden soll, die mit der Arbeitsklasse übereinstimmt. Welcher Aktionstyp für eine Arbeitsaktion gültig ist, hängt davon ab, ob das Arbeitsaktionsset, zu dem die Arbeitsaktion gehört, auf eine Datenbank, Workload oder Servicesuperklasse angewendet wird. Wenn ein Arbeitsaktionsset auf eine Datenbank angewendet wird, wird die gesamte Arbeit, die in der Datenbank eingeht, durch das Arbeitsaktionsset bewertet. Wenn ein Arbeitsaktionsset auf eine Workload angewendet wird, wird die gesamte Arbeit, die durch dieser Workload zugeordnete Workloadvorkommen übergeben wird, durch das Arbeitsaktionsset bewertet. Wenn ein Arbeitsaktionsset auf eine Servicesuperklasse angewendet wird, wird die gesamte Arbeit, die zur Ausführung direkt an die Servicesuperklasse übergeben wird, durch das Arbeitsaktionsset bewertet. Das heißt, für Workloaddefinitionen, in denen die Servicesuperklasse spezifisch als Zielserviceklasse angegeben ist, wird die übergebene Arbeit durch das Arbeitsaktionsset bewertet, das für die Servicesuperklasse definiert ist. Beispiel:
 - Ein Arbeitsaktionsset, das auf eine Datenbank oder Workload angewendet wird, kann Schwellenwertarbeitsaktionen enthalten. Wenn eine Aktivität einer Arbeitsklasse zugeordnet wird, für die eine Schwellenwertarbeitsaktion definiert ist, wird der Schwellenwert auf diese Aktivität angewendet.
 - Ein Arbeitsaktionsset, das auf eine Servicesuperklasse angewendet wird, kann eine Arbeitsaktion enthalten, die die Aktivität einer Serviceunterklasse in der Servicesuperklasse zuordnet. Wenn eine Aktivität einer bestimmten Arbeitsklasse im Arbeitsklassenset entspricht und das Arbeitsaktionsset eine Zuordnungsarbeitsaktion hat, die für diese Arbeitsklasse definiert ist, wird diese Aktivität der von der Arbeitsaktion definierten Serviceunterklasse zugeordnet.

Eine Liste der unterstützten Aktionen finden Sie in „Arbeitsaktionen und die Domäne des Arbeitsaktionssets“ auf Seite 114.

- Ein Objekt, das das Ziel der angegebenen Aktion ist. Je nach Aktion kann das Objekt eine Serviceunterklasse sein, der die Aktivität zugeordnet wird, ein

Schwellenwert, der angibt, welcher Schwellenwert auf die Aktivität anzuwenden ist (bzw. null, wenn die Aktion die Ausführung verhindern soll), eine der Erfassungsaktionen oder eine Zählaktivität.

- Die Schablone, die das Histogramm mit den erfassten Statistikdaten zur Dauer (in Millisekunden) der Aktivitäten beschreibt, die mit der dieser Arbeitsaktion zugeordneten Arbeitsklasse verknüpft sind und in einem bestimmten Intervall ausgeführt werden müssen. Diese Informationen werden nur erfasst, wenn der Arbeitsaktionstyp COLLECT AGGREGATE ACTIVITY DATA (entweder BASE oder EXTENDED) ist. Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.
- Eine Angabe, ob die Arbeitsaktion aktiviert ist.
- Eine automatisch generierte ID für die Arbeitsaktion.

Sie können eine Arbeitsaktion entweder mit dem Schlüsselwort WORK ACTION in der Anweisung CREATE WORK ACTION SET oder mit dem Schlüsselwort ADD in der Anweisung ALTER WORK ACTION SET erstellen. Mit dem Schlüsselwort ALTER in der Anweisung ALTER WORK ACTION SET können Sie eine Arbeitsaktion ändern. Sie können eine Arbeitsaktion aus einem Arbeitsaktionsset mit dem Schlüsselwort DROP in der Anweisung ALTER WORK ACTION SET oder durch Löschen des gesamten Arbeitsaktionssets entfernen.

Sie können Ihre Arbeitsaktionen durch Abfragen der Sicht SYSCAT.WORKACTIONS anzeigen.

Arbeitsaktionssets

Ein Arbeitsaktionsset hat die folgenden Attribute:

- Einen Namen für das Arbeitsaktionsset, der in der Datenbank eindeutig ist.
- Den Namen des Arbeitsklassensets, das mindestens eine Arbeitsklasse enthält, auf die die Gruppe von Aktionen angewendet wird.

Da die Definitionen der Arbeitsklassensets getrennt sind von den für sie definierten Arbeitsaktionssets, können Sie mehrere Arbeitsaktionssets für ein Arbeitsklassenset definieren.

- Den Objekttyp, dem das Arbeitsaktionsset zugeordnet ist (Datenbank, Servicesuperklasse oder Workload).
- Den Namen der Servicesuperklasse, auf die die Aktionen und das Arbeitsklassenset angewendet werden (für Arbeitsaktionssets, die einer Servicesuperklasse zugeordnet sind).
- Eine Angabe, ob das Arbeitsaktionsset aktiviert ist.
- Benutzerkommentare.
- Mindestens eine Arbeitsaktion (ein Arbeitsaktionsset muss keine Arbeitsaktionen enthalten).
- Eine automatisch generierte ID, die das Arbeitsaktionsset eindeutig bezeichnet.

Sie können ein Arbeitsaktionsset mit der Anweisung CREATE WORK ACTION SET erstellen, mit der Anweisung ALTER WORK ACTION SET ändern und mit der Anweisung DROP WORK ACTION SET löschen.

Sie können Ihre Arbeitsaktionssets durch Abfragen der Sicht SYSCAT.WORKACTIONSETS anzeigen.

Wenn Sie ein Arbeitsaktionsset erstellen, müssen Sie das Objekt angeben, auf das das Arbeitsaktionsset angewendet werden soll. Gültige Objekttypen sind die Datenbank, eine Workload oder eine Servicesuperklasse. Sie müssen auch angeben, mit welchem Arbeitsklassenset das Arbeitsaktionsset arbeiten soll. Dies erlaubt es Ihnen, die Arbeitsklassen im Arbeitsklassenset zu verwenden, um die Typen von Aktivitäten anzugeben, auf die Sie die Arbeitsaktionen anwenden wollen.

Wenn Sie eine Arbeitsaktionsset für eine Servicesuperklasse erstellen, müssen die folgenden wichtigen Aspekte beachtet werden:

1. Wenn Sie eine Workload so einrichten, dass ihre Datenbankaktivitäten direkt einer Serviceunterklasse zugeordnet werden, wird das Arbeitsaktionsset, das mit dieser Servicesuperklasse verknüpft ist, nie für die Aktivitäten verwendet, die von dieser Workload abgesetzt werden. Wenn eine Workload also Aktivitäten direkt einer Serviceunterklasse zuordnet, wird das Arbeitsaktionsset umgangen. Arbeitsaktionen im Arbeitsaktionsset werden nicht auf die Aktivitäten angewendet, die direkt der Serviceunterklasse zugeordnet sind.
2. Aktivitäten, die nicht durch eine Arbeitsaktion innerhalb des Arbeitsaktionssets für die Superklasse einer Servicesuperklasse zugeordnet sind, werden in der Unterklasse SYSDEFAULTSUBCLASS der Servicesuperklasse ausgeführt.
3. Alle Arbeiten, die von einer der Superklasse zugeordneten Workload übergeben und nicht als Aktivität erkannt werden, beispielsweise eine Anforderung PREPARE oder **RUNSTATS**, werden in der Unterklasse SYSDEFAULTSUBCLASS der Servicesuperklasse ausgeführt, da das Arbeitsaktionsset sich nicht auf sie auswirkt.

Erstellen einer Arbeitsaktion

Mit der Anweisung CREATE WORK ACTION SET oder ALTER WORK ACTION SET können Sie eine Arbeitsaktion erstellen.

Vorbereitende Schritte

Zum Erstellen einer Arbeitsaktion benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Voraussetzungen sind in den folgenden Abschnitten aufgeführt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Anhang A, „Allgemeine Namenskonventionen“, auf Seite 541

Wenn Sie eine Arbeitsaktion erstellen:

- Sie ordnen eine Arbeitsaktion einer Arbeitsklasse zu. Die Arbeitsklasse muss bereits im Arbeitsklassenset vorhanden sein, auf das das Arbeitsaktionsset angewendet wird.
- Wenn die Arbeitsaktion ein Schwellenwert ist, muss das Arbeitsaktionsset für eine Workload oder die Datenbank definiert sein. Eine Liste der unterstützten Schwellenwerte für Arbeitsaktionen finden Sie in „In Arbeitsaktionen verwendbare Schwellenwerte“ auf Seite 118.
- Wenn Sie eine Zuordnungsarbeitsaktion erstellen, muss das Arbeitsaktionsset für eine Servicesuperklasse definiert sein. Die zuzuordnende Serviceunterklasse muss bereits in der Servicesuperklasse vorhanden sein, für die dieses Arbeitsaktionsset definiert wird. Außerdem können Sie die Standardserviceunterklasse nicht angeben.
- Auf eine einzelne Arbeitsklasse können mehrere Arbeitsaktionen angewendet werden. Diese müssen jedoch verschiedene Arbeitsaktionstypen aufweisen. Sie können z. B. eine Zuordnungsarbeitsaktion und eine Erfassungsarbeitsaktion auf

dieselbe Arbeitsklasse anwenden. Auf dieselbe Arbeitsklasse können Sie jedoch jeweils nur eine Arbeitsaktion eines Typs anwenden. Sie können z. B. nicht mehrere Zuordnungsarbeitsaktionen auf dieselbe Arbeitsklasse anwenden. Die einzige Ausnahme für diese Regel ist eine Arbeitsaktion, die einen Schwellenwert darstellt. Sie können mehrere Schwellenwertarbeitsaktionen auf eine einzelne Arbeitsklasse anwenden, aber jeder Schwellenwert muss einen anderen Typ haben.

- Wenn Sie eine Arbeitsaktion für das Erfassen von zusammengefassten Aktivitätsdaten erstellen, muss das Arbeitsaktionsset für eine Servicesuperklasse oder Workload definiert sein.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsaktion zu erstellen:

1. Verwenden Sie das Schlüsselwort *arbeitsaktionsdefinition* der Anweisung CREATE WORK ACTION SET oder das Schlüsselwort ADD *arbeitsaktionsdefinition* der Anweisung ALTER WORK ACTION SET. Geben Sie mindestens eine der folgenden Optionen für die Arbeitsaktion an:

- Einen Namen für die Arbeitsaktion. Der Name der Arbeitsaktion muss im Arbeitsaktionsset eindeutig sein.
- Den Namen der Arbeitsklasse, auf die diese Arbeitsaktion angewendet wird. Die Arbeitsklasse muss eine der Arbeitsklassen im Arbeitsklassenset sein, mit dem das Arbeitsaktionsset verknüpft ist. Wenn Sie z. B. diese Arbeitsaktion auf die Arbeitsklasse LARGEDML anwenden wollen, geben Sie Folgendes an:

```
ON WORK CLASS LARGEDML
```

- Die Aktion, die die Aktivitäten anwenden soll, die der Arbeitsklasse für diese Arbeitsaktion entsprechen:

- Wenn das Arbeitsaktionsset mit einer Servicesuperklasse verknüpft ist, können Sie das Schlüsselwort MAP ACTIVITY angeben, sodass die Arbeitsaktion Aktivitäten einer Serviceunterklasse in der Servicesuperklasse zuordnet. Standardmäßig werden bei Zuordnungsarbeitsaktionen verschachtelte Aktivitäten derselben Serviceunterklasse wie das übergeordnete Element zugeordnet. Ein Cursor, der in einer Routine geöffnet wurde, ist ein Beispiel für eine verschachtelte Aktivität.

Wenn Sie z. B. wollen, dass die Arbeitsaktion der Serviceunterklasse SMALLREAD zugeordnet wird, und alle verschachtelten Aktivitäten derselben Serviceunterklasse zugeordnet werden sollen, geben Sie Folgendes an:

```
MAP ACTIVITY TO SMALLREAD
```

Sie könnten auch Folgendes angeben:

```
MAP ACTIVITY WITH NESTED TO SMALLREAD
```

Wenn die Arbeitsaktion der Serviceunterklasse zugeordnet werden soll, verschachtelte Aktivitäten jedoch nicht dieser Serviceunterklasse zugeordnet werden sollen, geben Sie Folgendes an:

```
MAP ACTIVITY WITHOUT NESTED TO SMALLREAD
```

Wenn Sie die Arbeitsaktion als WITHOUT NESTED definieren, werden verschachtelte Aktivitäten gemäß ihrem Aktivitätstyp behandelt, statt automatisch derselben Serviceunterklasse wie die übergeordnete Aktivität zugeordnet zu werden. Wenn z. B. eine CALL-Aktivität der Serviceunterklasse subsc1 zugeordnet wird und die Routine einen offenen Cursor enthält, könnte der offene Cursor einer anderen Serviceunterklasse zugeordnet werden, wenn er unter eine andere Arbeitsklasse fällt, auf die eine andere Zuordnungsarbeitsaktion angewendet wird.

- Wenn das Arbeitsaktionsset einer Datenbank oder Workload zugeordnet ist, können Sie ein Schlüsselwort WHEN angeben, um festzulegen, dass ein Schwellenwert auf die Aktivität angewendet werden soll, und um die Aktion zu definieren, die durchzuführen ist, wenn die Aktivität einen Schwellenwertverstoß verursacht. Sie können die folgenden Schwellenwerte für eine Arbeitsaktion angeben:
 - ACTIVITYTOTALTIME
 - ESTIMATEDSQLCOST
 - CONCURRENTDBCOORDACTIVITIES
 - CPUTIME
 - SQLROWSREAD
 - SQLTEMPSPACE
 - SQLROWSRETURNED

Bei einem Schwellenwertverstoß können Sie die folgenden zu ergreifenden Maßnahmen angeben:

- Ob Aktivitätsdaten zu der Aktivität erfasst werden sollen, die den Schwellenwertverstoß verursachte. Wenn die Aktivitätsdaten beim Abschluss der Aktivitätsausführung erfasst werden, werden sie in einen aktiven Ereignismonitor für Aktivitäten geschrieben. Standardmäßig werden Daten zu Aktivitäten nicht erfasst. Wenn Sie Daten zu dieser Aktivität erfassen wollen, können Sie diese vom Koordinatormember, einem bestimmten Member oder von allen Datenbankmitgliedern erfassen. Sie haben die Möglichkeit, diese Daten mit oder ohne Details zur Anweisung und ihrer Kompilierungsumgebung zu erfassen. Wenn Sie Details zur Anweisung und Kompilierungsumgebung erfassen wollen, können Sie auch angeben, dass die in der Aktivität verwendeten Eingabedatenwerte erfasst werden sollen.
- Ob die Aktivität, die den Schwellenwertverstoß verursachte, weiter ausgeführt werden darf. Standardmäßig wird die Aktivität gestoppt.

Wenn die Arbeitsaktion z. B. prüfen soll, ob DML-Anweisungen mit einem Aufwand von über 2.000 Timern vorhanden sind, bei einem Schwellenwertverstoß die Basisdaten zu dieser Aktivität erfassen soll und die Aktion dann weiter ausgeführt werden soll, geben Sie Folgendes an:

```
WHEN ESTIMATEDSQLCOST > 2000 COLLECT ACTIVITY DATA CONTINUE
```

- Mit dem Schlüsselwort PREVENT EXECUTION können Sie verhindern, dass Aktivitäten ausgeführt werden, die der für diese Arbeitsaktion definierten Arbeitsklasse entsprechen.
- Wenn Sie die Anzahl der Datenbankaktivitäten, die der Arbeitsklasse zugeordnet sind, ohne den zusätzlichen Aufwand einer weiteren Aktion (wie z. B. dem Erfassen von Daten oder dem Zuordnen einer Aktivität) zählen wollen, können Sie das Schlüsselwort COUNT ACTIVITY angeben.
- Geben Sie das Schlüsselwort COLLECT ACTIVITY DATA an, um Aktivitätsdaten für Aktivitäten zu erfassen, die unter die Arbeitsklasse fallen. Wenn die Aktivitätsdaten beim Abschluss der Aktivitätsausführung erfasst werden, werden sie in einen aktiven Ereignismonitor für Aktivitäten geschrieben. Standardmäßig werden Daten zu Aktivitäten nicht erfasst. Wenn Sie Daten zu dieser Aktivität erfassen wollen, können Sie diese vom Koordinatormember oder von allen Mitgliedern erfassen. Wenn Sie Aktivitätsdetails wie die Anweisung und die Kompilierungsumgebung erfassen wollen, können Sie dies mit dem Schlüsselwort WITH DETAILS tun. Sie können auch das Schlüsselwort AND VALUES verwenden, um Eingabedatenwerte (für Aktivitäten mit Eingabedatenwerten) an den Aktivitätsereignismonitor zu senden.

Nehmen wir beispielsweise an, Sie haben ein Arbeitsaktionsset, das auf eine Servicesuperklasse angewendet wird. Sie wollen, dass Aktivitätsdaten für alle Aktivitäten, die dieser Arbeitsaktion zugeordnet sind, in den entsprechenden Ereignismonitor geschrieben werden, einschließlich aller zusammengefasster Aktivitätsinformationen, aller Informationen zur Kompilierungsumgebung sowie aller Eingabedatenwerte. Geben Sie in diesem Fall Folgendes an:

```
COLLECT ACTIVITY DATA ON ALL WITH DETAILS AND VALUES
```

- Wenn das Arbeitsaktionsset einer Servicesuperklasse oder Workload zugeordnet ist, können Sie das Schlüsselwort `COLLECT AGGREGATE ACTIVITY DATA` angeben, um zusammengefasste Aktivitätsdaten für Aktivitäten zu erfassen, die unter die Arbeitsklasse fallen. In diesem Fall werden zusammengefasste Aktivitätsdaten erfasst und an den entsprechenden Ereignismonitor gesendet. Diese Informationen werden regelmäßig in einem vom Datenbankkonfigurationsparameter `wlm_collect_int` festgelegten Intervall erfasst.

Nehmen wir beispielsweise an, Sie haben ein Arbeitsaktionsset, das auf eine Servicesuperklasse angewendet wird. Sie wollen, dass zusammengefasste Aktivitätsdaten für alle Aktivitäten, die dieser Arbeitsaktion zugeordnet sind, in den entsprechenden Ereignismonitor geschrieben werden, einschließlich aller Basisdaten, dem Histogramm des per DLL geschätzten Aufwands für Aktivitäten sowie dem Histogramm der per DML ermittelten Zwischenankunftszeit (Inter-Arrival Time) von Aktivitäten. Geben Sie in diesem Fall Folgendes an:

```
COLLECT AGGREGATE ACTIVITY DATA EXTENDED
```

- Die von einer Arbeitsaktion `COLLECT AGGREGATE ACTIVITY DATA` verwendeten Histogrammschablonen zum Beschreiben der für die entsprechende Arbeitsklasse erstellten Histogramme. Durch die Angabe der von einer Arbeitsaktion verwendeten Histogrammschablonen werden die entsprechenden Zeilen in der Sicht `SYSCAT.HISTOGRAMTEMPLATEUSE` hinzugefügt, die die Histogrammschablonen anzeigt, auf die die Serviceklasse oder Arbeitsaktion verweist. Wenn Sie z. B. Zwischenankunftsstatistikdaten für die Standardhistogrammschablone für die Zwischenankunftszeit erfassen wollen, geben Sie Folgendes an:

```
INTERARRIVALTIME HISTOGRAM TEMPLATE SYSDEFAULTHISTOGRAM
```

Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

- Angabe, ob die Arbeitsaktion aktiviert oder inaktiviert ist. Standardmäßig wird eine Arbeitsaktion als aktiviert erstellt. Sie können jedoch mit dem Schlüsselwort `ENABLE` oder `DISABLE` angeben, ob sie aktiviert oder inaktiviert sein soll. Wenn die Arbeitsaktion inaktiviert ist, berücksichtigt der Datenserver diese Arbeitsaktion nicht, wenn Aktivitäten in die Datenbank oder die Servicesuperklasse eingehen (je nach Objekt, für das Sie das Arbeitsaktionsset erstellt haben).
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsaktion der Sicht `SYSCAT.WORKACTIONS` hinzugefügt. Wenn es sich bei der Arbeitsaktion um einen Schwellenwert handelt, wird der Schwellenwert der Sicht `SYSCAT.THRESHOLDS` hinzugefügt.

Eine neue Arbeitsaktion wird in der Datenbank erst nach dem Durchführen des Commits wirksam und beeinflusst die zurzeit ausgeführten Datenbankaktivitäten nicht.

Ändern einer Arbeitsaktion

Wenn Sie eine Arbeitsaktion ändern müssen, verwenden Sie die Anweisung `ALTER WORK ACTION SET`.

Vorbereitende Schritte

Zum Ändern einer Arbeitsaktion benötigen Sie die Berechtigung SQLADM, WLMADM oder DBADM. Soll eine andere Klausel als die Klausel COLLECT angegeben werden, muss die Berechtigungs-ID die Berechtigung WLMADM oder DBADM aufweisen.

Weitere Voraussetzungen sind in „DDL-Anweisungen für Workload-Management“ auf Seite 18 aufgeführt.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsaktion zu ändern:

1. Mit dem Schlüsselwort ALTER der Anweisung ALTER WORK ACTION SET können Sie eines oder mehrere der folgenden Merkmale der Arbeitsaktion ändern.
 - Sie können die Arbeitsklasse ändern, auf die diese Arbeitsaktion angewendet wird. Die Arbeitsklasse muss bereits im Arbeitsklassenset vorhanden sein, auf das das Arbeitsaktionsset angewendet wird.
 - Wenn die Arbeitsaktion einer Serviceunterklasse zugeordnet ist, können Sie ändern, welcher Serviceunterklasse die Datenbankaktivität zugeordnet werden soll. Sie können die Zuordnung nur zu einer Serviceunterklasse in derselben Servicesuperklasse ändern. Die Zuordnung zur Standardserviceunterklasse ist nicht möglich. Sie können auch ändern, ob verschachtelte Aktivitäten in der Aktivität derselben Serviceunterklasse zugeordnet werden sollen. Wenn z. B. die Arbeitsaktion zurzeit als WITH NESTED definiert ist, können Sie dies in WITHOUT NESTED ändern. Diese Änderung würde bewirken, dass die verschachtelten Aktivitäten gemäß ihrem Aktivitätstyp behandelt werden, statt automatisch derselben Serviceunterklasse wie die übergeordnete Aktivität zugeordnet zu werden. Wenn z. B. eine CALL-Aktivität der Serviceunterklasse SUBSC1 zugeordnet wird und die Routine einen offenen Cursor enthält, könnte der offene Cursor einer anderen Serviceunterklasse zugeordnet werden, wenn er unter eine andere Arbeitsklasse fällt, auf die eine andere Zuordnungsarbeitsaktion angewendet wird.
 - Sie können den Aktionstyp für die Arbeitsaktion (d. h. Zuordnung, Schwellenwert, Verhindern der Ausführung, Aktivitätszählung, Erfassung) ändern. Sie müssen ihn jedoch in einen gültigen Arbeitstyp ändern. Wenn z. B. die Arbeitsaktion die Aktivität einer Serviceunterklasse zuordnen soll, können Sie die Arbeitsaktion nicht in einen Schwellenwert ändern oder umgekehrt. Der Grund hierfür ist, dass in diesem Beispiel das Arbeitsaktionsset auf eine Servicesuperklasse angewendet werden musste, um eine Zuordnungsaktion zu haben, und Schwellenwertaktionen für Arbeitsaktionssets, die Servicesuperklassen zugeordnet sind, nicht gültig sind. Wenn Sie den Typ einer Arbeitsaktion ändern, die eine Schwellenwertarbeitsaktion ist, oder wenn Sie den Typ einer Arbeitsaktion in einen Schwellenwert ändern, geschieht Folgendes:
 - Wenn die Arbeitsaktion ein Schwellenwert war und in eine andere Aktion geändert wurde, wird der Schwellenwert aus der Sicht SYSCAT.THRESHOLDS entfernt.
 - Wenn die Arbeitsaktion kein Schwellenwert war und in einen Schwellenwert geändert wurde, wird ein neuer Schwellenwert in der Sicht SYSCAT.THRESHOLDS erstellt.

Anmerkung: Wenn die Aktion ein Schwellenwert ist, können Sie den Schwellenwerttyp nicht in einen anderen Schwellenwert ändern. Wenn also die Arbeitsaktion ein Schwellenwert SQLROWSRETURNED war, können Sie ihn nicht in einen Schwellenwert SQLTEMPSPACE ändern. Außerdem können Sie den Arbeitsaktionstyp eines aktivierten Arbeitsaktionsschwellenwerts CONCURRENTDBCOORDACTIVITIES nicht ändern.

- Sie können die von einer Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA verwendeten Histogrammschablonen zum Beschreiben der für die entsprechende Arbeitsklasse erstellten Histogramme ändern. Durch das Aktualisieren der von einer Arbeitsaktion verwendeten Histogrammschablonen werden die entsprechenden Zeilen in der Sicht SYSCAT.HISTOGRAMTEMPLATEUSE aktualisiert, die die Histogrammschablonen anzeigt, auf die die Serviceklasse oder Arbeitsaktion verweist. Weitere Informationen zu Histogrammen und Histogrammschablonen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.
 - Sie können angeben, ob Sie die Arbeitsaktion aktivieren oder inaktivieren wollen. Standardmäßig sind Arbeitsaktionen aktiviert. Wenn die Arbeitsaktion aktiviert ist, berücksichtigt sie der Datenserver für die Anwendung auf die Aktivität, die unter die Arbeitsklasse für die Arbeitsaktion fällt. Wenn die Arbeitsaktion inaktiviert ist, ignoriert der Datenserver sie.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsaktion in der Sicht SYSCAT.WORKACTIONS aktualisiert.

Inaktivieren einer Arbeitsaktion

Sie können eine Arbeitsaktion inaktivieren, die nicht mehr auf eine Arbeitsklasse angewendet werden soll. Während der Laufzeit wird die inaktivierte Arbeitsaktion so behandelt, als wäre sie nicht vorhanden.

Vorbereitende Schritte

Zum Inaktivieren einer Arbeitsaktion benötigen Sie die Berechtigung WLMADM oder DBADM.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsaktion zu inaktivieren:

1. Verwenden Sie abhängig davon, ob Sie ein Arbeitsaktionsset erstellen oder ändern, eine der folgenden Anweisungen:
 - Verwenden Sie das Schlüsselwort DISABLE und das Schlüsselwort ADD der Anweisung CREATE WORK ACTION SET. Beispiel:
`ADD WORK ACTION arbeitsaktionsname ON WORK CLASS arbeitsklassenname ... DISABLE`
 - Verwenden Sie das Schlüsselwort DISABLE und das Schlüsselwort ALTER der Anweisung ALTER WORK ACTION SET. Beispiel:
`ALTER WORK ACTION arbeitsaktionsname ... DISABLE`
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsaktion in der Sicht SYSCAT.WORKACTIONS aktualisiert.

Löschen einer Arbeitsaktion

Wenn Sie eine Arbeitsaktion nicht mehr benötigen, können Sie sie aus dem Arbeitsaktionsset löschen.

Vorbereitende Schritte

- Zum Löschen einer Arbeitsaktion benötigen Sie die Berechtigung WLMADM oder DBADM.
- Weitere Voraussetzungen sind in „DDL-Anweisungen für Workload-Management“ auf Seite 18 aufgeführt.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Arbeitsaktion zu löschen:

1. Verwenden Sie das Schlüsselwort DROP der Anweisung ALTER WORK ACTION SET. Wenn Sie eine Arbeitsaktion mit dem Schwellenwert CONCURRENTDBCOORDACTIVITIES löschen wollen, müssen Sie die Arbeitsaktion in einer Operation ALTER WORK ACTION SET inaktivieren, ein Commit für diese Änderung durchführen, sich vergewissern, dass es keine Aktivitäten in der Warteschlange gibt, und dann den Schwellenwert in einer zweiten Operation ALTER WORK ACTION SET löschen.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Arbeitsaktion aus der Sicht SYSCAT.WORKACTIONS entfernt. Wenn es sich bei der Arbeitsaktion um eine Schwellenwertarbeitsaktion handelt, wird auch der Schwellenwert aus der Sicht SYSCAT.THRESHOLDS entfernt.

Ein geändertes Arbeitsaktionsset und eine geänderte Arbeitsaktion werden in der Datenbank erst nach dem Durchführen eines Commits für die Änderung wirksam und beeinflussen die zurzeit ausgeführten Datenbankaktivitäten nicht.

Erstellen eines Arbeitsaktionssets

Mit der Anweisung CREATE WORK ACTION SET können Sie eine Arbeitsaktion und ein Arbeitsaktionsset erstellen.

Vorbereitende Schritte

Zum Erstellen eines Arbeitsaktionssets benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Voraussetzungen sind in den folgenden Abschnitten aufgeführt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18

Wenn Sie ein Arbeitsaktionsset erstellen:

- Sie ordnen es einem Arbeitsklassenset zu. Das Arbeitsklassenset muss bereits vorhanden sein.
- Sie ordnen es auch einer Datenbank, Workload oder Servicesuperklasse zu. Wenn Sie das Arbeitsaktionsset einer Servicesuperklasse zuordnen, muss die Serviceklasse bereits vorhanden sein. Das Arbeitsaktionsset kann nicht für die Standardsystemserviceklasse (SYSDEFAULTSYSTEMCLASS), die Standardbenutzerserviceklasse (SYSDEFAULTUSERCLASS) oder die Standardwartungsserviceklasse (SYSDEFAULTMAINTENANCECLASS) definiert werden. Wenn Sie das Arbeitsaktionsset einer Workload zuordnen, muss die Workload auf dem aktuellen Server vorhanden sein. Sie können das Arbeitsaktionsset nicht der Standardverwaltungsworkload (SYSDEFAULTADMWORKLOAD) zuordnen.

Vorgehensweise

Gehen Sie wie folgt vor, um ein Arbeitsaktionsset zu erstellen:

1. Verwenden Sie die Anweisung CREATE WORK ACTION SET mit den folgenden Optionen:
 - Geben Sie einen Namen für das Arbeitsaktionsset an. Der Name des Arbeitsaktionssets muss in der Datenbank eindeutig sein.
 - Geben Sie das Objekt an, dem das Arbeitsaktionsset zugeordnet werden soll. Sie können eine Datenbank, Workload oder Servicesuperklasse angeben. Wenn Sie angeben, dass das Arbeitsaktionsset einer Datenbank zugeordnet werden soll, kann keine der Arbeitsaktionen im Arbeitsaktionsset eine Zuordnungsarbeitsaktion oder eine Arbeitsaktion für das Erfassen von zusammengefassten Aktivitätsdaten sein. Wenn Sie angeben, dass das Arbeitsaktionsset mit einer Servicesuperklasse verknüpft werden soll, darf keine der Arbeitsaktionen im Arbeitsaktionsset ein Schwellenwert sein. Wenn Sie angeben, dass das Arbeitsaktionsset einer Workload zugeordnet werden soll, kann keine der Arbeitsaktionen im Arbeitsaktionsset eine Zuordnungsarbeitsaktion sein. Wenn Sie z. B. das Arbeitsaktionsset auf die Servicesuperklasse REPORTS anwenden wollen, geben Sie Folgendes an:


```
FOR SERVICE CLASS REPORTS
```

 Wenn Sie das Arbeitsaktionsset auf die Datenbank anwenden wollen, geben Sie Folgendes an:


```
FOR DATABASE
```

 Wenn Sie das Arbeitsaktionsset auf eine Workload namens WL1 anwenden wollen, geben Sie Folgendes an:


```
FOR WORKLOAD WL1
```
 - Geben Sie das Arbeitsklassenset an, mit dem das Arbeitsaktionsset verknüpft werden soll. Die Arbeitsklassen im Arbeitsklassenset klassifizieren die Datenbankaktivitäten, auf die die Arbeitsaktionen im Arbeitsaktionsset angewendet werden. Wenn Sie z. B. das Arbeitsaktionsset mit dem Arbeitsklassenset LARGEREADS verknüpfen wollen, geben Sie Folgendes an:


```
USING WORK CLASS SET LARGEREADS
```
 - Optional: Erstellen Sie eine oder mehrere Arbeitsaktionen für das Arbeitsaktionsset. Anweisungen hierzu finden Sie in „Erstellen einer Arbeitsaktion“ auf Seite 104.
 - Optional: Geben Sie an, ob das Arbeitsaktionsset aktiviert oder inaktiviert ist. Standardmäßig ist das Arbeitsaktionsset aktiviert. Wenn das Arbeitsaktionsset inaktiviert ist, berücksichtigt der Datenserver dieses Arbeitsaktionsset (und alle darin enthaltenen Arbeitsaktionen) nicht bei der Ausführung von Aktivitäten.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird das Arbeitsaktionsset zur Sicht SYSCAT.WORKACTIONSETS hinzugefügt.

Ein neues Arbeitsaktionsset wird in der Datenbank erst nach dem Durchführen des Commits wirksam und beeinflusst die zurzeit ausgeführten Datenbankaktivitäten nicht.

Ändern eines Arbeitsaktionssets

Mit der Anweisung ALTER WORK ACTION SET können Sie eine Arbeitsaktion in einem Arbeitsaktionsset hinzufügen, ändern oder löschen oder das Arbeitsaktionsset aktivieren oder inaktivieren.

Vorbereitende Schritte

Zum Ändern eines Arbeitsaktionssets benötigen Sie die Berechtigung SQLADM, WLMADM oder DBADM. Soll eine andere Klausel als die Klausel COLLECT angegeben werden, muss die Berechtigungs-ID die Berechtigung WLMADM oder DBADM aufweisen.

Weitere Voraussetzungen sind in den folgenden Abschnitten aufgeführt:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Anhang A, „Allgemeine Namenskonventionen“, auf Seite 541

Wenn Sie ein Arbeitsaktionsset erstellen, das mit einem bestimmten Arbeitsklassenset zusammenarbeiten soll, können Sie es nicht so ändern, dass es mit einem anderen Arbeitsklassenset zusammenarbeitet, da die Arbeitsaktionen im Arbeitsaktionsset von den Arbeitsklassen im Arbeitsklassenset abhängig sind. Wenn Sie das Arbeitsklassenset ändern wollen, auf das dieses Arbeitsaktionsset angewendet werden soll, müssen Sie das Arbeitsaktionsset löschen und neu erstellen.

Sie können nicht ändern, auf welches Objekt das Arbeitsaktionsset angewendet wird, da der Typ der Arbeitsaktionen im Arbeitsaktionsset davon abhängt, für welches Objekt (Datenbank, Workload oder Servicesuperklasse) das Arbeitsaktionsset definiert ist. Wenn Sie das Objekt ändern wollen, dem das Arbeitsaktionsset zugeordnet ist, müssen Sie das Arbeitsaktionsset löschen und neu erstellen.

Vorgehensweise

Gehen Sie wie folgt vor, um ein Arbeitsaktionsset zu ändern:

1. Wenn Sie eine neue Arbeitsaktion zum Arbeitsaktionsset hinzufügen wollen, verwenden Sie das Schlüsselwort ADD. Informationen zu den Parametern, die Sie beim Hinzufügen einer Arbeitsaktion zu einem Arbeitsaktionsset angeben können, finden Sie in „Erstellen einer Arbeitsaktion“ auf Seite 104.
2. Wenn Sie eine vorhandene Arbeitsaktion ändern wollen, verwenden Sie das Schlüsselwort ALTER. Informationen zum Ändern einer Arbeitsaktion finden Sie in „Ändern einer Arbeitsaktion“ auf Seite 107.
3. Wenn Sie eine Arbeitsaktion löschen wollen, verwenden Sie das Schlüsselwort DROP. Informationen zum Löschen einer Arbeitsaktion aus einem Arbeitsaktionsset finden Sie in „Löschen einer Arbeitsaktion“ auf Seite 109.
4. Sie können ein zurzeit inaktiviertes Arbeitsaktionsset aktivieren und umgekehrt. Wenn Sie ein aktiviertes Arbeitsaktionsset inaktivieren, ignoriert der Datenserver dieses Set nach dem Durchführen eines Commits für Ihre Änderungen. Weitere Informationen finden Sie in „Inaktivieren eines Arbeitsaktionssets“ auf Seite 113. Wenn Sie das Arbeitsaktionsset aktivieren, wird das Arbeitsaktionsset nach dem Durchführen eines Commits für Ihre Änderungen auf die nächste zutreffende Aktivität angewendet, die in die Datenbank eingeht.

Anmerkung: Durch die Inaktivierung eines Arbeitsaktionssets werden die Arbeitsaktionen innerhalb des Arbeitsaktionssets nicht inaktiviert, die Arbeitsaktion wirkt sich allerdings nicht mehr auf irgendwelche Arbeiten aus. Wenn Sie ein Arbeitsaktionsset löschen wollen, das einen Schwellenwert für gleichzeitig ablaufende Arbeitsaktionen enthält, müssen Sie die gleichzeitig ablaufende Arbeitsaktion vor dem Löschen des Arbeitsaktionssets inaktivieren, da Schwellenwerte für den gemeinsamen Zugriff inaktiviert werden müssen, bevor sie gelöscht werden können.

5. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht SYSCAT.WORKACTIONSETS aktualisiert. Alle hinzugefügten, geänderten oder gelöschten Arbeitsaktionen werden in der Sicht SYSCAT.WORKACTIONS aktualisiert.

Inaktivieren eines Arbeitsaktionssets

Mit dem Schlüsselwort `DISABLE` der Anweisung `CREATE WORK ACTION SET` oder der Anweisung `ALTER WORK ACTION SET` können Sie ein Arbeitsaktionsset inaktivieren.

Vorbereitende Schritte

Zum Inaktivieren eines Arbeitsaktionssets benötigen Sie die Berechtigung `WLMADM` oder `DBADM`.

Informationen zu diesem Vorgang

Durch die Inaktivierung eines Arbeitsaktionssets werden die Arbeitsaktionen innerhalb des Arbeitsaktionssets nicht inaktiviert, die Arbeitsaktion wirkt sich allerdings nicht mehr auf irgendwelche Arbeiten aus. Während der Laufzeit wird ein inaktiviertes Arbeitsaktionsset so behandelt, als wäre es nicht vorhanden.

Anmerkung: Wenn Sie ein Arbeitsaktionsset löschen wollen, das einen Schwellenwert für gleichzeitig ablaufende Arbeitsaktionen enthält, müssen Sie die gleichzeitig ablaufende Arbeitsaktion vor dem Löschen des Arbeitsaktionssets inaktivieren, da Schwellenwerte für den gemeinsamen Zugriff inaktiviert werden müssen, bevor sie gelöscht werden können.

Nehmen wir beispielsweise an, dass Sie ein Arbeitsaktionsset namens `READACTIVITIES` haben, das mit einem Arbeitsklassenset namens `READCLASSES` verknüpft ist, und dass dieses Arbeitsaktionsset für eine Servicesuperklasse namens `READSERVICECLASS` definiert ist. Das Arbeitsaktionsset `SMALLREAD` enthält eine Arbeitsaktion, die alle `SELECT`-Anweisungen der Serviceunterklasse `SMALLREADSERVICECLASS` neu zuordnet. Wenn das Arbeitsaktionsset `READACTIVITIES` inaktiviert ist, werden alle `SELECT`-Anweisungen so behandelt, als wäre das Arbeitsaktionsset `READACTIVITIES` nicht vorhanden. Sie werden daher der Standardserviceunterklasse zugeordnet.

Vorgehensweise

Gehen Sie wie folgt vor, um ein Arbeitsaktionsset zu inaktivieren:

1. Verwenden Sie abhängig davon, ob Sie ein Arbeitsaktionsset erstellen oder ändern, eine der folgenden Anweisungen:
 - Gehen Sie wie folgt vor, um ein Arbeitsaktionsset zu erstellen, das inaktiviert ist:

```
CREATE WORK ACTION SET name_des_arbeitsaktionssets ... DISABLE
```
 - Gehen Sie wie folgt vor, um ein bereits vorhandenes Arbeitsaktionsset zu inaktivieren:

```
ALTER WORK ACTION SET name_des_arbeitsaktionssets ... DISABLE
```
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird die Workload in der Sicht SYSCAT.WORKACTIONSETS aktualisiert.

Löschen eines Arbeitsaktionssets

Mit der Anweisung `DROP WORK ACTION SET` können Sie ein Arbeitsaktionsset löschen.

Vorbereitende Schritte

Zum Löschen eines Arbeitsaktionssets benötigen Sie die Berechtigung WLMADM oder DBADM.

Informationen zu diesem Vorgang

Beim Löschen eines Arbeitsaktionssets wird auch das Arbeitsaktionsset mit allen darin enthaltenen Arbeitsaktionen gelöscht.

Wenn das Arbeitsaktionsset eine Schwellenwertarbeitsaktion CONCURRENTDBCOORDACTIVITIES enthält, muss diese Arbeitsaktion zuerst inaktiviert werden, bevor das Arbeitsaktionsset gelöscht werden kann.

Vorgehensweise

Gehen Sie wie folgt vor, um ein Arbeitsaktionsset zu löschen:

1. Verwenden Sie die Anweisung DROP WORK ACTION SET.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird das Arbeitsaktionsset aus der Sicht SYSCAT.WORKACTIONSETS entfernt. Darüber hinaus werden alle Arbeitsaktionen, die Teil des Arbeitsaktionssets waren, aus der Sicht SYSCAT.WORKACTIONS entfernt. Wenn das Arbeitsaktionsset Schwellenwertarbeitsaktionen enthält, werden die Schwellenwerte aus der Sicht SYSCAT.THRESHOLDS entfernt.

Arbeitsaktionen und die Domäne des Arbeitsaktionssets

Sie können ein Arbeitsaktionsset für eine Datenbank, Servicesuperklasse oder Workload definieren. Der Typ von Arbeitsaktionen, der für ein Arbeitsaktionsset definiert werden kann, hängt vom Typ des Objekts ab, für das das Arbeitsaktionsset definiert ist.

Wenn das Arbeitsaktionsset für eine Datenbank definiert ist, müssen die Arbeitsaktionen im Arbeitsaktionsset einen der folgenden Typen haben:

- Schwellenwert
Die folgenden Schwellenwerte gelten für jede einzelne Aktivität in der entsprechenden Arbeitsklasse:
 - ACTIVITYTOTALTIME
 - CPUTIME
 - ESTIMATEDSQLCOST
 - SQLROWSREAD
 - SQLROWSRETURNED
 - SQLTEMPSPACE

Der folgende Schwellenwert gilt für alle Aktivitäten in der entsprechenden Arbeitsklasse als Gruppe:

- CONCURRENTDBCOORDACTIVITIES

Der tatsächliche Schwellenwert wird mit der Klausel WHEN *schwellenwerttyp* angegeben. Mehrere Schwellenwertarbeitsaktionen können auf eine einzelne Arbeitsklasse angewendet werden, wenn alle Schwellenwerte unterschiedliche Typen haben. Wenn diese Aktion angegeben wird, wird der Schwellenwert auf alle Datenbankaktivitäten angewendet, die mit der Arbeitsklasse verknüpft sind.

- PREVENT EXECUTION

Wenn diese Aktion angegeben wird, wird die Ausführung aller Datenbankaktivitäten verhindert, die mit der zugehörigen Arbeitsklasse übereinstimmen.

- COLLECT ACTIVITY DATA

Wenn diese Aktion angegeben wird, werden Informationen zu den Datenbankaktivitäten entsprechend der Arbeitsklasse, für die diese Arbeitsaktion definiert ist, in den aktiven ACTIVITIES-Ereignismonitor geschrieben, wenn die Ausführung der Aktivitäten abgeschlossen ist. Weitere Informationen finden Sie in „Erfassen von Daten für einzelne Aktivitäten“.

- COUNT ACTIVITY

Wenn diese Aktivität angegeben wird, erhöhen alle Datenbankaktivitäten, die der zugehörigen Arbeitsklasse zugeordnet sind, den Turnstile-Zähler (Drehkreuzzähler) für diesen Arbeitsklassentyp. (Der Turnstile-Zähler für die Arbeitsklasse wird jedesmal um 1 erhöht, wenn eine Aktivität mit dieser Arbeitsklasse verknüpft wird.) Die Arbeitsaktion COUNT ACTIVITY ist eine Möglichkeit, mit der effektiv sichergestellt werden kann, dass dieser Zähler aktualisiert wird. Wenn keine Arbeitsaktion auf eine Aktivität zu einer Arbeitsklasse angewendet wird, wird der Arbeitsklassenaktivitätenzähler nicht erhöht. Manchmal geht es nur darum, die Anzahl der Aktivitäten eines bestimmten Typs zu ermitteln. Weitere Informationen finden Sie in „Erfassen von Daten für einzelne Aktivitäten“.

Wenn die Arbeitsaktionen in dem für eine Datenbank definierten Arbeitsaktionsset mit keiner dieser Aktionen übereinstimmen, wird SQL4720N zurückgegeben.

Wenn Sie ein Arbeitsaktionsset für eine Servicesuperklasse definieren, muss es sich bei den Arbeitsaktionen im Arbeitsaktionsset um eine der folgenden Aktionen handeln:

- Eine Zuordnungsaktion

Sie können eine Aktivität einer beliebigen Serviceunterklasse in der Servicesuperklasse mit Ausnahme der Standardserviceunterklasse zuordnen. Sie geben die Serviceunterklasse, der die Aktivität zugeordnet werden soll, mit dem Schlüsselwort MAP ACTIVITY TO SERVICE CLASS an. Nur eine Zuordnungsarbeitsaktion im Arbeitsaktionsset kann auf dieselbe Arbeitsklasse angewendet werden.

- PREVENT EXECUTION

Das Verhalten ist mit dem für die Datenbankarbeitsaktion identisch.

- COLLECT ACTIVITY DATA

Das Verhalten ist mit dem für die Datenbankarbeitsaktion identisch.

- COLLECT AGGREGATE ACTIVITY DATA

Wenn diese Aktion angegeben wird, werden zusammengefasste Datenbankaktivitätsdaten entsprechend der Arbeitsklasse erfasst, für die diese Arbeitsaktion definiert ist.

- COUNT ACTIVITY

Das Verhalten ist mit dem für die Datenbankarbeitsaktion identisch.

Wenn die Arbeitsaktionen in dem für eine Servicesuperklasse definierten Arbeitsaktionsset mit keiner dieser Aktionen übereinstimmen, wird SQL4720N zurückgegeben.

Wenn das Arbeitsaktionsset für eine Workload definiert ist, müssen die Arbeitsaktionen im Arbeitsaktionsset einen der folgenden Typen haben:

- Schwellenwert

Die folgenden Schwellenwerte gelten für jede einzelne Aktivität in der entsprechenden Arbeitsklasse:

- ACTIVITYTOTALTIME

- CPUTIME
- ESTIMATEDSQLCOST
- SQLROWSREAD
- SQLROWSRETURNED
- SQLTEMPSPACE

Der folgende Schwellenwert gilt für alle Aktivitäten in der entsprechenden Arbeitsklasse als Gruppe:

- CONCURRENTDBCOORDACTIVITIES

Der tatsächliche Schwellenwert wird mit der Klausel *WHEN schwellenwerttyp* angegeben. Mehrere Schwellenwertarbeitsaktionen können auf eine einzelne Arbeitsklasse angewendet werden, wenn alle Schwellenwerte unterschiedliche Typen haben. Wenn diese Aktion angegeben wird, wird der Schwellenwert auf alle Datenbankaktivitäten angewendet, die mit der Arbeitsklasse verknüpft sind.

- PREVENT EXECUTION

Das Verhalten ist mit dem für die Datenbankarbeitsaktion identisch.

- COLLECT ACTIVITY DATA

Das Verhalten ist mit dem für die Datenbankarbeitsaktion identisch.

- COLLECT AGGREGATE ACTIVITY DATA

Das Verhalten ist mit dem für die Arbeitsaktion für Servicesuperklassen identisch.

- COUNT ACTIVITY

Das Verhalten ist mit dem für die Datenbankarbeitsaktion identisch.

Wenn die Arbeitsaktionen in dem für eine Workload definierten Arbeitsaktionsset mit keiner dieser Aktionen übereinstimmen, wird SQL4720N zurückgegeben.

Beispiel: Anwenden von Arbeitsklassenaktivitäten

Die folgende Abbildung zeigt ein Beispiel, wie die Arbeitsklassen in einem Arbeitsklassenset mit dem Namen LARGE ACTIVITIES sowohl auf die Datenbank als auch auf eine Servicesuperklasse angewendet werden. Zur Erreichung dieses Ziels werden die beiden Arbeitsaktionssets "Database large activities" und "Service class large activities" erstellt.

Sie können die Klassen im Arbeitsklassenset LARGE ACTIVITIES auch auf eine Workload anwenden, was im Beispiel nicht gezeigt wird. Erstellen Sie hierzu ein dem Workload zugeordnetes Arbeitsaktionsset und ordnen Sie dann das Arbeitsaktionsset dem Arbeitsklassenset LARGE ACTIVITIES zu.

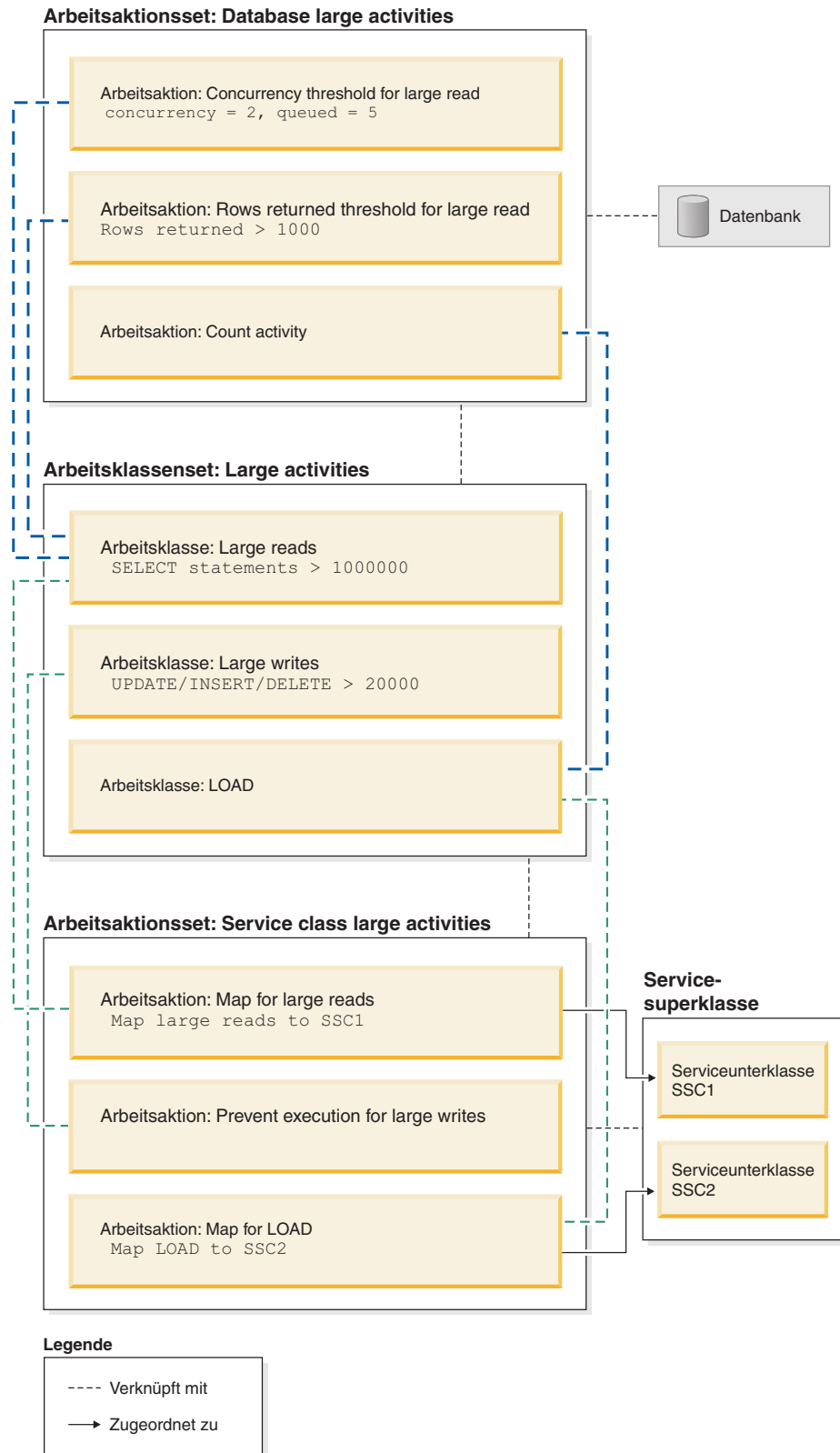


Abbildung 17. Beispiel von Arbeitsaktionen, Arbeitsaktionssets, Arbeitsklassen und Arbeitsklassensets

Die Arbeitsaktionssets sind wie folgt aufgebaut:

- Database large activities enthält:
 - Concurrency threshold for large reads. Diese Aktion erlaubt, dass zwei große Lesevorgänge gleichzeitig ausgeführt und fünf große Lesevorgänge in die Warteschlange gestellt werden können.
 - Rows returned threshold for large reads. Diese Aktion verhindert, dass große Lesevorgänge mehr als 1000 Zeilen zurückgeben.
 - Count activity for load. Diese Aktion zählt, wie oft das Dienstprogramm LOAD in der Datenbank ausgeführt wird.
- Service class large activities enthält:
 - Map for large reads. Diese Aktion ordnet große Lesevorgänge der Serviceunterklasse 1 zu.
 - Map for large writes. Diese Aktion verhindert, dass große Schreibvorgänge ausgeführt werden.
 - Map for LOAD. Diese Aktion ordnet Ladevorgänge der Serviceunterklasse 2 zu.

Ein Arbeitsaktionsset muss nicht für jede Arbeitsklasse im Arbeitsklassenset, auf das das Arbeitsaktionsset angewendet wird, eine Aktion enthalten. Außerdem können auf eine Arbeitsklasse mehrere Arbeitsaktionen angewendet werden, sofern die Aktionstypen unterschiedlich sind. Auf eine Arbeitsklasse können mehrere Schwellenwertarbeitsaktionen angewendet werden, sofern die Schwellenwerttypen unterschiedlich sind.

In Arbeitsaktionen verwendbare Schwellenwerte

Arbeitsaktionssets, die Sie für Datenbanken definieren, können Arbeitsaktionen enthalten, die Schwellenwerte angeben.

Die folgenden Schwellenwerte werden unterstützt:

- Zusammengefasster Schwellenwert:
 - CONCURRENTDBCOORDACTIVITIES
 - CONCURRENTWORKLOADACTIVITIES
- Aktivitätsschwellenwerte:
 - SQLTEMPSPACE
 - SQLROWSRETURNED
 - ACTIVITYTOTALTIME
 - ESTIMATEDSQLCOST
 - CPUTIME
 - SQLROWSREAD

Anwenden von Arbeitsaktionen auf Datenbankaktivitäten

Nur ein einziges Arbeitsaktionsset kann auf eine Datenbank, Servicesuperklasse oder Workload angewendet werden.

Wenn Arbeit an den Datenserver übergeben wird, wird sie entweder mit einer benutzerdefinierten Workload oder mit der Standardworkload verknüpft und dann einer Serviceklasse zugeordnet.

Die folgende Abbildung zeigt, wie eine Arbeitsaktion auf eine Aktivität angewendet wird.

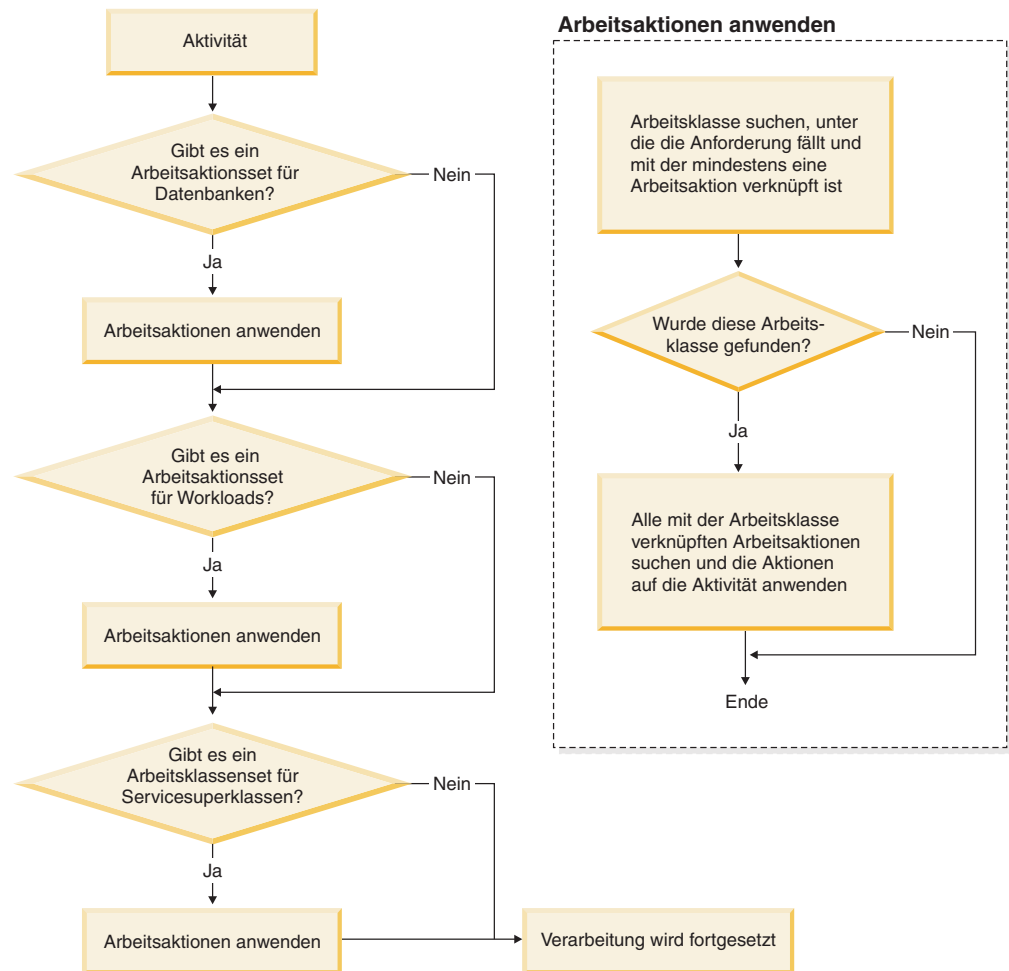


Abbildung 18. Anwenden einer Arbeitsaktion auf eine Aktivität

Eine Arbeitsaktion wird wie folgt einer Aktivität zugeordnet:

1. Wenn eine Aktivität einer Servicesuperklasse oder einer Serviceunterklasse zugeordnet wird, prüft der Datenserver, ob ein aktiviertes Arbeitsaktionsset auf Datenbankebene vorhanden ist.
2. Wenn ein aktiviertes Arbeitsaktionsset auf Datenbankebene vorhanden ist, prüft der Datenserver, ob die Aktivität unter eine der Arbeitsklassen im Arbeitsklassenset fällt, mit dem das Arbeitsaktionsset auf Datenbankebene verknüpft ist.
3. Wenn die Aktivität unter eine Arbeitsklasse fällt, auf die Arbeitsaktionen angewendet wurden, werden diese Arbeitsaktionen auf die Aktivität angewendet.
4. Der Datenserver prüft, ob ein aktiviertes Arbeitsaktionsset auf Workloadebene vorhanden ist. Ist dies der Fall, prüft der Datenserver, ob die Aktivität unter eine der Arbeitsklassen im Arbeitsklassenset fällt, mit dem das Arbeitsaktionsset auf Workloadebene verknüpft ist.
5. Wenn die Aktivität unter eine Arbeitsklasse fällt, auf die Arbeitsaktionen angewendet wurden, werden diese Arbeitsaktionen auf die Aktivität angewendet.
6. Wenn die Aktivität von der Workload einer Servicesuperklasse zugeordnet wird, prüft der Datenserver als nächstes, ob ein Arbeitsaktionsset auf die Servicesuperklasse angewendet ist.

7. Wenn ein Arbeitsaktionsset auf die Servicesuperklasse angewendet ist, prüft der Datenserver, ob die Aktivität unter eine der Arbeitsklassen im Arbeitsklassenset fällt, mit dem das Arbeitsaktionsset auf der Ebene der Servicesuperklasse verknüpft ist.
8. Wenn die Aktivität unter eine Arbeitsklasse fällt, auf die Arbeitsaktionen angewendet wurden, werden diese Arbeitsaktionen auf die Aktivität angewendet.

Beachten Sie, dass untergeordnete Aktivitäten einer gespeicherten Prozedur abhängig davon, ob die Klausel WITH NESTED oder WITHOUT NESTED in der Arbeitsaktionsdefinition angegeben ist, in derselben Serviceunterklasse oder in anderen Serviceunterklassen als die übergeordnete Aktivität ausgeführt werden, wenn eine Zuordnungsarbeitsaktion auf eine gespeicherte Prozedur angewendet wird.

In den folgenden Situationen ist eine Aktivität nicht von einem Arbeitsaktionsset betroffen:

- Aktivitäten fallen in die Standardsystemserviceklasse (SYSDEFAULTSYSTEMCLASS) und in die Standardwartungsserviceklasse (SYSDEFAULTMAINTENANCECLASS).
- Aktivitäten werden der Standardverwaltungsworkload (SYSDEFAULTADMWORKLOAD) zugeordnet.
- Aktivitäten befinden sich in einer Ladeoperation. Die Ladeoperation selbst durchläuft die Bewertung des Arbeitsaktionssets.
- Es handelt sich um untergeordnete Aktivitäten von gespeicherten Systemprozeduren. Die einzige Ausnahme ist die gespeicherte Prozedur SYSPROC.ADMIN_CMD. Untergeordnete Aktivitäten von SYSPROC.ADMIN_CMD durchlaufen die Bewertung des Arbeitsaktionssets.
- Das Arbeitsaktionsset ist inaktiviert.
- Die Workload ordnet die Aktivität direkt einer Serviceunterklasse zu.

Steuerung des gemeinsamen Zugriffs auf Workloadebene mithilfe von Arbeitsaktionssets

Auf der Workloadebene können Sie den gemeinsamen Zugriff von eingehender Arbeit mithilfe von Schwellenwerten für den gemeinsamen Zugriff steuern, die mit einem für die Workload definierten Arbeitsaktionsset angewendet werden. Sie können die Steuerung des gemeinsamen Zugriffs auf der Workloadebene mit der Prioritätssteuerung nach Verweildauer auf der Serviceklassenebene kombinieren.

Abbildung 1 zeigt ein Beispielszenario mithilfe von Arbeitsaktionssets zur Steuerung des gemeinsamen Zugriffs von eingehender Arbeit basierend auf der Quelle der Verbindung, während die gesamte Arbeit in der Datenbank mithilfe der Prioritätssteuerung nach Verweildauer gesteuert wird.

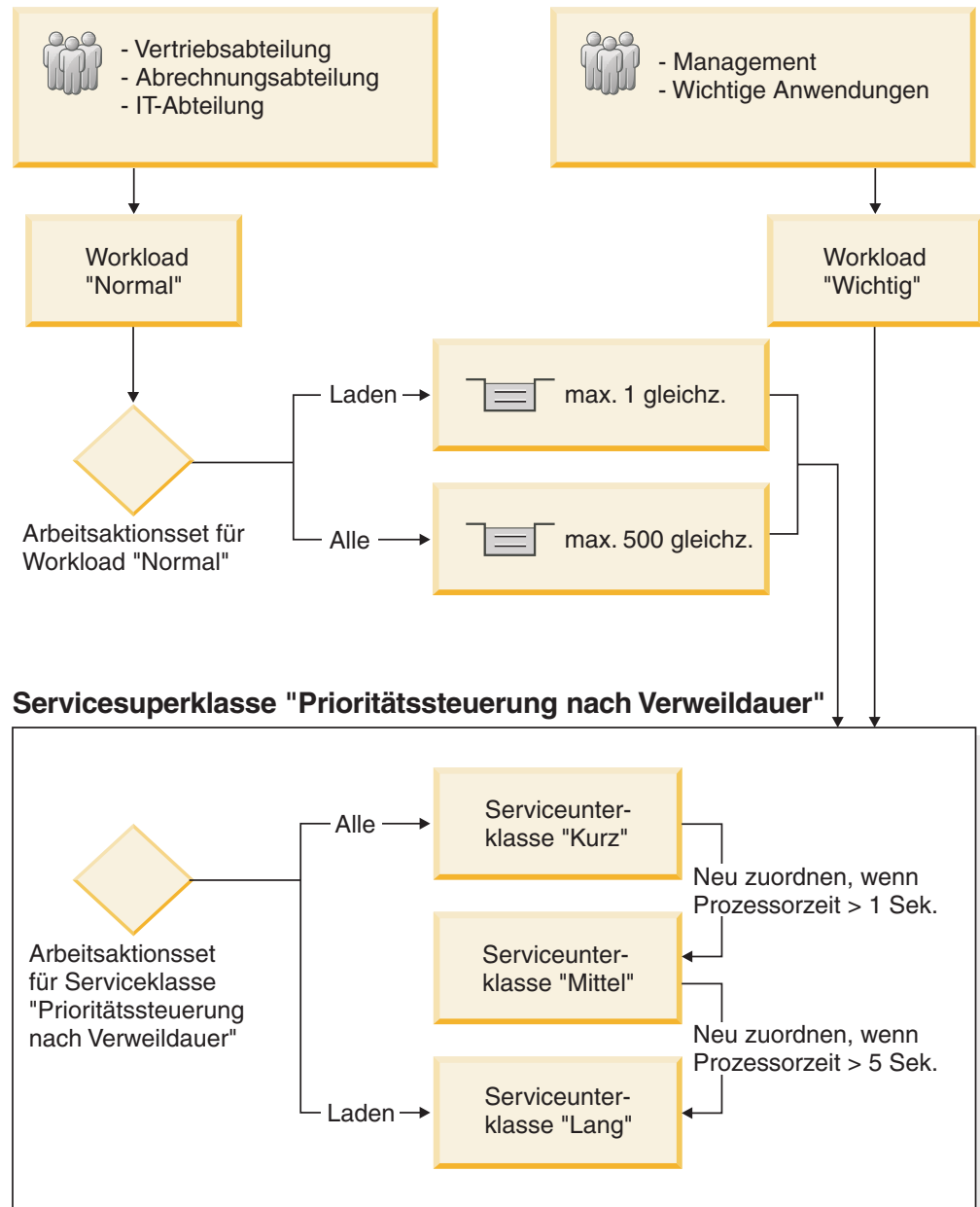


Abbildung 19. Steuerung des gemeinsamen Zugriffs auf Workloadenebene mithilfe von Arbeitsaktionssets

In diesem Beispielszenario werden zwei Workloads erstellt, um die von verschiedenen Quellen eingehende Arbeit anzugeben und zu unterscheiden. Verbindungen zu der Datenbank der Vertriebs-, Abrechnungs- und IT-Abteilung werden der Workload "Normal" zugeordnet. Verbindungen zu der Datenbank des Managements und von wichtigen Anwendungen werden der Workload "Wichtig" zugeordnet. Arbeit von der Workload "Wichtig" hat eine höhere Priorität und muss innerhalb kürzester Zeit abgeschlossen werden können. Um sicherzustellen, dass die Datenbank über genügend Kapazität für die Arbeit in der Workload "Normal" verfügt, werden Schwellenwerte für gemeinsamen Zugriff auf die Arbeit in der Workload "Normal" gesetzt. Ein Arbeitsaktionsset auf Workloadenebene namens "Normal" wird für die Workload "Normal" erstellt und auf ein Arbeitsklassenset angewendet, das zwei Arbeitsklassen hat. Ladeaktivitäten werden der einen Arbeitsklasse zugeordnet, alle anderen Aktivitäten werden der anderen Arbeitsklasse zugeordnet. Ein

Schwellenwert CONCURRENTDBCOORDACTIVITIES wird als Arbeitsaktion im Arbeitsaktionsset auf Workloadebene "Normal" erstellt, um immer nur jeweils eine Ladeaktivität im System zu ermöglichen und die anderen Ladeaktivitäten in die Warteschlange zu stellen. Darüber hinaus wird ein anderer Schwellenwert CONCURRENTDBCOORDACTIVITIES als Arbeitsaktion im Arbeitsaktionsset auf Workloadebene "Normal" erstellt, um maximal 500 gleichzeitig ablaufende Aktivitäten zu ermöglichen und Aktivitäten, die dieses Maximum überschreiten, in die Warteschlange zu stellen.

Anmerkung: Wenn als Schwellenwertaktion CONTINUE für einen Warteschlangenschwellenwert angegeben ist, hebt dies die Begrenzung der Warteschlangengröße auf, unabhängig von einem festen Wert, der für die Warteschlangengröße angegeben wird.

Verbindungen zur Datenbank von der Workload "Normal" sowie von der Workload "Wichtig" werden der Servicesuperklasse "Prioritätssteuerung nach Verweildauer" zugeordnet. Diese Servicesuperklasse wird zur Implementierung der Prioritätssteuerung nach Verweildauer erstellt, die kürzere Aktivitäten bevorzugt. Das Arbeitsaktionsset für Serviceklassen "Prioritätssteuerung nach Verweildauer" wird für die Servicesuperklasse "Prioritätssteuerung nach Verweildauer" erstellt, um die lang laufenden Ladeaktivitäten von allen Aktivitäten mit kurzer Laufzeit zu trennen. Alle Aktivitäten, außer Ladeaktivitäten, werden der Serviceunterklasse "Kurz" zugeordnet. Die Serviceunterklasse "Kurz" ist mit der höchsten Agenten-, Vorablesezugriffs- und Pufferpoolpriorität konfiguriert. Ein Schwellenwert CPUTIMEINSC wird für die Serviceunterklasse "Kurz" erstellt, um eine Aktivität neu zur Serviceunterklasse "Mittel" zuzuordnen, wenn diese mehr als 1 Sekunde Prozessorzeit in der Serviceunterklasse "Kurz" verbraucht. Die Serviceunterklasse "Mittel" hat eine mittlere Agenten-, Vorablesezugriffs- und Pufferpoolpriorität. Ein Schwellenwert CPUTIMEINSC wird für die Serviceunterklasse "Mittel" erstellt, um eine Aktivität neu zur Serviceunterklasse "Lang" zuzuordnen, wenn diese mehr als 5 Sekunden Prozessorzeit in der Serviceunterklasse "Mittel" verbraucht. Die Serviceunterklasse "Lang" hat die niedrigste Agenten-, Vorablesezugriffs- und Pufferpoolpriorität. Ladeaktivitäten werden direkt vom Arbeitsaktionsset für Serviceklassen "Prioritätssteuerung nach Verweildauer" der Serviceunterklasse "Lang" zugeordnet, weil Ladeaktivitäten eine lange Laufzeit haben können, ressourcenintensiv sein können und ihr Abschluss weniger zeitkritisch sein kann.

Vergleich von Workload und Arbeitsaktionsset

Je nach Art der Kontrolle, die Sie über Ihre Datenbankaktivitäten haben wollen, können Sie ausschließlich Workloads oder Workloads und Arbeitsklassen (bei der Verwendung mit Arbeitsaktionen) verwenden, um Aktivitäten zu Serviceklassen zuzuordnen.

Mit Workloads werden Anforderungen basierend auf Verbindungsattributen angegeben und einer Serviceklasse zugeordnet. Workloads sind die primäre Methode, um Arbeit zur Ausführung an eine bestimmte DB2-Serviceklasse weiterzuleiten. Wenn Sie weiter eingrenzen wollen, wie Anforderungen angegeben werden, können Sie Arbeitsklassen verwenden, um die Aktivitäten basierend auf ihrem Typ und anderen Aktivitätsattributen zu klassifizieren. Beispielsweise können Sie READ-, WRITE- und LOAD-Aktivitäten in unterschiedlichen Arbeitsklassen klassifizieren und jeden Aktivitätstyp unterschiedlich behandeln lassen.

Wenn Sie Arbeitsklassen verwenden (die in Arbeitsklassensets gruppiert sind), können Sie über Arbeitsaktionen die unterschiedlichen Aktivitätstypen steuern. Sie können z. B. eine Arbeitsaktion in einem Arbeitsaktionsset verwenden, das für

eine Servicesuperklasse definiert ist, um einen bestimmten Typ von Aktivität einer Serviceunterklasse zuzuordnen. In einem Arbeitsaktionsset, das für die Datenbank oder die Workload definiert ist, können Sie eine Arbeitsaktion definieren, um eine als Schwellenwert bezeichnete Steuerung anzuwenden, die sicherstellt, dass derselbe Typ von Aktivität nicht gegen bestimmte Bedingungen verstößt.

Arbeitsaktionen werden in Arbeitsaktionssets gruppiert. Ein einzelnes Arbeitsaktionsset kann sich auf Aktivitäten in der Datenbank, in einer Servicesuperklasse oder in einer Workload beziehen. Dasselbe Arbeitsaktionsset kann sich jedoch nicht auf mehrere Objekte beziehen. Arbeitsklassensets und Arbeitsaktionssets arbeiten zusammen. Eine Arbeitsklasse muss zur Kategorisierung einer Aktivität als bestimmter Typ von Arbeit vorhanden sein, bevor eine Arbeitsaktion darauf angewendet werden kann. Ein Arbeitsklassenset kann mit mehreren Arbeitsaktionssets verknüpft sein, aber ein Arbeitsaktionsset kann nur mit einem Arbeitsklassenset verknüpft sein.

Abbildung 1 zeigt ein Beispiel einer DB2-Workload-Manager-Implementierung mit Workloads und Arbeitsaktionssets. In dieser Abbildung wird davon ausgegangen, dass eine Anforderung der Workload WL_A zugeordnet wird, basierend auf den Verbindungsattributen der Verbindung, die die Anforderung übergeben hat. Die Workload WL_A definiert, dass die Anforderung in der Servicesuperklasse SC_A ausgeführt werden soll. Nehmen wir an, dass eine Arbeitsklasse im Arbeitsklassenset WCS_1 dem Arbeitstyp entspricht, den die Anforderung ausführt, die mit der Workload WL_A verknüpft ist.

Nehmen wir nun z. B. an, dass eine Aktivität, die die Kataloge nicht aktualisiert (eine READ-Aktivität), in das System eintritt. Das Arbeitsaktionsset WAS_1 auf Datenbankebene (das mit dem Arbeitsklassenset WCS_1 verknüpft ist) enthält eine Arbeitsaktion, die auf die Arbeitsklasse READ angewendet wird. Die Anforderung wird anschließend (von der Workload WL_A) der Servicesuperklasse SC_A zugeordnet. Hier trifft die Anforderung auf das Arbeitsaktionsset WAS_2 auf der Ebene der Servicesuperklasse, das auch mit dem Arbeitsklassenset WCS_1 verknüpft ist und sich auf Aktivitäten in der Servicesuperklasse SC_A bezieht. Dieses Arbeitsaktionsset enthält eine Zuordnungsarbeitsaktion, die auch auf die Arbeitsklasse READ angewendet wird, sodass alle READ-Aktivitäten der Serviceunterklasse SSC_1a in der Servicesuperklasse SC_A zugeordnet werden.

Eine ähnliche Situation tritt mit der Anforderung auf, die (ebenfalls basierend auf ihren Verbindungsattributen) mit der Workload WL_B verknüpft ist. Die Workload WL_B ordnet Aktivitäten der Servicesuperklasse SC_B zu. Nehmen wir an, dass die Anforderung sich auf eine LOAD-Aktivität bezieht und dass das Arbeitsklassenset WCS_2 eine Arbeitsklasse für LOAD-Aktivitäten enthält. Das Arbeitsklassenset WCS_2 ist mit dem Arbeitsaktionsset WAS_3 auf der Ebene der Servicesuperklasse verknüpft, das auf Aktivitäten in der Servicesuperklasse SC_B angewendet wird. Nehmen wir an, dass das Arbeitsaktionsset WAS_3 eine Zuordnungsarbeitsaktion enthält, die auf die Arbeitsklasse LOAD angewendet wird, sodass sie beim Zuordnen der LOAD-Aktivität zur Servicesuperklasse SC_B durch die Workload WL_B von der Arbeitsaktion zur Ausführung der Serviceunterklasse SSC_1b zugeordnet wird.

Der Zweck von Workload WL_C in diesem Beispiel ist es, eingehende Anforderungen direkt der Serviceunterklasse SSC_1b zuzuordnen, unabhängig vom Arbeitsaktionsset WAS_3 auf Servicesuperklassenebene und seiner Zuordnungsarbeitsaktion.

Wenn eine eingehende Anforderung Workload WL_C zugeordnet ist, bei der es sich um eine LOAD-Aktivität handelt, wird diese Anforderung ebenfalls direkt der Serviceunterklasse SSC_1b zur Ausführung zugeordnet und die für die Arbeitsklasse LOAD geltende Zuordnungsarbeitsaktion hat keine Auswirkung auf sie.

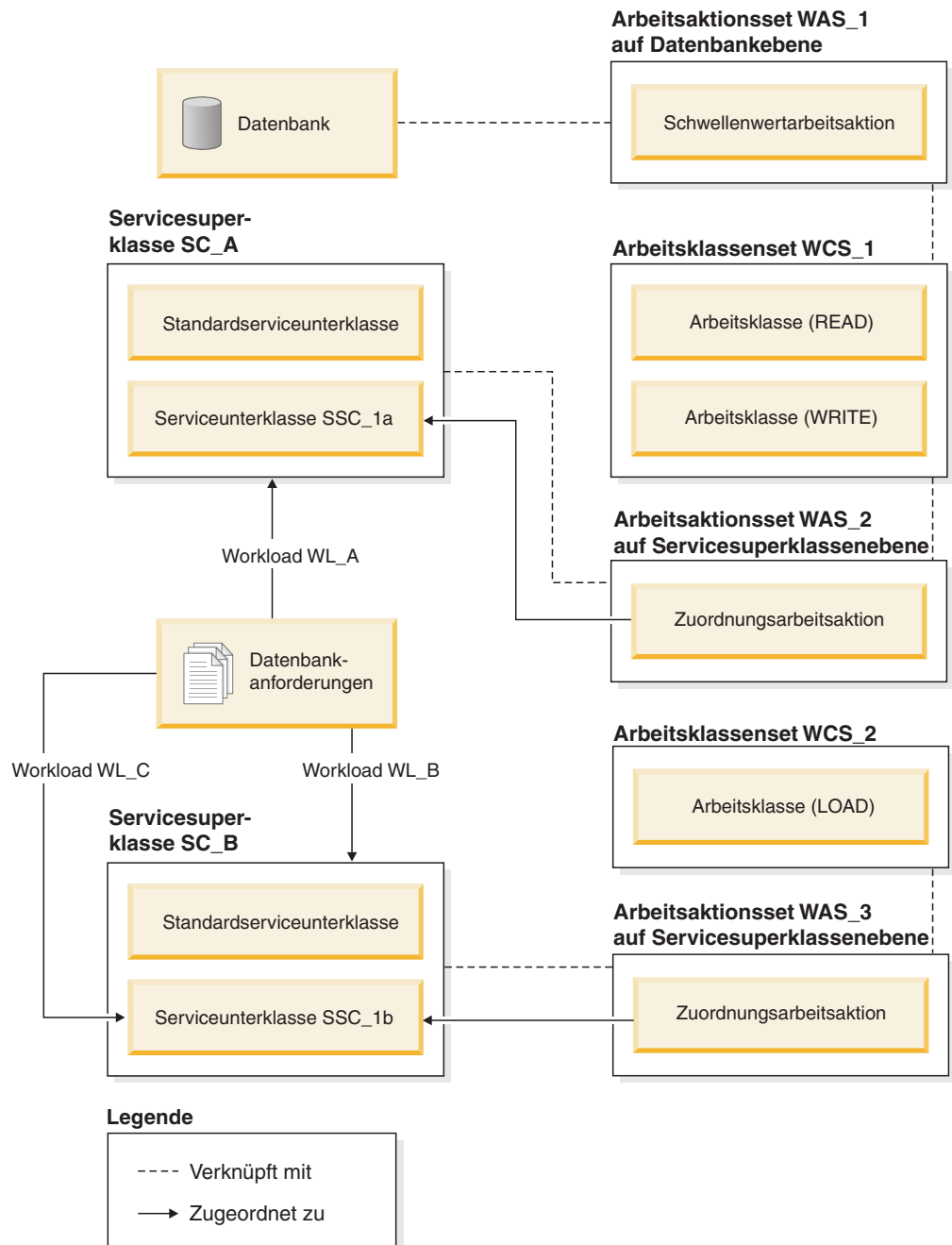


Abbildung 20. Workloads und Arbeitsaktionssets

Beispiel: Verwenden von Datenbankarbeitsaktionssets und Datenbankschwellenwerten

Dieses Beispiel zeigt unterschiedliche Strategien zur Verwendung von Arbeitsaktionssets und Schwellenwerten zur Steuerung der von DB2-Aktivitäten verbrauchten Ressourcen. Vor dem Erstellen von DB2-Workload-Management-Objekten müssen Sie verstanden haben, wie diese eingesetzt werden.

Nehmen wir an, dass Sie ein Arbeitsklassenset namens ALLSQL haben und dass dieses die folgenden Arbeitsklassen in der aufgeführten Reihenfolge enthält:

1. SMALLDML, für alle SQL-Anweisungen vom Typ DML mit einem geschätzten Aufwand von weniger als 1.000 Timeron
2. MEDDML, für alle SQL-Anweisungen vom Typ DML mit einem geschätzten Aufwand zwischen 1.000 und 20.000 Timeron
3. LARGEDML, für alle SQL-Anweisungen vom Typ DML mit einem geschätzten Aufwand von mehr als 20.000 Timeron
4. ALLDDL, für alle SQL-Anweisungen vom Typ DDL
5. ALLACTIVITY, für die gesamte Datenbankaktivität

Die folgenden SQL-Anweisungen erstellen das Arbeitsklassenset und die Arbeitsklassen:

```
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)
```

Auf diese Arbeitsklassen wurden bereits Arbeitsaktionen angewendet, z. B. COUNT ACTIVITY, COLLECT und Schwellenwerte (die keine ACTIVITYTOTAL-TIME-Schwellenwerte sind).

Nehmen wir an, dass Sie die Ausführung von großen DML-Aktivitäten auf maximal 5 Stunden begrenzen wollen. Die Ausführung aller anderen SQL-Aktivitäten darf nicht länger als 30 Minuten dauern. Die folgenden beiden Beispiele zeigen Methoden, mit denen dieses Ziel erreicht werden kann.

Methode 1

Eine Methode besteht darin, auf der Datenbankebene ein Arbeitsaktionsset mit Arbeitsaktionen zu erstellen, die für jede Arbeitsklasse einen Schwellenwert ACTIVITYTOTALTIME angeben:

Tabelle 33. Schwellenwert ACTIVITYTOTALTIME für jede Arbeitsklasse

Arbeitsaktion	Angewendet auf Arbeitsklasse	Schwellenwerttyp und -wert	Aktionen
SMALLDMLTIMEALLOWED	SMALLDML	ACTIVITYTOTALTIME > 30 MINUTES	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
MEDDMLTIMEALLOWED	MEDDML	ACTIVITYTOTALTIME > 30 MINUTES	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
LARGEDMLTIMEALLOWED	LARGEDML	ACTIVITYTOTALTIME > 5 HOURS	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
ALLDDLTIMEALLOWED	ALLDDL	ACTIVITYTOTALTIME > 30 minutes	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten
ALLACTIVITYTIMEALLOWED	ALLACTIVITY	ACTIVITYTOTALTIME > 30 minutes	<ul style="list-style-type: none"> • Stoppen der Ausführung • Erfassen von Aktivitätsdaten

SQL-Anweisungen für diese Methode:

```
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)
```

Methode 2

Eine weitere Methode wäre, nur eine Arbeitsklasse (LARGEDML) zu verwenden und dann ein Arbeitsaktionsset für die Datenbank mit einer Arbeitsaktion (LARGEDMLTIMEALLOWED) zu erstellen, die auf die Arbeitsklasse angewendet wird.

Table 34. Arbeitsaktion LARGEDMLTIMEALLOWED angewendet auf die Arbeitsklasse LARGEDML

Arbeitsaktion	Angewendet auf Arbeitsklasse	Schwellenwerttyp und -wert	Aktion
LARGEDMLTIMEALLOWED	LARGEDML	ACTIVITYTOTALTIME < 5 HOURS	<ul style="list-style-type: none">• Stoppen der Ausführung• Erfassen von Aktivitätsdaten

Sie würden dann einen Schwellenwert ACTIVITYTOTALTIME von weniger als 31 Minuten auf die Datenbank anwenden. Mit dieser Methode wird der 5-Stunden-Schwellenwert nur auf Aktivitäten angewendet, die der Arbeitsklasse LARGEDML entsprechen. Für andere Aktivitäten gilt der Datenbankschwellenwert ACTIVITYTOTALTIME von weniger als 31 Minuten.

SQL-Anweisungen für diese Methode:

```
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
```

Beispiel: Festlegen des Typs der auszuführenden Arbeit mithilfe von Arbeitsaktionssets

Durch die Verwendung von Arbeitsklassensets, Arbeitsklassen, Arbeitsaktionssets, Arbeitsaktionen und einigen DB2-Workload-Manager-Überwachungsfeatures können Sie die unterschiedlichen Typen von Arbeit auf Ihrem System sowie die Verteilung der Arbeit ermitteln.

Eine Methode zur Durchführung dieser Aufgabe wird hier beschrieben. Erstellen Sie zuerst ein Arbeitsklassenset, das Arbeitsklassen für die unterschiedlichen Typen von Arbeit enthält, an denen Sie interessiert sind. Wenn Sie z. B. wissen wollen, wie viele READ-, WRITE-, DDL- und LOAD-Aktivitäten auf Ihrem System ausgeführt werden, erstellen Sie ein Arbeitsklassenset ACTIVITYTYPES (siehe folgendes Beispiel):

```
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)
```

Als Nächstes erstellen Sie ein Arbeitsaktionsset auf Datenbankebene, COUNTACTIONS, das Sie auf das Arbeitsklassenset ACTIVITYTYPES anwenden. Das Arbeitsaktionsset enthält eine Arbeitsaktion COUNT ACTIVITY für jede Arbeitsklasse im Arbeitsklassenset ACTIVITYTYPES (siehe folgendes Beispiel):

```
CREATE WORK ACTION SET COUNTACTIONS FOR DATABASE USING WORK CLASS SET ACTIVITYTYPES
(WORK ACTION COUNTREAD ON WORK CLASSREADWC 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)
```

Nachdem ausreichend Zeit vergangen ist, können Sie die Anzahl der einzelnen ausgeführten Aktivitätstypen mit der Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS ermitteln:

```
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
```

Steuern von Arbeit mit Schwellenwerten

Mit Schwellenwerten können Sie die Stabilität im System verwalten, indem Sie Arbeit identifizieren, die sich abnormal verhält. Sie können abnormales Verhalten basierend auf der projizierten Auswirkung vor der Arbeitsausführung prädiktiv identifizieren. Sie können abnormales Verhalten auch während der Arbeitsausführung und des Verbrauchs von Ressourcen reaktiv identifizieren.

Ein Beispiel für Arbeit, die mithilfe von Schwellenwerten gesteuert werden kann, ist eine Abfrage, die viel Prozessorzeit zu Lasten der anderen auf dem System ausgeführten Arbeit verbraucht. Eine solche Abfrage kann gesteuert werden, bevor ihre Ausführung beginnt, basierend auf dem geschätzten Aufwand, oder wenn die Arbeit bereits ausgeführt wird und mehr als den zulässigen Anteil von Ressourcen verbraucht.

Typen von Schwellenwerten

Verbindungsschwellenwerte

Verwenden Sie einen Verbindungsschwellenwert, wenn Sie beschränken wollen, wie lange eine Verbindung inaktiv sein darf. Diese Schwellenwerte können verwendet werden, um Verbindungen zu erkennen, die zu lange inaktiv sind.

Tabelle 35. Verbindungsschwellenwerte

Schwellenwert	Beschreibung
CONNECTIONIDLETIME	Steuert die Zeitspanne, während der eine Verbindung im Leerlauf und nicht für Benutzeranforderungen aktiv ist. Mit diesem Schwellenwert können Sie eine ineffiziente Nutzung von Datenserverressourcen und Anwendungswartebedingungen erkennen.

UOW-Schwellenwerte

Wenn Sie die Ausführungsdauer einer UOW begrenzen wollen, verwenden Sie einen UOW-Schwellenwert. Diese Schwellenwerte begrenzen die Höchstdauer einer UOW in der DB2-Engine und können zur Ermittlung von UOWs verwendet werden, deren Beendigung zu lange dauert.

Tabelle 36. UOW-Schwellenwerte

Schwellenwert	Beschreibung
UOWTOTALTIME	Steuert die Ausführungsdauer einer UOW.

Aktivitätsschwellenwerte

Aktivitätsschwellenwerte sind eine der Möglichkeiten zum Beschränken der Auswirkung, die bestimmte Aktivitäten auf die Ausführung des Daten-servers haben können. Zu viel Ausführungszeit, anormal hohe zurückgegebene Datenmengen oder anormal hoher Ressourcenverbrauch sind alles Beispiele für Warnzeichen dafür, dass potenziell problematische Aktivitäten möglicherweise zu viele Ressourcen verbrauchen. Dies können Sie mit Aktivitätsschwellenwerten steuern.

Tabelle 37. Aktivitätsschwellenwerte

Schwellenwert	Beschreibung
ACTIVITYTOTALTIME	Steuert die Zeitspanne, die eine beliebige angegebene Aktivität von der Übergabe bis zur Fertigstellung verbringen darf (Ausführungszeit plus Wartezeit). Mit diesem Schwellenwert können Sie Jobs erkennen, die anormal lange bis zu ihrer Fertigstellung benötigen.
CPUTIME	Steuert den maximalen Anteil von kombinierter Benutzer- und Systemprozessorzeit, die eine Aktivität in einem bestimmten Member während der Ausführung der Aktivität verbrauchen darf. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die zu viele Prozessorressourcen verbrauchen.
CPUTIMEINSC	Steuert den maximalen Anteil von kombinierter Benutzer- und Systemprozessorzeit, die eine Aktivität in einem bestimmten Member während ihrer Ausführung in einer bestimmten Serviceunterklasse verbrauchen darf. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die zu viele Prozessorressourcen in der aktuellen Serviceklasse verbrauchen.
DATATAGINSC	Steuert die Daten, die von einer Aktivität während der Ausführung einer bestimmten Serviceunterklasse (nicht) verarbeitet werden können.
ESTIMATEDSQLCOST	Steuert DML-Aktivitäten, bei denen das Abfrageoptimierungsprogramm einen großen geschätzten Aufwand ermittelt. Mit diesem Schwellenwert können Sie SQL-Code mit potenziell hoher Ressourcenauslastung vorhersagen, bevor dieser auf dem System ausgeführt wird, sowie schlecht geschriebenen SQL-Code identifizieren.
SQLROWSREAD	Steuert die maximale Anzahl der Zeilen, die von einer Aktivität in einem beliebigen Member gelesen werden können. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die übermäßig viele Zeilen lesen.
SQLROWSREADINSC	Steuert die maximale Anzahl der Zeilen, die von einer Aktivität in einem bestimmten Member bei der Ausführung in einer bestimmten Serviceunterklasse gelesen werden können. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die übermäßig viele Zeilen in der aktuellen Serviceklasse lesen.
SQLROWSRETURNED	Steuert die Anzahl der Zeilen, die bei der SQL-Ausführung zurückgegeben werden. Mit diesem Schwellenwert können Sie ermitteln, wann das Datenvolumen eine angemessene Schwelle überschreitet.
SQLTEMPSPACE	Steuert die Größe des Tabellenbereichs für temporäre Tabellen, den eine bestimmte Aktivität in einem Member verbrauchen darf. Mit diesem Schwellenwert können Sie verhindern, dass bestimmte SQL-Anweisungen einen unverhältnismäßig großen Anteil des temporären Speicherbereichs verbrauchen und so andere Arbeit behindern.

Der Datenserver betrachtet Anforderungen von Dienstprogrammen wie REORGCHK, IMPORT und EXPORT als Benutzerlogik. Auf diese Anforderungen werden daher alle definierten Schwellenwerte angewendet.

Zusammengefasste Schwellenwerte

Wenn Sie die Gesamtauswirkung steuern wollen, die bestimmte Aktivitäten, Workloads oder Verbindungen auf Ihrem Datenserver haben, können Sie zusammengefasste Schwellenwerte definieren. Zusammengefasste Schwellenwerte erzwingen oft, aber nicht immer, eine Steuerung des gemeinsamen Zugriffs, wenn Sie die Anzahl bestimmter gleichzeitig ausgeführter Aktivitäten beschränken müssen. Einige zusammengefasste Schwellenwerte haben eine integrierte Warteschlange und werden auch Warteschlangenschwellenwerte genannt.

Tabelle 38. Zusammengefasste Schwellenwerte

Schwellenwert	Beschreibung
AGGSQLTEMPSPACE	Steuert den maximalen Anteil des Tabellenbereichs für temporäre Systemtabellen, der insgesamt über alle Aktivitäten in der Serviceunterklasse verbraucht werden darf. Mit diesem Schwellenwert können Sie Aktivitäten erkennen und steuern, die zu einer Serviceunterklasse gehören, deren Aktivitäten zusammen über alle Aktivitäten in der Serviceunterklasse zu viel Tabellenbereich für temporäre Systemtabellen verbrauchen.
CONCURRENTWORKLOADOCCURRENCES	Steuert die Anzahl aktiver Vorkommen einer Workload, die in einem Koordinatormember gleichzeitig ausgeführt werden können. Mit diesem Schwellenwert können Sie die Verteilung von Verbindungen von einer bestimmten Quelle steuern.
CONCURRENTWORKLOADACTIVITIES	Steuert die Anzahl der einzelnen Aktivitäten, die in einem Workloadvorkommen ausgeführt werden können. Mit diesem Schwellenwert können Sie Arbeit in einem einzigen Workloadvorkommen beschränken.
CONCURRENTDBCOORDACTIVITIES	Steuert die Anzahl gleichzeitig ablaufender Aktivitäten in der Domäne, der der Schwellenwert zugeordnet ist (Datenbank, Arbeitsaktion, Servicesuperklasse oder Serviceunterklasse).
TOTALMEMBERCONNECTIONS	Steuert die Anzahl der Datenbankverbindungen, die gleichzeitig zu einem bestimmten Member hergestellt werden können. Mit diesem Schwellenwert können Sie verhindern, dass ein bestimmtes Member überlastet wird.
TOTALSCMEMBERCONNECTIONS	Steuert die Anzahl der Datenbankverbindungen zu einem bestimmten Member für Arbeit, die in einer bestimmten Serviceklasse gleichzeitig ausgeführt werden kann. Dieser Schwellenwert ist dem Schwellenwert TOTALMEMBERCONNECTIONS ähnlich, aber feiner unterteilt, da die Verbindung mit einer Serviceklasse verknüpft ist.

Für die zusammengefassten Schwellenwerte, die die Steuerung des gemeinsamen Zugriffs unterstützen, wird sie über ein System von Ausführungstickets bereitgestellt. Jede ankommende Aktivität muss ein Ticket vom entsprechenden Schwellenwert für gemeinsamen Zugriff anfordern, bevor ihre Ausführung beginnen kann. Wenn alle Tickets verbraucht sind, werden zusätzliche Aktivitäten in die Warteschlange gestellt, bis wieder ein Ticket verfügbar wird oder ein Fehler zurückgegeben wird, je nach Ihrer Definition des Schwellenwerts. Wenn die Warteschlangensteuerung für den Schwellenwert für gemeinsamen Zugriff aktiviert ist, wird ein Ticket von einer Aktivität, deren Ausführung abgeschlossen ist, an eine andere Aktivität übergeben, die sich noch in der Warteschlange befindet. Diese Aktivität

kann dann die Warteschlange verlassen und die Ausführung beginnen. Wie viele Tickets pro Schwellenwert für gemeinsamen Zugriff verfügbar sind, hängt von Ihrer Definition des Schwellenwerts ab. Wenn Sie beispielsweise einen Schwellenwert `CONCURRENTDBCOORDACTIVITIES` definiert haben, um die Anzahl der gleichzeitig ausgeführten Datenbankaktivitäten auf 10 zu begrenzen, sind 10 Ausführungstickets verfügbar.

Bei gespeicherten Prozeduren werden Aktivitätsschwellenwerte auf die gespeicherte Prozedur und auf ihre untergeordneten Aktivitäten angewendet. Schwellenwerte für gemeinsamen Zugriff werden nur auf die untergeordneten Aktivitäten der gespeicherten Prozedur, nicht jedoch auf die Anweisung `CALL` selbst angewendet.

Maßnahmen bei Schwellenwertverstößen

Welche Aktion bei einem Schwellenwertverstoß dynamisch ausgeführt wird, hängt davon ab, wie Sie den Schwellenwert definieren.

Ausführung stoppen (STOP EXECUTION)

Eine übliche Aktion bei einem Schwellenwertverstoß ist das Stoppen der Aktivität. In diesem Fall wird ein Fehlercode an die übergebende Anwendung zurückgegeben, der angibt, dass gegen den Schwellenwert verstoßen wurde. Beachten Sie, dass eine Aktion `STOP EXECUTION` bei den Schwellenwerten `TOTALMEMBERCONNECTIONS` und `TOTALSCMEMBERCONNECTIONS` verhindert, dass eine Verbindung hergestellt wird. Beim Schwellenwert `CONNECTIONIDLETIME` wird die Verbindung geschlossen. Bei `CONCURRENTWORKLOADOCCURRENCES` wird verhindert, dass ein neues Workloadvorkommen erstellt wird. Bei allen aktivitätsbezogenen Schwellenwerten wird die weitere Ausführung der Aktivität gestoppt. Wenn ein `THRESHOLD VIOLATIONS`-Ereignismonitor aktiv ist, wird ein Datensatz in den Ereignismonitor geschrieben, der angibt, dass gegen den Schwellenwert verstoßen wurde.

Ausführung fortsetzen (CONTINUE)

In einigen Situationen ist das Stoppen einer Aktivität eine zu harsche Maßnahme. Eine bevorzugte Maßnahme ist, die Ausführung der Aktivität weiterlaufen zu lassen und die relevanten Daten für einen Administrator zu erfassen, der dann später eine Analyse durchführen kann, um zu ermitteln, wie sich das erneute Auftreten dieses Zustands verhindern lässt. In dieser Situation wird kein Fehlercode an die übergebende Anwendung zurückgegeben. Wenn die Ausführung der Aktion fortgesetzt werden soll, erhält der Benutzer keinen Hinweis auf den Schwellenwertverstoß. Wenn ein `THRESHOLD VIOLATIONS`-Ereignismonitor aktiv ist, wird ein Datensatz in den Ereignismonitor geschrieben. Wenn als Schwellenwertaktion `CONTINUE` für einen Warteschlangenschwellenwert angegeben ist, hebt dies die Begrenzung der Warteschlangengröße auf, unabhängig von einem festen Wert, den Sie angeben.

Anwendungsende erzwingen (FORCE APPLICATION)

Sie können angeben, dass die lokalen bzw. fernen Benutzer oder Anwendungen bei einem Verstoß gegen den Schwellenwert `UOWTOTALTIME` zwangsweise aus dem System abgemeldet werden. Davon profitieren andere Anwendungen, die dieselben Serverressourcen verwenden wollen.

Aktivität neu zuordnen (REMAP ACTIVITY TO)

Wenn eine Aktivität einen bestimmten Grenzwert nicht einhält, möchten Sie möglicherweise einfach eine andere Ressourcensteuerung darauf anwenden. Die Aktivität soll aber dennoch weiter ausgeführt werden. Eine

solche Maßnahme erlaubt es Ihnen, den Anteil der Ressourcen, die eine Aktivität während ihrer Laufzeit verbrauchen kann, zu erhöhen oder zu verringern. In diesem Fall darf eine bereits ausgeführte Aktivität weiter ausgeführt werden, ohne dass eine Meldung an den Benutzer oder die Anwendung über den Schwellenwertverstoß erfolgt, obwohl die Aktivität jetzt mit anderen Ressourcen ausgeführt wird. Die Neuuzuordnung ist verfügbar für alle serviceklasseninternen Schwellenwerte wie CPUTIMEINSC, SQL-ROWSREADINSC und DATATAGINSC.

Daten erfassen (COLLECT ACTIVITY DATA)

Bei einigen Schwellenwertverstößen werden Daten erfasst. Standardmäßig wird ein Verstoß gegen einen Aktivitätsschwellenwert in einem aktivierten Ereignismonitor für Schwellenwertverstöße aufgezeichnet. Wenn Sie ausführlichere Informationen zu der Aktivität wünschen, die gegen den Schwellenwert verstoßen hat, können Sie mit der Klausel COLLECT ACTIVITY DATA anfordern, dass Informationen zur Aktivität in den aktiven Aktivitätsereignismonitor geschrieben werden, wenn die Ausführung der Aktivität abgeschlossen ist.

Schwellenwertdomäne und Umsetzungsbereich

Jeder Schwellenwert arbeitet in einer Domäne. Nur Aktivitäten, die in der Domäne eines Schwellenwerts stattfinden, sind von diesem Schwellenwert betroffen.

Es gibt die folgenden Schwellenwertdomänen:

- Datenbank
- Servicesuperklasse
- Serviceunterklasse
- Arbeitsaktion
- Workload
- Anweisung

In jeder dieser Schwellenwertdomänen hat ein Schwellenwert einen Bereich, über den er erzwungen werden kann, wie z. B. ein einzelnes Workloadvorkommen, ein Member, die Ausführung einer bestimmten Anweisung oder alle Member. Er wird als *Umsetzungsbereich* des Schwellenwerts bezeichnet. Beispiel: Zusammengefasste Schwellenwerte von Serviceklassen können einen von zwei Umsetzungsbereichen haben: Datenbank und Member. Ein Beispiel eines zusammengefassten Schwellenwerts, der nur auf der Ebene des Members angewendet wird, ist die maximale Anzahl gleichzeitiger Verbindungen für eine Servicesuperklasse in einem Member (TOTALSCMEMBERCONNECTIONS). Analog dazu zeigt die folgende Tabelle, dass Sie den Schwellenwert für die Prozessorzeit (CPUTIME) auf der Datenbank-, Superklassen-, Unterklassen-, Arbeitsaktions- oder Workloaddomäne angeben können und dass er pro Member erzwungen wird. Die Obergrenze gibt also den maximalen Anteil von Benutzer- und Systemprozessorzeit pro Member an, den eine Aktivität verwenden kann.

Tabelle 39. Schwellenwertdomänen und Umsetzungsbereiche

Schwellenwertdomäne	Schwellenwerte mit einer Datenbank als Umsetzungsbereich	Schwellenwerte mit einem Member als Umsetzungsbereich	Schwellenwerte mit einem Workloadvorkommen als Umsetzungsbereich
Datenbank	<ul style="list-style-type: none"> • „ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143 • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „CONNECTIONIDLETIME (Schwellenwert)“ auf Seite 141 • „ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147 • „SQLROWSRETURNED (Schwellenwert)“ auf Seite 150 • „UOWTOTALTIME (Schwellenwert)“ auf Seite 160 	<ul style="list-style-type: none"> • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153(Eine Datenbank als Umsetzungsbereich ist nur in anderen Umgebungen als DB2 pureScale-Umgebungen zulässig. In einer DB2 pureScale-Umgebung ist nur ein Member als Umsetzungsbereich zulässig.) • „CPUTIME (Schwellenwert)“ auf Seite 144 • „SQLROWSREAD (Schwellenwert)“ auf Seite 148 • „SQLTEMPSPACE (Schwellenwert)“ auf Seite 151 • „TOTALMEMBERCONNECTIONS (Schwellenwert)“ auf Seite 158 	Nicht zutreffend
Arbeitsaktion	<ul style="list-style-type: none"> • „ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143 • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147 • „SQLROWSRETURNED (Schwellenwert)“ auf Seite 150 	<ul style="list-style-type: none"> • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „CPUTIME (Schwellenwert)“ auf Seite 144 • „SQLROWSREAD (Schwellenwert)“ auf Seite 148 • „SQLTEMPSPACE (Schwellenwert)“ auf Seite 151 	Nicht zutreffend
Service-superklasse	<ul style="list-style-type: none"> • „ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143 • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „CONNECTIONIDLETIME (Schwellenwert)“ auf Seite 141 • „ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147 • „SQLROWSRETURNED (Schwellenwert)“ auf Seite 150 • „UOWTOTALTIME (Schwellenwert)“ auf Seite 160 	<ul style="list-style-type: none"> • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „CPUTIME (Schwellenwert)“ auf Seite 144 • „SQLROWSREAD (Schwellenwert)“ auf Seite 148 • „SQLTEMPSPACE (Schwellenwert)“ auf Seite 151 • „TOTALSCMEMBERCONNECTIONS (Schwellenwert)“ auf Seite 159 	Nicht zutreffend
Serviceunterklasse	<ul style="list-style-type: none"> • „ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143 • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147 • „SQLROWSRETURNED (Schwellenwert)“ auf Seite 150 	<ul style="list-style-type: none"> • „AGGSQLTEMPSPACE (Schwellenwert)“ auf Seite 152 • „CONCURRENTDBCOORDACTIVITIES (Schwellenwert)“ auf Seite 153¹ • „CPUTIME (Schwellenwert)“ auf Seite 144 • „CPUTIMEINSC (Schwellenwert)“ auf Seite 145 • „DATATAGINSC (Schwellenwert)“ auf Seite 146 • „SQLROWSREAD (Schwellenwert)“ auf Seite 148 • „SQLROWSREADINSC (Schwellenwert)“ auf Seite 149 • „SQLTEMPSPACE (Schwellenwert)“ auf Seite 151 	Nicht zutreffend
Workload	<ul style="list-style-type: none"> • „ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143 • „ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147 • „SQLROWSRETURNED (Schwellenwert)“ auf Seite 150 • „UOWTOTALTIME (Schwellenwert)“ auf Seite 160 	<ul style="list-style-type: none"> • „CONCURRENTWORKLOADOCCURRENCES (Schwellenwert)“ auf Seite 157 • „CPUTIME (Schwellenwert)“ auf Seite 144 • „SQLROWSREAD (Schwellenwert)“ auf Seite 148 • „SQLTEMPSPACE (Schwellenwert)“ auf Seite 151 	<ul style="list-style-type: none"> • „CONCURRENTWORKLOADACTIVITIES (Schwellenwert)“ auf Seite 155
Anweisung	<ul style="list-style-type: none"> • „ACTIVITYTOTALTIME (Schwellenwert)“ auf Seite 143 • „ESTIMATEDSQLCOST (Schwellenwert)“ auf Seite 147 • „SQLROWSRETURNED (Schwellenwert)“ auf Seite 150 	<ul style="list-style-type: none"> • „CPUTIME (Schwellenwert)“ auf Seite 144 • „SQLROWSREAD (Schwellenwert)“ auf Seite 148 • „SQLTEMPSPACE (Schwellenwert)“ auf Seite 151 	Nicht zutreffend

Bewertungsreihenfolge für Schwellenwerte

Schwellenwerte werden in einer bestimmten Reihenfolge bewertet, wenn sie in einer Datenbank definiert werden.

Die folgenden Schwellenwerte werden vor allen anderen bewertet:

- **TOTALMEMBERCONNECTIONS**. Dieser Schwellenwert wird bewertet, wenn eine neue Verbindung zur Datenbank hergestellt wird.
- **CONCURRENTWORKLOADOCCURRENCES**. Dieser Schwellenwert wird bewertet, wenn ein neues Workloadvorkommen für eine Workloaddefinition gestartet wird, für die dieser Schwellenwert gilt.
- **TOTALSCMEMBERCONNECTIONS**. Dieser Schwellenwert wird bewertet, wenn eine Verbindung einer Serviceklasse beitrifft (entweder eine neue Verbindung oder eine Übertragung zwischen Serviceklassen als Ergebnis einer erneuten Workloadzuordnung).

Alle anderen Schwellenwerte basieren auf erkannten Aktivitäten aus einer SQL-Anweisung oder aus der Ausführung eines Dienstprogramms, wie z. B. des Dienstprogramms **LOAD**, und werden in der folgenden Reihenfolge bewertet:

1. „Prädiktive Schwellenwerte“
2. „Reaktive Schwellenwerte“ auf Seite 136

Prädiktive Schwellenwerte

Prädiktive Schwellenwerte werden vor reaktiven Schwellenwerten geprüft, da sie sich darauf auswirken, ob die Ausführung einer Datenbankaktivität gestartet werden kann.

Prädiktive Schwellenwerte werden in der unten aufgeführten Reihenfolge bewertet. Wenn Sie keinen bestimmten Schwellenwert definieren, wird dieser Schritt übersprungen. Außerdem können die beschriebenen Schritte aus Leistungsgründen während der Ausführung kombiniert werden.

1. Es wird geprüft, ob ein Schwellenwert **CONCURRENTWORKLOADACTIVITIES** vorhanden ist, und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt. Fahren Sie gegebenenfalls mit dem nächsten Schritt fort.
2. Es wird geprüft, ob ein Schwellenwert **ESTIMATEDSQLCOST** vorhanden ist, und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Wenn Sie diesen Schwellenwert in mehreren Domänen definieren, wird der Schwellenwert gemäß den Regeln für die Bereichsauflösung aufgelöst. (Weitere Informationen finden Sie in „Domänenvorrang für Aktivitätsschwellenwerte“ auf Seite 142.) Das Ergebnis dieser Operation ist ein Wert von **ESTIMATEDSQLCOST**, der auf die Aktivität anwendbar ist. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt.
3. Es wird geprüft, ob ein Schwellenwert **CONCURRENTDBCOORDACTIVITIES** für den Arbeitsaktionssettschwellenwert der Datenbankdomäne vorhanden ist, und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt.
4. Es wird geprüft, ob ein Schwellenwert **CONCURRENTDBCOORDACTIVITIES** für den Arbeitsaktionssettschwellenwert der Datenbankdomäne vorhanden ist,

und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt.

5. Es wird geprüft, ob ein Schwellenwert `CONCURRENTDBCOORDACTIVITIES` für den Serviceunterklassenschwellenwert der Domäne vorhanden ist, und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt.
6. Es wird geprüft, ob ein Schwellenwert `CONCURRENTDBCOORDACTIVITIES` für den Servicesuperklassenschwellenwert der Domäne vorhanden ist, und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt.
7. Es wird geprüft, ob ein Schwellenwert `CONCURRENTDBCOORDACTIVITIES` für den Datenbankschwellenwertdomäne vorhanden ist, und wenn dies der Fall ist, ob gegen diesen Schwellenwert verstoßen wurde. Bei einem Verstoß gegen den Schwellenwert wird die entsprechende Aktion ausgeführt.

Aspekte von Schwellenwerten für gemeinsamen Zugriff: Die Bewertungsreihenfolge für Schwellenwerte für gemeinsamen Zugriff folgt nicht der Hierarchie für die Auflösung von Aktivitätsschwellenwerten. Eine Aktivität muss jeden definierten Schwellenwert für gemeinsamen Zugriff passieren, bevor ihre Ausführung erlaubt wird.

Bei Schwellenwerten für gemeinsamen Zugriff werden zuerst Schwellenwerte für Arbeitsaktionssets auf Wordloadebene und danach Arbeitsaktionssets auf Datenbankebene geprüft. Schwellenwerte für Arbeitsaktionssets werden zuerst geprüft, um zu vermeiden, dass Schwellenwerte für Arbeitsaktionssets bei bestimmten Arten von Arbeit andere Arten blockieren, was sich auf den gemeinsamen Zugriff auswirken würde. Indem z. B. zuerst Schwellenwerte für gemeinsamen Zugriff von Arbeitsaktionssets auf Datenbankebene geprüft werden, wird die folgende Situation vermieden.

Nehmen wir an, dass die folgenden Schwellenwerte definiert sind:

- Ein Arbeitsaktionsset für Datenbanken mit einem Arbeitsaktionsschwellenwert für gemeinsamen Zugriff für LOAD-Aktivitäten ist mit dem Wert 1 definiert.
- Der Grenzwert für gemeinsamen Zugriff der Servicesuperklasse S1 ist auf 10 gesetzt.

Nehmen wir weiter an, dass eine LOAD-Aktivität bereits in der Datenbank (unter einer beliebigen Servicesuperklasse) ausgeführt wird und dass bereits neun Aktivitäten in der Servicesuperklasse S1 ausgeführt werden. Eine zweite neue LOAD-Aktivität startet als 10. Aktivität. Würde bei der Schwellenwertbewertung die Hierarchie für die Bereichsauflösung des Aktivitätsschwellenwerts verwendet, würde die eingehende LOAD-Aktivität nicht gegen den Serviceklassenschwellenwert verstoßen und den gemeinsamen Zugriff auf 10 erhöhen. Die LOAD-Aktivität wird dann mit dem Grenzwert für gemeinsamen Zugriff des Arbeitsaktionsschwellenwerts auf Datenbankebene bewertet. Gegen diesen wird verstoßen, da bereits eine LOAD-Aktivität in der Datenbank ausgeführt wird und der Wert für gemeinsamen Zugriff für den Arbeitsaktionsschwellenwert nur 1 beträgt. Die zweite LOAD-Aktivität wird nun in die Warteschlange gestellt.

Alle neu in der Servicesuperklasse S1 eintreffenden Aktivitäten werden nun in die Warteschlange gestellt (da der Serviceklassengrenzwert für gemeinsamen Zugriff bereits erreicht ist). Die Schwellenwertwarteschlange der Arbeitsaktion wirkt sich

auf die Serviceklasse aus. Dies ist nicht erwünscht, da Aktivitäten, die versuchen, in der Serviceklasse ausgeführt zu werden, nicht notwendigerweise eine Beziehung zur Schwellenwertbedingung der Arbeitsaktion haben (beispielsweise sollte eine Einfügeoperation, die versucht, in der Servicesuperklasse S1 ausgeführt zu werden, nicht auf eine LOAD-Aktivität warten müssen, die aufgrund einer Arbeitsaktions-Schwellenwertbedingung auf Datenbankebene in die Warteschlange gestellt wurde). Zur Vermeidung solcher Situationen wird der Arbeitsaktionsschwellenwert für gemeinsamen Zugriff auf Datenbankebene daher vor Serviceunterklasse-, Service-superklasse- und Datenbankschwellenwerten für gemeinsamen Zugriff geprüft. Da der Schwellenwert für gemeinsamen Zugriff von Arbeitsaktionssets auf Datenbankebene zuerst geprüft wird, wird die 10. Aktivität in der Serviceklasse (bei der es sich zufällig um eine LOAD-Aktivität handelt) auf der Ebene des Arbeitsaktionsschwellenwerts für Datenbanken blockiert, bevor sie versuchen kann, einen Platz in der Servicesuperklasse S1 einzunehmen.

Reaktive Schwellenwerte

Reaktive Schwellenwerte werden in diskreter Weise beim Ausführen einer Aktivität bewertet. Für die Bewertung reaktiver Schwellenwerte wird keine bestimmte Reihenfolge verwendet. Die folgenden reaktiven Schwellenwerte sind verfügbar:

- ACTIVITYTOTALTIME
- AGGSQLTEMPSPACE
- CONNECTIONIDLETIME
- CPUTIME
- CPUTIMEINSC
- DATATAGINSC
- SQLTEMPSPACE
- SQLROWSREAD
- SQLROWSREADINSC
- SQLROWSRETURNED
- UOWTOTALTIME

Erstellen eines Schwellenwerts

Mit der DDL-Anweisung `CREATE THRESHOLD` (oder der Anweisung `CREATE WORK ACTION SET`) können Sie Schwellenwerte erstellen. Mit einem Schwellenwert legen Sie einen Grenzwert für die Ressourcenauslastung fest.

Vorbereitende Schritte

Zum Erstellen eines Schwellenwerts benötigen Sie die Berechtigung `WLMADM` oder `DBADM`.

Weitere Informationen zu Voraussetzungen finden Sie in den folgenden Abschnitten:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Namenskonventionen

Verwenden Sie die Anweisung `CREATE WORK ACTION SET` oder die Anweisung `ALTER WORK ACTION SET` mit den Schlüsselwörtern `ADD WORK ACTION`, um einen Schwellenwert für ein Arbeitsaktionsset zu erstellen. Weitere Informationen finden Sie in Anweisung `CREATE WORK ACTION SET` oder Anweisung `ALTER WORK ACTION SET`.

Vorgehensweise

Gehen Sie wie folgt vor, um einen Schwellenwert zu erstellen:

1. Setzen Sie die Anweisung `CREATE THRESHOLD` ab und geben Sie dabei mindestens eine der folgenden Eigenschaften für den Schwellenwert an:
 - Den Namen des Schwellenwerts.
 - Die Schwellenwertdomäne. Die Schwellenwertdomäne ist das Datenbankobjekt, dem der Schwellenwert zugeordnet ist und auf das er angewendet wird. Welche Domäne angewendet wird, hängt von der Art des Schwellenwerts ab. Weitere Informationen finden Sie in „Schwellenwertdomäne und Umsetzungsbereich“ auf Seite 132.
 - Den Umsetzungsbereich für den Schwellenwert. Der Schwellenwertbereich ist der Umsetzungsbereich des Schwellenwerts in seiner Domäne. Welcher Umsetzungsbereich angewendet wird, hängt von der Art des Schwellenwerts ab. Weitere Informationen finden Sie in „Schwellenwertdomäne und Umsetzungsbereich“ auf Seite 132.
 - Optional: Inaktivieren Sie den Schwellenwert, wenn er erstellt wird. Standardmäßig wird ein Schwellenwert als aktiviert erstellt. Wenn Sie den Schwellenwert als inaktiviert erstellen und ihn später aktivieren wollen, verwenden Sie die Anweisung `ALTER THRESHOLD`.
 - Das Schwellenwertvergleichselement, um den Typ von Schwellenwert und den zulässigen Maximalwert anzugeben. Wenn der Maximalwert erreicht wird, wird die für den Schwellenwert angegebene Aktion erzwungen. Weitere Informationen zu den verfügbaren Schwellenwerten finden Sie in „Verbindungsschwellenwerte“ auf Seite 141, „Aktivitätsschwellenwerte“ auf Seite 142, „Zusammengefasste Schwellenwerte“ auf Seite 152 und „UOW-Schwellenwerte“ auf Seite 160.
 - Die Aktionen, die auszuführen sind, wenn der Maximalwert für den Schwellenwert überschritten wird. Die Aktionen bestehen aus einer obligatorischen Aktion, die sich auf die Ausführung der Aktivität auswirkt (`STOP EXECUTION`, `CONTINUE`, `FORCE APPLICATION` oder `REMAP ACTIVITY TO`) und einer optionalen Aktivitätserfassungsaktion (`COLLECT ACTIVITY DATA`). Die Optionen, die Sie für die Aktivitätserfassungsaktion angeben, bestimmen, welche Informationen für die Aktivität erfasst werden sollen, die den Verstoß gegen die Schwellenwertbegrenzung verursachte.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird der Schwellenwert zur Sicht `SYSCAT.THRESHOLDS` hinzugefügt.

Ändern eines Schwellenwerts

Mit der Anweisung `ALTER THRESHOLD` können Sie Schwellenwerte ändern. Sie können einen Schwellenwert z. B. ändern, um die Begrenzung für eine bestimmte Ressource zu ändern.

Vorbereitende Schritte

Zum Ändern eines Schwellenwerts benötigen Sie die Berechtigung `SQLADM`, `WLMADM` oder `DBADM`. Soll eine andere Klausel als die Klausel `COLLECT` angegeben werden, muss die Berechtigungs-ID die Berechtigung `WLMADM` oder `DBADM` aufweisen.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Verwenden Sie die Anweisung ALTER WORK ACTION SET mit dem Schlüsselwort ADD WORK ACTION, um einen Schwellenwert für ein Arbeitsaktionsset zu ändern.

Einschränkungen

Mit der Anweisung ALTER THRESHOLD können Sie nicht den Schwellenwerttyp ändern. Sie können z. B. einen Schwellenwert TOTALMEMBERCONNECTIONS nicht in einen Schwellenwert TOTALSCMEMBERCONNECTIONS ändern. Wenn Sie einen anderen Schwellenwerttyp benötigen, löschen Sie die vorhandenen Schwellenwerte und erstellen Sie dann einen neuen Schwellenwert.

Vorgehensweise

Gehen Sie wie folgt vor, um einen Schwellenwert zu ändern:

1. Geben Sie mindestens eines der folgenden Eigenschaften für den Schwellenwert in der Anweisung ALTER THRESHOLD an. Sie können die folgenden Eigenschaften ändern:
 - Die Begrenzung für das Schwellenwertvergleichselement.
 - Die Aktionen, die bei einem Verstoß gegen die Schwellenwertbegrenzung auszuführen sind.
 - Eine Angabe, ob der Schwellenwert aktiviert ist.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird der Schwellenwert in der Sicht SYSCAT.THRESHOLDS aktualisiert.

Löschen eines Schwellenwerts

Sie können einen Schwellenwert, den Sie nicht mehr benötigen, mit der DDL-Anweisung DROP THRESHOLD löschen.

Vorbereitende Schritte

Zum Löschen eines Schwellenwerts benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Wenn Sie einen Schwellenwert in einem Arbeitsaktionsset löschen wollen, verwenden Sie die Anweisung ALTER WORK ACTION SET. Sie können einen Schwellenwert auch löschen, indem Sie das gesamte Arbeitsaktionsset mit der Anweisung DROP löschen.

Vorgehensweise

Gehen Sie wie folgt vor, um einen Schwellenwert zu löschen:

1. Führen Sie einen der folgenden Schritte aus:
 - Wenn es sich bei dem Schwellenwert um einen Warteschlangenschwellenwert handelt, inaktivieren Sie ihn mit der ALTER THRESHOLD.
 - Wenn Sie einen Warteschlangenschwellenwert mit einer Anweisung ALTER THRESHOLD inaktiviert haben, führen Sie ein COMMIT durch, um die Änderung festzuschreiben.
2. Löschen Sie den Schwellenwert mit der Anweisung DROP THRESHOLD.

3. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird der Schwellenwert aus der Sicht SYSCAT.THRESHOLDS entfernt.

Beispiel: Verwenden von Schwellenwerten

Sie können Schwellenwerte für eine Vielzahl von Zwecken verwenden. In diesem Szenario werden Schwellenwerte verwendet, um die Anzahl ausgeführter großer Jobs zu steuern, sodass unterschiedliche Ausführungszeiten für unterschiedliche Anwendungen möglich sind und das Verhalten einer in Entwicklung befindlichen Anwendung gesteuert werden kann.

Sie können mit einer DB2-Workload-Management-Lösung die Datenbankressourcen für die verschiedenen Abteilungen in einem Unternehmen aufteilen und verwalten. Nehmen wir z. B. an, dass die Vertriebsabteilung zwei Hauptberichte führt, die aus den monatlichen und den jährlichen Verkäufen bestehen. Nehmen wir weiter an, dass die Personalabteilung alle zwei Wochen eine Anwendung der Lohnbuchhaltung ausführt und dass das Entwicklungsteam auf Wunsch des Managements an einem neuen Typ von Bericht arbeitet. Erstellen Sie Serviceklassen, um unterschiedliche Ausführungsumgebungen für diese Abteilungen zu definieren:

```
CREATE SERVICE CLASS SALES
CREATE SERVICE CLASS HUMANRESOURCES
CREATE SERVICE CLASS DEVELOPMENT
```

In dieser Situation erstellen Sie für jede dieser Anwendungen eine Workloaddefinition, um die jeweilige Anwendung der zutreffenden Servicesuperklasse zuzuordnen:

```
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
```

Der Datenbankkatalog enthält daher die folgenden Workloaddefinitionen:

- MonthlySales, der Servicesuperklasse Sales zugeordnet
- YearlySales, der Servicesuperklasse Sales zugeordnet
- Payroll, der Servicesuperklasse Human Resources zugeordnet
- NewReport, der Servicesuperklasse Development zugeordnet

Schwellenwert für die Anzahl großer Jobs

Da der Bericht YearlySales sehr groß ist, soll nicht mehr als ein Vorkommen dieser Anwendung auf einmal in der Datenbank ausgeführt werden. Sie erstellen daher einen Schwellenwert, um die maximale Anzahl gleichzeitiger Vorkommen dieser Workload auf 1 zu setzen:

```
CREATE THRESHOLD SINGLEYEARLYSALESRPT FOR WORKLOAD YEARLYSALES ACTIVITIES
ENFORCEMENT MEMBER
WHEN CONCURRENTWORKLOADOCCURRENCES > 1
STOP EXECUTION
```

Sie können eine ähnliche Lösung erzielen, indem Sie die Anwendung YearlySales mit einer Serviceunterklasse YearlySalesReports (unter der Servicesuperklasse Sales) verknüpfen und den Schwellenwert für maximalen gemeinsamen Zugriff für die Serviceunterklasse auf 1 setzen:

```
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
```

In beiden Fällen können Sie die Schwellenwertaktion auf STOP EXECUTION setzen, um zu verhindern, dass mehr als ein Vorkommen der Workload ausgeführt wird. Sie können auch Aktivitätsinformationen erfassen, wenn Sie weitere Informationen zu den Bedingungen erfassen wollen, unter denen ein Schwellenwertverstoß auftritt.

Schwellenwert für die Aktivitätslaufzeiten

Da erwartet wird, dass alle Anwendungen in maximal einer Stunde abgeschlossen sind, erstellen Sie einen Schwellenwert mit einer Datenbankdomäne, der verhindert, dass eine Aktivität länger als 1 Stunde ausgeführt wird. Die einzige Ausnahme dieser Regel ist der Jahresbericht, dessen Ausführung bis zu 5 Stunden dauern kann. +Sie können also der Workload YearlySales einen Schwellenwert für die Gesamtzeit der Aktivitäten von 5 Stunden zuordnen. Dadurch wird der Schwellenwert für die Gesamtzeit der Aktivitäten, der auf den Jahresverkaufsbericht angewendet wird, außer Kraft gesetzt, wodurch die Zeitvorgaben gelockert werden. Der neue Wert von 5 Stunden gilt jetzt für die Workload YearlySales. Für alle anderen Elemente in der Datenbank gilt jedoch weiterhin der globale Wert von 1 Stunde:

```
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
```

Schwellenwert für die Anzahl der Koordinator- und verschachtelten Aktivitäten

Die Anwendung NewReport nutzt sehr viele gespeicherte Prozeduren und benutzerdefinierte Funktionen und ist noch nicht vollständig getestet. Sie erzeugt daher große Mengen von Aktivitäten, die sich auf den Rest des Systems auswirken. Nach Beratung mit dem Entwickler erfahren Sie, dass dieser neue Bericht nicht mehr als 20 Aktivitäten insgesamt generieren sollte. Sie definieren daher einen Schwellenwert für Workloadaktivitäten in der Workload NewReport und setzen diesen auf 20. Anfangs legen Sie als Schwellenwertaktion STOP EXECUTION und COLLECT ALL fest, um unerwünschte Nebeneffekte durch das Starten sehr vieler Aktivitäten auszuschließen und dem Entwickler bei der Problemerkennung zu helfen:

```
CREATE THRESHOLD MAXDEVACTIVITIES FOR SERVICE CLASS DEVELOPMENT ACTIVITIES
  ENFORCEMENT DATABASE
  WHEN CONCURRENTDBCOORDACTIVITIES > 20
  COLLECT ACTIVITY DATA WITH DETAILS AND VALUES
  STOP EXECUTION
```

Wenn die Anwendung stabiler wird, kommt sie in die Optimierungsphase. In dieser Phase versucht der Entwickler, die Anzahl der von der Anwendung generierten Aktivitäten von 15 bis 20 auf 15 zu verringern. Jetzt ändern Sie den Schwellenwert, indem Sie die obere Begrenzung auf 15 und die Schwellenwertaktion auf CONTINUE setzen. Diese Schwellenwertdefinition hilft bei der Ermittlung und Behebung

von Situationen, in denen die Anzahl der generierten Aktivitäten 15 überschreitet, die höhere Stabilität der Anwendung aber nicht das Stoppen der Ausführung erforderlich macht.

```
ALTER THRESHOLD MAXDEVACTIVITIES
  WHEN CONCURRENTDBCOORDACTIVITIES > 15
  COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS AND VALUES
  CONTINUE
```

Schwellenwert zur Begrenzung der UOW-Ausführungsdauer

Die Anwendung LongUOW setzt Transaktionen ab, die gelegentlich länger als die gewünschten zehn Minuten ausgeführt werden. Dies führt zu Sperren, die zu lange aktiviert sind, und verhindert die Ausführung wichtigerer Anwendungen. In diesem Fall wollen Sie das Ende der Anwendung erzwingen, anstatt sie andere Arbeit aufhalten zu lassen. Sie können die Laufzeit für die Transaktionen dieser Anwendung mit dem Schwellenwert UOWTOTALTIME auf einen vom Administrator definierten Zeitraum beschränken.

Erstellen Sie zuerst eine Workload für die Anwendung LongUOW:

```
CREATE WORKLOAD LONG_UOW APPLNAME('LONGUOW') SERVICE CLASS SYSDEFAULTUSERCLASS
```

Erstellen Sie dann einen Schwellenwert für diese Workload, der das Ende der Anwendung LongUOW erzwingt, wenn die Beendigung der Anwendungstransaktionen 10 Minuten überschreitet:

```
CREATE THRESHOLD FORCELONGUOW FOR WORKLOAD LONG_UOW ACTIVITIES ENFORCEMENT DATABASE
  WHEN UOWTOTALTIME > 10 MINUTES FORCE APPLICATION
```

Sie können diesen Schwellenwert auf der Ebene der Serviceunterklasse oder auf der Datenbankebene anwenden.

Verbindungsschwellenwerte

Ein Verbindungsschwellenwert wendet Steuerelemente auf einzelne Datenbankverbindungen an. Sie können Verbindungsschwellenwerte verwenden, um die Gesamtzahl gleichzeitiger Verbindungen zur Datenbank und die Inaktivitätszeit einer Verbindung zu beschränken.

CONNECTIONIDLETIME (Schwellenwert)

Der Schwellenwert CONNECTIONIDLETIME gibt die Höchstdauer an, während der eine Verbindung inaktiv sein kann (d. h. nicht an einer Benutzeranforderung arbeitet).

Typ Verbindung

Definitionsdomäne

Datenbank oder Servicesuperklasse

Umsetzungsbereich

Datenbank

Überwachte Arbeit

Benutzerverbindungen

Warteschlangensteuerung

Nein

Einheit

Dauer in Minuten, Stunden oder Tagen

Prädiktiv oder reaktiv

Reaktiv

Wenn eine Verbindung länger als während der vom Schwellenwert definierten Dauer inaktiv bleibt und die Schwellenwertaktion STOP EXECUTION ist, wird die Verbindung geschlossen.

Aktivitätsschwellenwerte

Ein Aktivitätsschwellenwert bezieht sich auf eine einzelne Aktivität. Wenn die Ressourcenauslastung einer einzelnen Aktivität gegen die Obergrenze des Schwellenwerts verstößt, der diese Aktivität verfolgt, wird die entsprechende Aktion ausgelöst und einmal auf die Aktivität angewendet.

Nachdem der Schwellenwert einmal angewendet wurde, wird er für die Aktivität inaktiviert und nicht erneut angewendet.

Beispiel: Nehmen wir an, dass Sie einen zeitbasierten Schwellenwert definiert haben, der nach 5 Minuten eine Aktion CONTINUE auslöst. Wenn eine Aktivität gegen diesen Schwellenwert verstößt, wird die Aktion nur einmal und nicht alle 5 Minuten angewendet.

Domänenvorrang für Aktivitätsschwellenwerte

Aktivitätsschwellenwerte gelten für einzelne Aktivitäten. Wenn sich mehrere Schwellenwerte auf dieselbe ausgeführte Aktivität beziehen, muss eine Entscheidung getroffen werden, welcher Schwellenwert erzwungen werden soll.

Zusammengefasste Schwellenwerte sind nicht betroffen, da dieselbe Aktivität gleichzeitig zu mehreren Aktivitätszusammenfassungen beitragen kann, wie es z. B. bei Schwellenwerten für gemeinsamen Zugriff zutrifft.

Die Entscheidung, welcher Aktivitätsschwellenwert auf eine ausgeführte Aktivität angewendet werden soll, folgt der Regel, dass ein in einer lokalen Domäne definierter Wert alle Werte aus einer breiter gefassten oder globaleren Domäne außer Kraft setzt. Nachfolgend ist eine Hierarchie von Domänen von lokal bis global aufgeführt:

1. Anweisung
2. Arbeitsaktion (auf Workloadebene)
3. Workload
4. Serviceunterklasse
5. Service-
superklasse
6. Arbeitsaktion (auf Datenbankebene)
7. Datenbank

Das folgende Beispiel zeigt, wie Schwellenwerte überschrieben werden:

Beispiele

Das folgende Beispiel zeigt, wie Schwellenwerte überschrieben werden:

- Ein Schwellenwert, der eine maximale Ausführungszeit von 1 Stunde für alle in der Datenbankdomäne definierten Datenbankabfragen definiert, wird von einem Schwellenwert überschrieben, der eine maximale Ausführungszeit von 5 Stunden für eine Servicesuperklasse definiert, die für die Handhabung großer Abfragen eingerichtet ist.

- Derselbe Schwellenwert für die Servicesuperklasse wird von einem Schwellenwert für sehr große Abfragen überschrieben, der 10 Stunden als maximale Ausführungszeit für eine Serviceunterklasse definiert.
- Die in der Datenbankdomäne definierte maximale Ausführungszeit von 1 Stunde kann von einem Wert von 10 Minuten in einer zweiten Servicesuperklasse überschrieben werden, wodurch sichergestellt werden soll, dass kürzere wichtige Abfragen schneller ausgeführt werden können.
- Die Ausführung einer Anweisung mit Text, der mit dem Text übereinstimmt, der für einen Schwellenwert in der Anweisungsdomäne angegeben ist, führt zu einem Schwellenwertverstoß, der alle anderen Schwellenwerte überschreitet.

ACTIVITYTOTALTIME (Schwellenwert)

Der Schwellenwert ACTIVITYTOTALTIME gibt die Zeitdauer an, die der Datenserver höchstens für die Verarbeitung einer Aktivität aufwenden kann.

Typ Aktivität

Definitionsdomäne

Datenbank, Servicesuperklasse, Serviceunterklasse, Arbeitsaktion, Workload und Anweisung

Umsetzungsbereich

Datenbank

Überwachte Arbeit

Erkannte Koordinator- und verschachtelte Aktivitäten (siehe „Aktivitäten“ auf Seite 15)

Warteschlangensteuerung

Nein

Einheit

In Sekunden, Minuten, Stunden oder Tagen ausgedrückte Zeitdauer. Wird die Zeiteinheit in Sekunden angegeben, muss dieser Wert ein Vielfaches von 10 sein.

Prädiktiv oder reaktiv

Reaktiv

Diese Zeitdauer stellt die gesamte Lebensdauer der Aktivität dar. Hierzu gehören die Zeit, in der auf Workload-Management-Warteschlangen gewartet wurde, sowie andere Wartezeiten während der Ausführung der Aktivität, beispielsweise Wartezeiten auf Sperren. Wenn ein Cursor geöffnet wird, dauert die zugehörige Aktivität so lange, bis der Cursor geschlossen wird. Zu den von diesem Schwellenwert abgedeckten Aktivitäten gehören die Ausführung von SQL-Anweisungen ohne Kompilierzeit sowie das Dienstprogramm LOAD.

Wenn ein Zeitschwellenwert auf eine gespeicherte Prozedur angewendet wird, bezieht er sich auch auf Arbeit, die innerhalb der gespeicherten Prozedur ausgeführt wird. Wenn daher der Zeitschwellenwert für eine gespeicherte Prozedur abläuft, wird auch alle Arbeit in dieser gespeicherten Prozedur gestoppt. Hierarchien von Aufrufen gespeicherter Prozeduren können zu Hierarchien von Zeitschwellenwerten führen, die auf Aktivitäten angewendet werden, die in der untersten Verschachtelungsebene ausgeführt werden. Es wird immer der restriktivste Zeitschwellenwert in der Hierarchie (d. h. der Zeitschwellenwert mit dem kürzesten Endtermin) angewendet.

Der Datenserver fasst IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors als Benutzerlogik auf. Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten Schwellenwerte.

CPUTIME (Schwellenwert)

Der Schwellenwert CPUTIME gibt den maximalen Anteil kombinierter Benutzer- und Systemprozessorzeit an, die eine Aktivität in einem bestimmten Member während ihrer Ausführung verwenden kann. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die Prozessorressourcen exzessiv verwenden.

Typ Aktivität

Definitionsdomäne

Datenbank, Servicesuperklasse, Serviceunterklasse, Arbeitsaktion, Workload und Anweisung

Umsetzungsbereich

Member

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Zeit

Prädiktiv oder reaktiv

Reaktiv

Der Anteil der Prozessorzeit, die eine Aktivität mit der Ausführung verbringt, wird gemessen von der Zeit, zu der die Ausführung der Aktivität in dem Member beginnt, nachdem sie eventuell durch Schwellenwerte in die Warteschlange gestellt wurde, bis zu der Zeit, zu der die Ausführung der Aktivität beendet ist.

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- Alle DML-Aktivitäten.
- CALL-Aktivitäten. Die Prozessorzeit für eine CALL-Aktivität umfasst nicht die Prozessorzeit untergeordneter Aktivitäten. Die Prozessorzeit, die in abgeschirmten Prozessen verbracht wird, wird ebenfalls nicht zur Gesamtprozessorzeit für die CALL-Aktivität gezählt.

Aktivitäten, die vom Datenbankmanager über ein Dienstprogramm oder eine Prozedur mit Ausnahme der Prozedur ADMIN_CMD ausgelöst wurden, werden bei dieser Bedingung nicht berücksichtigt. Der Datenserver fasst IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors als Benutzerlogik auf. Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten Schwellenwerte. Untergeordnete Aktivitäten des Befehls LOAD werden von diesen Schwellenwert nicht überwacht.

Beispiel

Das folgende Beispiel erstellt einen Schwellenwert TH1 vom Typ CPUTIME für die Datenbankdomäne mit einem Member als Umsetzungsbereich. Dieser Schwellenwert stoppt jede Aktivität, deren Ausführung länger als 30 Sekunden dauert. Dies wird in einem Intervall von 5 Sekunden überprüft. Sie können mithilfe dieses

Schwellenwerts sicherstellen, dass keine Abfragen auf dem System einen unverhältnismäßig hohen Anteil der Prozessorzeit verwenden, was sich nachteilig auf die andere Arbeit auf dem System auswirken könnte.

```
CREATE THRESHOLD TH1 FOR DATABASE ACTIVITIES
  ENFORCEMENT MEMBER
  WHEN CPUTIME > 30 SECONDS CHECKING EVERY 5 SECONDS
  STOP EXECUTION;
```

CPUTIMEINSC (Schwellenwert)

Der serviceklasseninterne Schwellenwert CPUTIMEINSC gibt den maximalen Anteil kombinierter Benutzer- und Systemprozessorzeit an, die eine Aktivität in einem bestimmten Member während ihrer Ausführung in einer bestimmten Serviceunterklasse verwenden kann. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die Prozessorressourcen exzessiv verwenden.

Klasse Aktivität

Definitionsdomäne

Serviceunterklasse

Umsetzungsbereich

Member

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Zeit

Prädiktiv oder reaktiv

Reaktiv

Die Prozessorzeit, die eine Aktivität mit der Ausführung verbringt, wird gemessen von der Zeit, zu der die Aktivität in die aktuelle Serviceunterklasse eintritt, bis zu der Zeit, zu der die Aktivität die Serviceunterklasse verlässt oder ihre Ausführung abgeschlossen ist.

Dieser Schwellenwert unterscheidet sich vom Schwellenwert CPUTIME darin, dass er nur den Anteil der Prozessorzeit steuert, der in einer bestimmten Serviceunterklasse verwendet werden kann, und nicht den gesamten Anteil der Prozessorzeit, der während der Laufzeit der Aktivität verwendet wird.

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- Alle DML-Aktivitäten.
- CALL-Aktivitäten. Die Prozessorzeit für eine CALL-Aktivität umfasst nicht die Prozessorzeit untergeordneter Aktivitäten. Die Prozessorzeit, die in abgeschirmten Prozessen verbraucht wird, wird ebenfalls nicht zur Gesamtprozessorzeit für die CALL-Aktivität gezählt.

Aktivitäten, die vom Datenbankmanager über ein Dienstprogramm oder eine Prozedur mit Ausnahme der Prozedur ADMIN_CMD ausgelöst wurden, werden bei dieser Bedingung nicht berücksichtigt. Der Datenserver fasst IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors als Benutzerlogik auf. Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors

aufgerufen werden, gelten Schwellenwerte. Untergeordnete Aktivitäten des Befehls LOAD werden von diesen Schwellenwert nicht überwacht.

Mithilfe der Aktion REMAP ACTIVITY können Sie Aktivitäten steuern, indem Sie sie einer Serviceunterklasse mit anderen Ressourcenzuordnungen zuordnen.

Beispiel

Das folgende Beispiel erstellt zwei Serviceunterklassen, A1 und A2, unter einer Superklasse A, mit einem einzigen serviceklasseninternen Schwellenwert CPUTIMEINSC, der Aktivitäten zwischen den Unterklassen neu zuordnet, nachdem 1 Minute Prozessorzeit während der Abfragebewertung in der Serviceunterklasse A1 verbraucht wurde. Ein Ereignismonitordatensatz wird protokolliert.

```
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;
```

DATATAGINSC (Schwellenwert)

Der serviceklasseninterne Schwellenwert DATATAGINSC prüft den Datentagwert eines Tabellenbereichs oder einer Speichergruppe, auf den bzw. die eine Aktivität zugreift. Mit diesem Schwellenwert können Sie einer anderen DB2-Serviceunterklasse auf der Basis des Datenzugriffs eine Aktivität dynamisch zuordnen.

Typ Aktivität

Definitionsdomäne
Serviceunterklasse

Umsetzungsbereich
Member

Überwachte Arbeit
Anerkannter Koordinator

Warteschlangensteuerung
Nein

Einheit
Datentag

Prädiktiv oder reaktiv
Reaktiv

Daten können beim Erstellen oder Ändern eines Tabellenbereichs oder einer Speichergruppe durch Angabe des Attributs DATA TAG mit einem numerischen Wert gekennzeichnet werden. Schwellenwerte von Datatags werden bewertet, wenn ein Suchlauf für eine Tabelle zum ersten Mal geöffnet wird und wenn Daten in eine Tabelle eingefügt werden. Datentagschwellenwerte, die von der Aktivität nach dem Öffnen eines Suchlaufs als Ergebnis einer Operation für erneute Zuordnung festgestellt werden, werden auf diesen Suchlauf nicht angewendet.

Lediglich die Datentags für die Datentabellenbereiche, auf die in einer Abfrage zugegriffen wird, werden vom Schwellenwert berücksichtigt. Die Datentagwerte für Tabellenbereiche für Indizes oder für lange Objektdaten werden während einer Schwellenwertbewertung nicht berücksichtigt. Wenn z. B. für die Abfrage SELECT COUNT(*) FROM T1 reiner Indexzugriff besteht, der Index sich in einem Tabellenbe-

reich mit dem Datentag 3 befindet und sich die Tabellendaten in einem Tabellenbereich mit dem Datentag 1 befinden, verwendet die Schwellenwertbewertung bei der Abfrageausführung den Datentag 1 (Datentag für den Datentabellenbereich). Das Verhalten des Schwellenwerts für den Datentag ist von der im Abfragezugriffsplan ausgewählten Zugriffsmethode (Indexsuchlauf oder direkter Suchlauf) unabhängig.

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- Koordinatoraktivitäten des Typs DML und die zugehörige Arbeit von Unteragenten (Ausführung von Unterabschnitten).
- Verschachtelte DML-Aktivitäten, die von Benutzeranwendungen abgeleitet werden. Daher sind vom Datenbankmanager ausgegebene DML-Aktivitäten wie Dienstprogramme, SYSPROC-Prozeduren (ausgenommen SYSPROC.ADMIN_CMD) oder interne SQL-Anweisungen von diesem Schwellenwert nicht betroffen.

Im folgenden Einsatzszenario erstellt der Benutzer 3 Tabellenbereiche. TBHIGH enthält Daten mit hoher Priorität und hat den Datentag 1. TBMED enthält die Daten mit mittlerer Priorität und hat den Datentag 4. TBLow enthält Daten mit niedriger Priorität und hat den Datentag 9. Die Aktivitäten sind anfänglich einer Serviceunterklasse mit hoher Priorität zugeordnet. Wenn Aktivitäten Daten in einem anderen Tabellenbereich als dem Tabellenbereich TBHIGH verarbeiten, will der Benutzer die Priorität dieser Aktivitäten senken. Im folgenden Beispiel wurde die Servicesuperklasse MAINSC mit den drei Serviceunterklassen SCHIGH, SCMED und SCLOW (für die Ausführung von Arbeit mit hoher, mittlerer und niedriger Priorität) bereits erstellt.

Erstellen Sie den Schwellenwert DATATAGINSC für die Serviceunterklasse mit hoher Priorität und ordnen Sie ihn der Serviceunterklasse mit mittlerer Priorität zu, wenn eine Aktivität Daten mit einem anderen Datentag als 1 verarbeitet.

```
CREATE THRESHOLD MAPTOMED FOR SERVICE CLASS SCHIGH UNDER MAINSC ACTIVITIES  
ENFORCEMENT MEMBER WHEN DATATAGINSC NOT IN (1) REMAP ACTIVITY TO SCMED
```

Erstellen Sie den Schwellenwert SQLDATATAGINSC für die Serviceunterklasse mit mittlerer Priorität und ordnen Sie ihn der Serviceunterklasse mit niedriger Priorität zu, wenn eine Aktivität Daten mit dem Datentag 9 verarbeitet.

```
CREATE THRESHOLD MAPTOLow FOR SERVICE CLASS SCMED UNDER MAINSC ACTIVITIES  
ENFORCEMENT MEMBER WHEN DATATAGINSC IN (9) REMAP ACTIVITY TO SCLOW
```

Wenn eine Aktivität einen Tabellenbereich mit dem Datentag 9 verarbeitet, wird diese Aktivität SCMED erneut zugeordnet und dann sofort wieder SCLOW erneut zugeordnet (aufgrund des Schwellenwerts für die Serviceunterklasse SCMED). Dem Benutzer werden in diesem Fall zwei Schwellenwertverstöße angezeigt.

ESTIMATEDSQLCOST (Schwellenwert)

Der Schwellenwert ESTIMATEDSQLCOST gibt den maximalen geschätzten Aufwand an, der für DML-Aktivitäten zulässig ist.

Typ Aktivität

Definitionsdomäne

Datenbank, Servicesuperklasse, Serviceunterklasse, Arbeitsaktion, Workload und Anweisung

Umsetzungsbereich

Datenbank

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Geschätzter SQL-Aufwand, ausgedrückt in Timeron

Prädiktiv oder reaktiv

Prädiktiv

Dieser Schwellenwert verfolgt die folgenden Aktivitäten:

- DML-Aktivitäten, die in dem Koordinatormember abgesetzt werden.
- Verschachtelte DML-Aktivitäten, die über eine Benutzeranwendung aufgerufen werden. Daher sind DML-Aktivitäten, die vom Datenserver intern abgesetzt werden, (z. B. DML-Aktivitäten, die aus den DB2-Dienstprogrammen, gespeicherten SYSPROC-Prozeduren und internem SQL-Code abgesetzt werden) von diesem Schwellenwert nicht betroffen, sofern ihr Aufwand nicht in der Schätzung für die übergeordnete Aktivität enthalten ist. In dieser Situation werden diese Aktivitäten indirekt überwacht. Ein Trigger ist ein Beispiel für eine indirekt überwachte Aktivität. IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors werden als Benutzerlogik aufgefasst. Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten Schwellenwerte. Informationen zu den Aktivitäten, die in eine Arbeitsklasse mit dem Arbeitstyp DML fallen, finden Sie in „Identifizieren der Arbeit nach Typ, Aufwand oder Datenzugriff mithilfe von Arbeitsklassen“ auf Seite 52.

SQLROWSREAD (Schwellenwert)

Der Schwellenwert SQLROWSREAD gibt die maximale Anzahl von Zeilen an, die eine DML-Aktivität in einem Member lesen kann. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die übermäßig viele Zeilen lesen.

Klasse Aktivität

Definitionsdomäne

Datenbank, Servicesuperklasse, Serviceunterklasse, Arbeitsaktion, Workload und Anweisung

Umsetzungsbereich

Member

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Anzahl der Zeilen

Prädiktiv oder reaktiv

Reaktiv

Dieser Schwellenwert unterscheidet sich vom Schwellenwert SQLROWSRETURNED darin, dass er die maximale Anzahl der Zeilen steuert, die während der Abfragebewertung gelesen werden, und nicht die Anzahl der Zeilen, die vom Datenserver an eine Clientanwendung zurückgegeben werden.

Indexzugriffe werden bei der Gesamtzahl der gelesenen Zeilen nicht berücksichtigt. Wenn ein Zugriffsplan bei der Abfragebewertung nur Indizes verwendet, wird nicht gegen den Schwellenwert SQLROWSREAD verstoßen.

Dieser Schwellenwert wird in vom Benutzer konfigurierbaren Zeitintervallen bewertet. Ist das Intervall größer als der zum Überschreiten der Anzahl gelesener Zeile erforderliche Zeitraum, überschreitet die Anzahl der für eine Aktivität in einem Member gelesener Zeilen möglicherweise den Grenzwert, bevor diese Übertretung festgestellt wird.

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- Koordinatoraktivitäten des Typs DML und die zugehörige Arbeit von Unteragenten wie die Ausführung von Unterabschnitten.
- Verschachtelte DML-Aktivitäten, die von Benutzeranwendungen abgeleitet werden. Daher sind DML-Aktivitäten, die von DB2-Logik, wie Dienstprogrammen, SYSPROC-Prozeduren oder internem SQL-Code, abgesetzt werden, von diesem Schwellenwert nicht betroffen. IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors werden als Benutzerlogik aufgefasst; Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten daher Schwellenwerte.

Beispiel

Das folgende Beispiel erstellt einen Schwellenwert TH1 vom Typ SQLROWSREAD für die Datenbankdomäne mit einem Member als Umsetzungsbereich. Dieser Schwellenwert stoppt die Ausführung jeder Aktivität, die mehr als 5.000.000 Zeilen während der Abfragebewertung liest, was in einem Intervall von 10 Sekunden überprüft wird. Sie können mithilfe dieses Schwellenwerts sicherstellen, dass keine Abfragen auf dem System eine unverhältnismäßig große Anzahl Zeilen lesen, was sich nachteilig auf die andere Arbeit auf dem System auswirken könnte.

```
CREATE THRESHOLD TH1 FOR DATABASE ACTIVITIES
  ENFORCEMENT MEMBER
  WHEN SQLROWSREAD > 5000000 CHECKING EVERY 10 SECONDS
  STOP EXECUTION;
```

SQLROWSREADINSC (Schwellenwert)

Der serviceklasseninterne Schwellenwert SQLROWSREADINSC gibt die maximale Anzahl von Zeilen an, die eine DML-Aktivität in einem bestimmten Member während der Ausführung in einer bestimmten Serviceunterklasse lesen kann. Über diesen Schwellenwert können Sie Aktivitäten ermitteln und steuern, die übermäßig viele Zeilen lesen.

Klasse Aktivität

Definitionsdomäne

Serviceunterklasse

Umsetzungsbereich

Member

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Anzahl der Zeilen

Prädiktiv oder reaktiv

Reaktiv

Dieser Schwellenwert unterscheidet sich vom Schwellenwert SQLROWSREAD darin, dass er die Anzahl gelesener Zeilen nur ab dem Zeitpunkt steuert, zu dem eine Aktivität in eine bestimmte Serviceunterklasse eintritt, und nicht die Gesamtzahl der Zeilen, die während der Laufzeit der Aktivität gelesen werden. Dieser Schwellenwert unterscheidet sich vom Schwellenwert SQLROWSRETURNED darin, dass er die maximale Anzahl von Zeilen steuert, die während der Abfragebewertung in der aktuellen Serviceunterklasse gelesen werden, und nicht die Anzahl der Zeilen, die vom Datenserver an eine Clientanwendung zurückgegeben werden.

Indezzugriffe werden bei der Gesamtzahl der gelesenen Zeilen nicht berücksichtigt. Wenn ein Zugriffsplan bei der Abfragebewertung nur Indizes verwendet, wird nicht gegen den Schwellenwert SQLROWSREADINSC verstoßen.

Dieser Schwellenwert wird in vom Benutzer konfigurierbaren Zeitintervallen bewertet. Ist das Intervall größer als der zum Überschreiten der Anzahl gelesener Zeile erforderliche Zeitraum, überschreitet die Anzahl der für eine Aktivität in einem Member gelesener Zeilen möglicherweise den Grenzwert, bevor diese Übertretung festgestellt wird.

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- Koordinatoraktivitäten des Typs DML und die zugehörige Arbeit von Unteragenten wie die Ausführung von Unterabschnitten.
- Verschachtelte DML-Aktivitäten, die von Benutzeranwendungen abgeleitet werden. Daher sind DML-Aktivitäten, die von DB2-Logik, wie Dienstprogrammen, SYSPROC-Prozeduren oder internem SQL-Code, abgesetzt werden, von diesem Schwellenwert nicht betroffen. IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors werden als Benutzerlogik aufgefasst; Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten daher Schwellenwerte.

Mithilfe der Aktion REMAP ACTIVITY können Sie Aktivitäten steuern, indem Sie sie einer Serviceunterklasse mit anderen Ressourcenzuordnungen zuordnen.

Beispiel

Das folgende Beispiel erstellt zwei Serviceunterklassen, A1 und A2, unter einer Superklasse A, mit einem einzigen serviceklasseninternen Schwellenwert SQLROWSREADINSC, der Aktivitäten zwischen den Unterklassen neu zuordnet, nachdem bei der Abfragebewertung 10.000 Zeilen in der Serviceunterklasse A1 gelesen wurden. Ein Ereignismonitordatensatz wird protokolliert.

```
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;
```

SQLROWSRETURNED (Schwellenwert)

Der Schwellenwert SQLROWSRETURNED gibt die maximale Anzahl von Zeilen an, die vom Datenserver an den Client zurückgegeben werden können.

Typ Aktivität

Definitionsdomäne

Datenbank, Servicesuperklasse, Serviceunterklasse, Arbeitsaktion, Workload und Anweisung

Umsetzungsbereich

Datenbank

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Anzahl der Zeilen

Prädiktiv oder reaktiv

Reaktiv

Wenn mehrere Ergebnismengen von einer Anweisung CALL zurückgegeben werden, gilt der Schwellenwert separat für jede Ergebnismenge und nicht als zusammengefasster Schwellenwert für die Gesamtzahl der Zeilen, die in allen Ergebnismengen zurückgegeben werden. Wenn Sie z. B. einen Schwellenwert von 20 Zeilen definieren und die Anweisung CALL zwei Ergebnismengen mit 15 und mit 19 Zeilen zurückgibt, tritt kein Schwellenwertverstoß auf.

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- DML-Aktivitäten, die in dem Koordinatormember abgesetzt werden.
- Verschachtelte DML-Aktivitäten, die über eine Benutzeranwendung aufgerufen werden. Daher sind DML-Aktivitäten, die vom Datenserver intern abgesetzt werden, (z. B. DML-Aktivitäten, die aus den DB2-Dienstprogrammen, gespeicherten SYSPROC-Prozeduren und internem SQL-Code abgesetzt werden) von diesem Schwellenwert nicht betroffen.

SQLTEMPSPACE (Schwellenwert)

Der Schwellenwert SQLTEMPSPACE gibt den Maximalwert für den Tabellenbereich für temporäre Systemtabellen an, der von einer DML-Aktivität in jedem beliebigen Member verbraucht werden kann. DML-Aktivitäten verwenden oft einen Tabellenbereich für temporäre Tabellen für Operationen wie die Sortierung und die Bearbeitung von Zwischenergebnismengen.

Typ Aktivität

Definitionsdomäne

Datenbank, Servicesuperklasse, Serviceunterklasse, Arbeitsaktion, Workload und Anweisung

Umsetzungsbereich

Member

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

Größe des Tabellenbereichs für temporäre Tabellen in Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB)

Prädiktiv oder reaktiv

Reaktiv

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- DML-Aktivitäten, die in dem Koordinatormember abgesetzt werden.
- Verschachtelte DML-Aktivitäten, die von Benutzeranwendungen abgeleitet werden. Daher sind DML-Aktivitäten, die von DB2-Logik (Dienstprogrammen, SYS-PROC-Prozeduren oder internem SQL-Code) abgesetzt werden, von diesem Schwellenwert nicht betroffen.

Der Datenserver fasst IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors als Benutzerlogik auf. Für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten Schwellenwerte.

Zusammengefasste Schwellenwerte

Ein zusammengefasster Schwellenwert steuert Arbeitselemente in einer Datenbank gemeinsam. Die mit einem zusammengefassten Schwellenwert definierte Begrenzung fungiert als laufende Summe, zu der die gesamte vom Schwellenwert überwachte Arbeit beiträgt.

Wenn neu instanziierte Arbeit einen Verstoß gegen die obere Begrenzung verursacht, wird die entsprechende Aktion ausgelöst. Nur die Arbeit, die den Begrenzungsverstoß verursachte, ist von der ausgelösten Aktion betroffen.

Aktivitätenwarteschlange

Einige Schwellenwerte haben eine integrierte Warteschlange. Sie erlauben es Ihnen, einen Wert für die gleichzeitige Ausführung von Aktivitäten zu erzwingen, indem alle zusätzlichen Aktivitäten in die Warteschlange gestellt werden, sobald der Schwellenwert für gemeinsamen Zugriff erreicht wird, bis der festgelegte Grenzwert für die Warteschlange überschritten wird.

Wenn die Anzahl der Aktivitäten den Grenzwert für den gemeinsamen Zugriff eines Warteschlangenschwellenwerts überschreitet, werden neue Anforderungen automatisch nach dem First-in/First-out-Prinzip in die Warteschlange gestellt, bis die Warteschlange die von der Warteschlangengrenzung der Schwellenwertdefinition angegebene Größe erreicht. Die Warteschlange ist voll, wenn die Obergrenze erreicht ist. Die nächste Anforderung wird als Schwellenwertverstoß eingestuft und die für den Schwellenwert angegebene Aktion wird auf die Anforderung angewendet. Bei einer Aktion STOP EXECUTION wird neu eintreffende Arbeit z. B. zurückgewiesen.

Sie können auch die obere Warteschlangengrenzung als *unbegrenzt* definieren. In diesem Fall gibt es keine Obergrenze für die Größe der Warteschlange. Wenn neue Arbeit eintrifft und der Warteschlange hinzugefügt wird, kommt es in dieser Situation nicht zu einem Schwellenwertverstoß, unabhängig davon, wie groß die Warteschlange geworden ist. Wenn Sie als Obergrenze der Warteschlange einen festen Grenzwert und als Aktion bei einem Schwellenwertverstoß CONTINUE definieren, wird die gesamte neu eintreffende Arbeit, die die feste Grenze überschreitet, als Verstoß gegen den Schwellenwert eingestuft. Es wird ein Schwellenwertverstoß erstellt, aber die neue Arbeit wird trotzdem der Warteschlange hinzugefügt.

AGGSQLTEMPSPACE (Schwellenwert)

Der Schwellenwert AGGSQLTEMPSPACE gibt den Maximalwert für den Tabellenbereich für temporäre Systemtabellen an, der insgesamt über alle gleichzeitig aus-

geführten DML-Aktivitäten in einer Serviceunterklasse verbraucht werden darf. DML-Aktivitäten verwenden oft einen Tabellenbereich für temporäre Tabellen für Operationen wie die Sortierung und die Bearbeitung von Zwischenergebnismengen.

Klasse Zusammenfassung

Definitionsdomäne
Serviceunterklasse

Umsetzungsbereich
Member

Überwachte Arbeit
Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung
Nein

Einheit
Kilobyte, Megabyte oder Gigabyte

Prädiktiv oder reaktiv
Reaktiv

Von diesem Schwellenwert werden die folgenden Aktivitäten verfolgt:

- DML-Aktivitäten, die in dem Koordinatormember abgesetzt werden.
- Verschachtelte DML-Aktivitäten, die von Benutzeranwendungen abgeleitet werden. Daher sind DML-Aktivitäten, die von DB2-Logik, wie Dienstprogrammen, SYSPROC-Prozeduren oder internem SQL-Code, abgesetzt werden, von diesem Schwellenwert nicht betroffen. IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors werden als Benutzerlogik aufgefasst; für Aktivitäten, die über IMPORT, EXPORT und andere Befehle des Befehlszeilenprozessors aufgerufen werden, gelten daher Schwellenwerte.

CONCURRENTDBCOORDACTIVITIES (Schwellenwert)

Der Schwellenwert CONCURRENTDBCOORDACTIVITIES gibt die maximale Anzahl erkannter Koordinatoraktivitäten an, die gleichzeitig in der angegebenen Definitionsdomäne und im angegebenen Umsetzungsbereich ausgeführt werden können.

Die Verwendung dieses Schwellenwerttyps eignet sich am besten für Anwendungen, die nur jeweils eine Aktivität ausführen. Wenn eine Anwendung mehrere Aktivitäten gleichzeitig startet (wie z. B. das Absetzen einer SQL-Anweisung UPDATE bei einem geöffneten Cursor), können in Abhängigkeit von der Ebene des gemeinsamen Zugriffs, die der Schwellenwert zulässt, und dem Verhalten der anderen beteiligten Anwendungen bestimmte Szenarios mit Konkurrenzsituationen aufgrund von Warteschlangen auftreten. Wenn dieser Schwellenwert in Szenarios vorhanden ist, in denen Anwendungen mehrere Aktivitäten gleichzeitig ausführen können oder das Anwendungsverhalten unbekannt ist, wird Ihnen empfohlen, einen Schwellenwert ACTIVITYTOTALTIME für diese Aktivitäten zu definieren, damit potenzielle Szenarios mit Konkurrenzsituationen aufgrund von Warteschlangen automatisch behoben werden.

Typ Zusammenfassung

Definitionsdomäne
Datenbank, Arbeitsaktion, Servicesuperklasse, Serviceunterklasse

Umsetzungsbereich

Datenbank in anderen Umgebungen als DB2 pureScale-Umgebungen
Member in einer DB2 pureScale-Umgebung

Überwachte Arbeit

Erkannte Koordinatoraktivitäten und verschachtelte Aktivitäten (siehe „Identifizieren der Arbeit nach Typ, Aufwand oder Datenzugriff mithilfe von Arbeitsklassen“ auf Seite 52)

Warteschlangensteuerung

Ja

Einheit

Anzahl gleichzeitig ablaufender Datenbankaktivitäten

Prädiktiv oder reaktiv

Prädiktiv

Dieser Schwellenwert ist eine Verallgemeinerung des Schwellenwerts CONCURRENTWORKLOADACTIVITIES. Der Schwellenwert CONCURRENTWORKLOADACTIVITIES bezieht sich nur auf Aktivitäten in einer Workloaddomäne. Den Schwellenwert CONCURRENTDBCOORDACTIVITIES können Sie dagegen auf eine Vielzahl von Domänen, von der gesamten Datenbank bis hin zu einer einzelnen Arbeitsaktion, anwenden. Der Schwellenwert CONCURRENTDBCOORDACTIVITIES verfolgt Koordinatoraktivitäten für Anweisungen, die keine CALL-Anweisung sind, und für alle durch CALL-Anweisungen generierten verschachtelten Aktivitäten. Im Gegensatz zum Schwellenwert CONCURRENTWORKLOADACTIVITIES ist der Schwellenwert CONCURRENTDBCOORDACTIVITIES ein Warteschlangenschwellenwert.

Anmerkung: Wenn als Schwellenwertaktion CONTINUE für einen Warteschlangenschwellenwert angegeben ist, hebt dies die Begrenzung der Warteschlangengröße auf, unabhängig von einem festen Wert, der für die Warteschlangengröße angegebenen wird. Verwenden Sie das Monitorelement **concurrent_act_top**, um den höchsten gemeinsamen Zugriff von Aktivitäten (einschließlich verschachtelter Aktivitäten) zu ermitteln, der im erfassten Zeitintervall in einem Member für eine Serviceunterklasse erreicht wurde.

Beim Erstellen von Warteschlangenschwellenwerten vom Typ CONCURRENTDBCOORDACTIVITIES müssen Sie sich bewusst sein, dass es Konfigurationen gibt, die zu nicht auflösbaren Konkurrenzsituationen aufgrund von Warteschlangen führen. Beispiel:

1. Ein Schwellenwert für gemeinsamen Zugriff vom Typ CONCURRENTDBCOORDACTIVITIES wird mit einem maximalen Wert für gemeinsamen Zugriff von 1 und einer Warteschlangengröße > 1 erstellt.
2. Eine Anwendung öffnet einen Cursor, den der DB2-Datenserver als Aktivität A1 erkennt, die die eindeutige für den Schwellenwert verfügbare Zugriffsberechtigung verbraucht.
3. Während der Cursor noch geöffnet ist, setzt die Anwendung eine Anweisung UPDATE ab, die der Datenserver als Aktivität A2 erkennt und die auch dem Schwellenwert für gemeinsamen Zugriff unterliegt. Da die Aktivität A1 bereits aktiv ist, wird die neue Aktivität A2 in die Warteschlange gestellt.

Die Anwendung befindet sich jetzt in einer nicht behebbaren Konkurrenzsituation aufgrund von Warteschlangen. Sie wartet darauf, dass A2 ausgeführt wird, A2 wartet jedoch darauf, dass A1 fertig gestellt wird. Diese Situation kann ohne Eingriff von außen nicht behoben werden, außer ein Schwellenwert ACTIVITY-

TOTALTIME ist auch für diese Aktivitäten definiert. Ist der Schwellenwert ACTIVITYTOTALTIME definiert, wird die Situation behoben, wenn A1 gegen diesen Schwellenwert verstößt, indem sie die Höchstdauer überschreitet, die der Datenserver für die Verarbeitung dieser Aktivität aufwenden darf.

Dieses Beispiel kann auf mehrere Anwendungen und Warteschlangen verallgemeinert werden. Sie können diese Situation beheben, indem Sie die Werte für den gemeinsamen Zugriff erhöhen oder bestimmte Aktivitäten abbrechen, wenn die Werte für den gemeinsamen Zugriff korrekt festgelegt sind.

Der Schwellenwert CONCURRENTDBCOORDACTIVITIES wirkt sich auf unterschiedliche Aktivitätstypen wie im Folgenden dargestellt aus, um das Risiko der versehentlichen Erstellung von nicht behebaren Szenarios mit Konkurrenzsituationen aufgrund von Warteschlangen zu verringern:

- CALL-Anweisungen werden nicht vom Schwellenwert gesteuert, aber alle verschachtelten untergeordneten Aktivitäten werden vom Schwellenwert gesteuert. Sowohl anonyme Blöcke als auch autonome Routinen werden als CALL-Anweisungen klassifiziert.
- Benutzerdefinierte Funktionen (UDFs) werden vom Schwellenwert gesteuert, aber in benutzerdefinierten Funktionen verschachtelte untergeordnete Aktivitäten werden nicht von ihm gesteuert. Wenn eine autonome Routine aus einer benutzerdefinierten Funktion heraus aufgerufen wird, werden weder die autonome Routine noch eine untergeordnete Aktivität der autonomen Routine vom Schwellenwert gesteuert.
- Triggeraktionen, die CALL-Anweisungen aufrufen, und die untergeordneten Aktivitäten dieser CALL-Anweisungen werden nicht vom Schwellenwert gesteuert. Die Anweisungen INSERT, UPDATE und DELETE, die das Auslösen eines Triggers verursachen können, werden selbst vom Schwellenwert gesteuert.

CONCURRENTWORKLOADACTIVITIES (Schwellenwert)

Der Schwellenwert CONCURRENTWORKLOADACTIVITIES gibt die maximale Anzahl von Koordinatoraktivitäten und verschachtelten Aktivitäten an, die gleichzeitig in einem Workloadvorkommen ausgeführt werden können.

Typ Zusammenfassung

Definitionsdomäne

Workload

Umsetzungsbereich

Workloadvorkommen

Überwachte Arbeit

Erkannte Koordinatoraktivitäten und verschachtelte Aktivitäten (siehe „Aktivitäten“ auf Seite 15)

Warteschlangensteuerung

Nein

Einheit

Anzahl gleichzeitig ablaufender Workloadaktivitäten

Prädiktiv oder reaktiv

Prädiktiv

Dieser Schwellenwert bezieht sich auf ein einziges Workloadvorkommen. Wenn Sie mehrere Vorkommen einer Workload haben, die gleichzeitig ausgeführt werden, gilt der Schwellenwert separat für jedes Workloadvorkommen. Zu den verfolgten Aktivitäten gehören alle erkannten Koordinatoraktivitäten und alle verschachtelten

Aktivitäten, die durch die Ausführung der Koordinatoraktivität generiert werden. Wenn z. B. eine gespeicherte Prozedur aufgerufen wird und diese gespeicherte Prozedur SQL-Code ausführt, werden anders als beim Schwellenwert CONCURRENTDBCOORDACTIVITIES sowohl die Anweisung CALL (die Koordinatoraktivität) als auch die von der gespeicherten Prozedur ausgeführten SQL-Anweisungen (die verschachtelten Aktivitäten) in der Gesamtsumme des Schwellenwerts berücksichtigt.

Dieser Schwellenwert bezieht sich nicht auf die Anweisungen COMMIT, ROLLBACK und ROLLBACK to SAVEPOINT.

Aspekte von verschachtelten Aktivitäten

Die verschachtelten Aktivitäten, die von diesem Schwellenwert verfolgt werden, müssen die folgenden Bedingungen erfüllen:

- Sie müssen eine erkannte Koordinatoraktivität sein. Verschachtelte Koordinatoraktivitäten, die laut der Beschreibung in „Identifizieren der Arbeit nach Typ, Aufwand oder Datenzugriff mithilfe von Arbeitsklassen“ auf Seite 52 keine erkannten Typen sind, werden nicht gezählt.
- Sie müssen direkt von Benutzerlogik aufgerufen werden, z. B. einer benutzerdefinierten gespeicherten Prozedur, die SQL-Code absetzt, oder von der gespeicherten Prozedur SYSPROC.ADMIN_CMD. Verschachtelte Koordinatoraktivitäten, die durch den Aufruf eines DB2-Dienstprogramms oder von anderem Code in den Schemata SYSIBM, SYSFUN. oder SYSPROC gestartet werden, werden nicht zu der von diesem Schwellenwert festgelegten Obergrenze gezählt.

Beispiel

In diesem Beispiel ist der Maximalwert des Schwellenwerts CONCURRENTWORKLOADACTIVITIES auf 5 gesetzt. Durch die Benutzerlogik ergibt sich die folgende Operationsfolge in einem Workloadvorkommen:

1. Ein Befehl **load** wird abgesetzt: die aktuelle Anzahl Workloadaktivitäten beträgt 1.
 - Der Befehl **load** setzt intern SQL-Code ab. Die aktuelle Anzahl Workloadaktivitäten beträgt 1. (Von einem Dienstprogramm generierter SQL-Code wird beim Schwellenwert CONCURRENTWORKLOADACTIVITIES nicht berücksichtigt.)
 - Der Befehl **load** endet. Die aktuelle Anzahl Workloadaktivitäten beträgt 0.
2. Die gespeicherte Prozedur SYSPROC.SP1 wird mit CALL aufgerufen. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.
 - Die gespeicherte Prozedur SYSPROC.SP1 generiert SQL-Code. Die aktuelle Anzahl Workloadaktivitäten beträgt 1. (Von einem Dienstprogramm generierter SQL-Code wird beim Schwellenwert CONCURRENTWORKLOADACTIVITIES nicht berücksichtigt.)
 - Die gespeicherte Prozedur SYSPROC.SP1 endet. Die aktuelle Anzahl Workloadaktivitäten beträgt 0.
3. Ein Cursor C1 wird geöffnet. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.
4. Ein Befehl **runstats** wird abgesetzt. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.
 - Der Befehl **runstats** generiert SQL-Code. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.

- Der Befehl **runstats** endet. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.
- 5. Der Cursor C1 wird geschlossen. Die aktuelle Anzahl Workloadaktivitäten beträgt 0.
- 6. Die gespeicherte Prozedur BOB.SP1 wird mit CALL aufgerufen. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.
 - Die gespeicherte Prozedur BOB.SP1 öffnet drei Cursor. Die aktuelle Anzahl Workloadaktivitäten beträgt 4.
 - Die gespeicherte Prozedur BOB.SP1 ruft die gespeicherte Prozedur SYSPROC.SP2 auf. Die aktuelle Anzahl Workloadaktivitäten beträgt 5.
 - Die gespeicherte Prozedur SYSPROC.SP2 setzt SQL-Code ab. Die aktuelle Anzahl Workloadaktivitäten beträgt 5.
 - Die gespeicherte Prozedur SYSPROC.SP2 endet. Die aktuelle Anzahl Workloadaktivitäten beträgt 4.
 - Die gespeicherte Prozedur BOB.SP1 ruft die gespeicherte Prozedur BOB.SP2 auf. Die aktuelle Anzahl Workloadaktivitäten beträgt 5.
 - Die gespeicherte Prozedur BOB.SP2 setzt SQL-Code ab. Zu diesem Zeitpunkt wird der Schwellenwert ausgelöst.
 - Die gespeicherte Prozedur BOB.SP2 endet. Die aktuelle Anzahl Workloadaktivitäten beträgt 4.
 - Die gespeicherte Prozedur BOB.SP1 endet. Die aktuelle Anzahl Workloadaktivitäten beträgt 0.
- 7. Ein Cursor C2 wird geöffnet. Die aktuelle Anzahl Workloadaktivitäten beträgt 1.
- 8. Die gespeicherte Prozedur BOB.SP2 wird mit CALL aufgerufen. Die aktuelle Anzahl Workloadaktivitäten beträgt 2.

CONCURRENTWORKLOADOCCURRENCES (Schwellenwert)

Der Schwellenwert CONCURRENTWORKLOADOCCURRENCES ist ein zusammengefasster Schwellenwert, der die maximale Anzahl von Workloadvorkommen angibt, die gleichzeitig in dem Koordinatormember ausgeführt werden können.

Typ Zusammenfassung

Definitionsdomäne

Workload

Umsetzungsbereich

Member

Überwachte Arbeit

Workloadvorkommen

Warteschlangensteuerung

Nein

Einheit

Anzahl gleichzeitig ablaufender Workloadvorkommen

Prädiktiv oder reaktiv

Prädiktiv

Wenn beim Start eines Workloadvorkommens die von ihm generierte Arbeit an Nicht-Koordinatormember gesendet wird, zählt die Arbeit in diesen Mitgliedern nicht zur Gesamtsumme des Schwellenwerts im Koordinatormember. Nehmen wir z. B. an, dass ein Schwellenwert CONCURRENTWORKLOADOCCURRENCES so definiert ist, dass nur ein Vorkommen der Workload A in einem Member zugelassen

sen ist. Nehmen wir weiter an, dass eine Anwendung eine Verbindung zu Member 1 herstellt, wobei ein Vorkommen der Workload A gestartet wird, und dass diese Workload Arbeit an die Datenbankmember 1, 2 und 3 sendet. In dieser Situation gibt es von Workload A ein Vorkommen in Member 1 und kein Vorkommen auf den Datenbankpartitionen 2 und 3. Wenn eine andere Anwendung eine Verbindung zu Member 1 herstellt und versucht wird, ein anderes Vorkommen von Workload A in Member 1 zu instanzieren, wird diese Workload daher zurückgewiesen. In den Datenbankmembern 2 und 3 können jedoch noch neue Vorkommen von Workload A gestartet werden.

TOTALMEMBERCONNECTIONS (Schwellenwert)

Der Schwellenwert TOTALMEMBERCONNECTIONS gibt die maximale Anzahl gleichzeitig bestehender Datenbankverbindungen in einem Koordinatormember für eine Datenbank an. Dieser Schwellenwert steuert also die maximale Anzahl von Clients, die in jedem ihrer Datenbankmember eine Verbindung zur Datenbank herstellen können.

Dieser Schwellenwert wird für Benutzer mit der Berechtigung DBADM und WLMADM nicht erzwungen.

Typ Zusammenfassung

Definitionsdomäne
Datenbank

Umsetzungsbereich
Member

Überwachte Arbeit
Verbindungen

Warteschlangensteuerung
Ja (erzwungen bei 0)

Einheit
Anzahl gleichzeitig bestehender Verbindungen

Prädiktiv oder reaktiv
Prädiktiv

Wenn Sie z. B. den Schwellenwert TOTALMEMBERCONNECTIONS auf 10 setzen und die Datenbank fünf Member hat, können in jedem Member bis zu 10 Clients gleichzeitig verbunden sein, d. h. insgesamt 50 Clientverbindungen für die gesamte Datenbank.

Der Schwellenwert TOTALMEMBERCONNECTIONS steuert nur Koordinatorverbindungen. Verbindungen, die von Subagenten hergestellt werden, werden nicht in den Schwellenwert eingerechnet.

Dieser Schwellenwert ist hilfreich für Situationen, in denen Sie mehrere Datenbanken in derselben Instanz haben wollen. Durch das Festlegen des Schwellenwerts TOTALMEMBERCONNECTIONS in einem Member wird sichergestellt, dass Clientverbindungen von einer Datenbank nicht alle verfügbaren Verbindungen in einem Member verwenden können.

Stellen Sie sicher, dass Sie den Konfigurationsparameter **max_connections** des Datenbankmanagers hoch genug eingestellt haben, um die maximale Anzahl von Verbindungen zu unterstützen, die Sie für die gesamte Datenbank erwarten. Wenn Sie einen Schwellenwert TOTALMEMBERCONNECTIONS für eine Datenbank festlegen, müssen Sie **max_connections** auf mindestens denselben Wert einstellen. Wenn

Sie mehrere Datenbanken auf derselben Instanz ausführen wollen, müssen Sie sicherstellen, dass Sie **max_connections** hoch genug einstellen, sodass er die maximale Anzahl von Verbindungen für alle Datenbanken unterstützt. Der Datenserver prüft diese Bedingung nicht, da es unmöglich ist, vorab zu wissen, wie viele der Datenbanken gleichzeitig aktiv sein werden.

Anmerkung: Wenn als Schwellenwertaktion CONTINUE für diesen Warteschlangenschwellenwert angegeben ist, hebt dies die Begrenzung der Warteschlangengröße auf, unabhängig von einem festen Wert, der für die Warteschlangengröße angegeben wird, obwohl die Warteschlangengröße normalerweise bei 0 für den Schwellenwert TOTALMEMBERCONNECTIONS erzwungen wird.

TOTALSCMEMBERCONNECTIONS (Schwellenwert)

Der Schwellenwert TOTALSCMEMBERCONNECTIONS gibt die maximale Anzahl gleichzeitig bestehender Datenbankverbindungen in einem Koordinatormember für eine Servicesuperklasse an.

Typ Zusammenfassung

Definitionsdomäne

Servicesuperklasse

Umsetzungsbereich

Member

Überwachte Arbeit

Verbindungen

Warteschlangensteuerung

Ja

Einheit

Anzahl gleichzeitig bestehender Verbindungen in einer Serviceklasse

Prädiktiv oder reaktiv

Prädiktiv

Wenn der Schwellenwert TOTALSCMEMBERCONNECTIONS in der Serviceklasse erreicht wird, werden nachfolgende Koordinatorverbindungen, die mit der Servicesuperklasse verknüpft sind, in die Warteschlange gestellt, bis die festgelegte Warteschlangengröße erreicht ist. Standardmäßig ist die Warteschlangengröße null, was heißt, dass keine Verbindungen in die Warteschlange gestellt werden können. Wenn eine Verbindung in die Warteschlange eines Schwellenwerts TOTALSCMEMBERCONNECTIONS gestellt wird, gilt die Verbindung als *temporär*.

Anmerkung: Wenn als Schwellenwertaktion CONTINUE für einen Warteschlangenschwellenwert angegeben ist, hebt dies die Begrenzung der Warteschlangengröße auf, unabhängig von einem festen Wert, der für die Warteschlangengröße angegeben wird.

Überwachte Verbindungen sind sowohl neue Clientverbindungen als auch vorhandene Clientverbindungen, die von einer anderen Serviceklasse in diese Serviceklasse wechseln. Verbindungen wechseln zwischen Serviceklassen, indem sie mit einer anderen Workloaddefinition verknüpft werden, die einer anderen Serviceklasse zugeordnet ist. Die Workloadneubewertung erfolgt nur an Transaktionsgrenzen, sodass Verbindungen nur an Transaktionsgrenzen zwischen Serviceklassen wechseln können; da jedoch Ressourcen, die einem WITH HOLD-Cursor zugeordnet sind, über Transaktionsgrenzen beibehalten werden, können Verbindungen mit einem offenen WITH HOLD-Cursor nicht zwischen Servicesuperklassen wechseln. Wenn der Verbindungskonzentrator aktiv ist, verlässt jede gewechselte Anwendung die

Serviceklasse. Wenn die Anwendung bei der nächsten Anweisung eingewechselt wird, muss sie der Serviceklasse erneut beitreten und demzufolge den Schwellenwert passieren.

Wenn der Schwellenwert für die Warteschlangenlänge erreicht ist, wird die Schwellenwertaktion ausgelöst. Der Schwellenwert TOTALSCMEMBERCONNECTIONS steuert nur Koordinatorverbindungen. Verbindungen, die von Subagenten hergestellt werden, werden nicht in den Schwellenwert eingerechnet.

Wenn Sie einen Schwellenwert für TOTALMEMBERCONNECTIONS festlegen, definieren Sie ihn groß genug, sodass er den Schwellenwert für TOTALSCMEMBERCONNECTIONS aufnehmen kann. Wenn Sie z. B. fünf Servicesuperklassen für eine Datenbank definieren und jeder von ihnen einen Schwellenwert von 10 für TOTALSCMEMBERCONNECTIONS hat, sollte TOTALMEMBERCONNECTIONS mindestens einen Schwellenwert von 50 haben.

UOW-Schwellenwerte

Ein UOW-Schwellenwert wendet Steuerelemente auf einzelne UOWs an. Wenn die Ressourcenauslastung einer einzelnen UOW gegen die Obergrenze des Schwellenwerts verstößt, der diese Aktivität verfolgt, wird die entsprechende Aktion ausgelöst und einmal auf entweder die Anweisung (beim Erzwingen des Anwendungsende) oder die UOW (bei Rollback) angewendet.

UOWTOTALTIME (Schwellenwert)

Der Schwellenwert UOWTOTALTIME gibt die Höchstdauer an, die eine UOW in der DB2-Engine verbringen darf.

Typ UOW

Definitionsdomäne

Datenbank, Workload, Servicesuperklasse

Umsetzungsbereich

Datenbank

Überwachte Arbeit

Details finden Sie in den später in diesem Abschnitt enthaltenen Informationen.

Warteschlangensteuerung

Nein

Einheit

In Sekunden, Minuten, Stunden oder Tagen ausgedrückte Zeitdauer. Wird die Zeiteinheit in Sekunden angegeben, muss dieser Wert ein Vielfaches von 10 sein.

Prädiktiv oder reaktiv

Reaktiv

Die Aktion STOP EXECUTION für einen Schwellenwert UOWTOTALTIME führt eine Rollback-Operation für die UOW durch. Die Aktion FORCE APPLICATION erzwingt das Ende der Anwendung, zu der die UOW gehört. Die Option COLLECT ACTIVITY DATA kann für diesen Schwellenwert zwar angegeben werden, wird aber ignoriert.

Ein in der Workloaddomäne definierter Schwellenwert UOWTOTALTIME überschreibt jeglichen in der Domäne für Servicesuperklassen definierten Schwellenwert UOWTOTALTIME. Ein in der Domäne für Servicesuperklassen definierter

Schwellenwert UOWTOTALTIME überschreibt jeglichen in der Datenbankdomäne definierten Schwellenwert UOWTOTALTIME.

Beispiel

Das folgende Beispiel erstellt einen Schwellenwert FORCE10MINUTEUOW, der einen Befehl FORCE APPLICATION absetzt, wenn eine UOW länger als 10 Minuten ausgeführt wurde.

```
CREATE THRESHOLD FORCE10MINUTEUOW FOR DATABASE ACTIVITIES
ENFORCEMENT DATABASE
WHEN UOWTOTALTIME > 10 MINUTES
FORCE APPLICATION;
```

Prioritätssteuerung nach Verweildauer für laufende Aktivitäten

Die Prioritätssteuerung nach Verweildauer ist eine Methode des Workload-Managements, bei der sich die Priorität von in Bearbeitung befindlichen Aktivitäten automatisch im Lauf der Zeit ändert.

Je länger eine Aktivität ausgeführt wird, desto niedriger ist ihre Priorität und desto weniger Ressourcen werden ihr zugewiesen. Mit der Prioritätssteuerung nach Verweildauer können Sie länger laufende Aktivitäten steuern, sodass der Durchsatz für kürzer laufende Aktivitäten verbessert werden kann. Die Methode der Prioritätssteuerung nach Verweildauer funktioniert, wenn die Ressourcensteuerung auf Verschiebungen der Arbeit zwischen Serviceklassen reagiert. Wenn sich die Serviceklasse von Arbeit ändert, die bereits ausgeführt wird, spiegelt sich die Verschiebung also in den Ressourcen wieder, die diese Arbeit (sowie andere Arbeit in der neuen Serviceklasse) erhält. Diese Methode lässt sich am besten bei Verwendung der expliziten CPU-Steuererelemente implementieren, die vom CPU-Dispatcher oder durch Integration in Workload-Management-Produkte des Betriebssystems bereitgestellt werden.

Ändern der Priorität von Aktivitäten durch Neuordnung

Systemressourcen werden mithilfe von Serviceklassen zugeordnet und gesteuert. Mit der Prioritätssteuerung nach Verweildauer kann die Priorität einer Aktivität geändert werden, indem die Aktivität von einer Serviceklasse in eine andere Serviceklasse verschoben wird. Die Priorität erhöht sich, wenn die neue Serviceklasse mehr Ressourcen hat, und sie verringert sich, wenn die neue Serviceklasse weniger Ressourcen hat. Aktivitäten werden bei einem Verstoß gegen einen Schwellenwert mit einer Aktion REMAP ACTIVITY verschoben, basierend auf einer vordefinierten maximalen Nutzung einer bestimmten Ressource wie Prozessorzeit oder gelesene Zeilen. Nachdem eine Aktivität einer neuen Serviceklasse zugeordnet wurde, wird sie weiter ausgeführt, wobei die neuen Ressourcenbeschränkungen gelten.

Eine einfache Methode, die Sie für eine schnellere Ausführung kurzer Abfragen verwenden können, besteht darin, eine Reihe von Serviceklassen mit abnehmender Ressourcenpriorität und mit Schwellenwertaktionen zu definieren, die Aktivitäten zwischen den Serviceunterklassen verschieben. Mit dieser Konfiguration können Sie die Priorität einer über einen längeren Zeitraum ausgeführten Arbeit im Lauf der Zeit herabsetzen bzw. nach Verweildauer steuern und dadurch vielleicht die Antwortzeiten für die Arbeit verbessern, die über einen kürzeren Zeitraum ausgeführt, ohne die auf Ihrem Datenserver ausgeführten Aktivitäten genau kennen zu müssen.

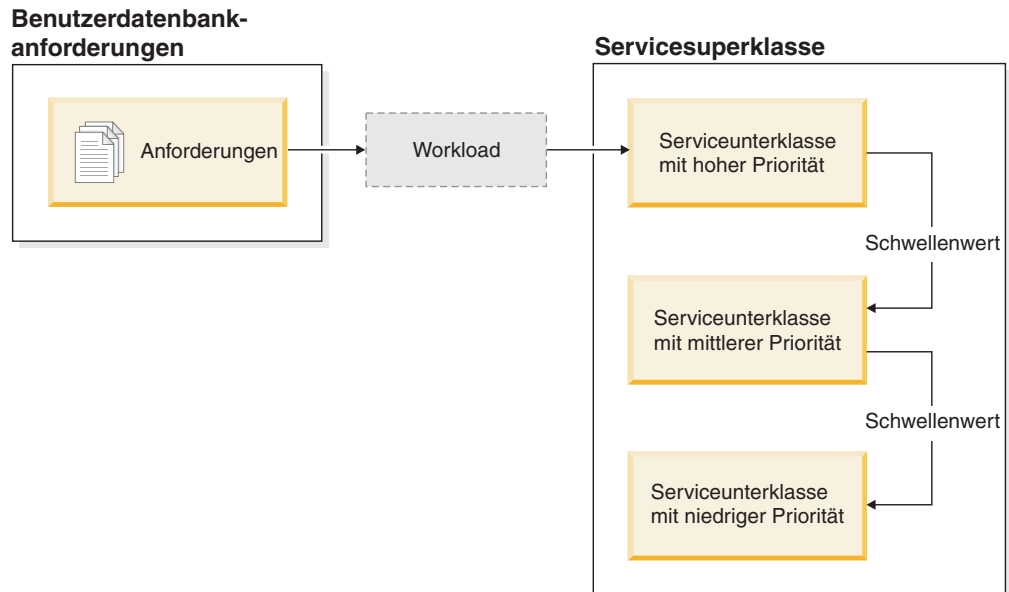


Abbildung 21. Eine einfache mehrschichtige Konfiguration mit drei Serviceklassen mit abnehmender Priorität

Sie können diese Konfiguration erstellen, indem Sie einer Serviceklasse für alle relevanten Ressourcen eine hohe Priorität, einer zweiten Serviceklasse eine mittlere Priorität und einer dritten Serviceklasse eine niedrige Priorität zuweisen. Sobald Arbeit im System eintrifft, wird sie automatisch in die erste Serviceklasse gesetzt und ihre Ausführung beginnt mit den Einstellungen für hohe Priorität dieser Serviceklasse. Wenn Sie auch Schwellenwerte für jede der Serviceklassen definieren, die die verwendete Zeit oder die belegten Ressourcen während der Ausführung beschränken, wird die Arbeit dynamisch der nächst niedrigeren Serviceklasse zugeordnet, wenn gegen den Schwellenwert der nächst höheren Serviceklasse verstoßen wird. Diese dynamische Ressourcensteuerung wird wiederholt angewendet, bis die Arbeit abgeschlossen ist oder bis sie sich in der Klasse mit der niedrigsten Priorität befindet, wo sie verbleibt, bis sie abgeschlossen ist oder bis Sie den Abbruch der Ausführung erzwingen.

Serviceklasseninterne Schwellenwerte

Die Neuordnung von Aktivitäten ist mit jedem der serviceklasseninternen Schwellenwerte verfügbar, die den Anteil einer Ressource steuern, der bei der Ausführung einer Aktivität in einer bestimmten Serviceunterklasse verwendet werden darf. Beispiele für Ressourcen sind der Anteil der verwendeten Prozessorzeit (Schwellenwert CPUTIMEINSC) und die Anzahl der von einer Anwendung pro Aktivität pro Member gelesenen Zeilen (Schwellenwert SQLROWSREADINSC). Diese Schwellenwerte unterscheiden sich von anderen Aktivitätsschwellenwerten, die Ressourcen steuern, die während der gesamten Laufzeit einer Aktivität verwendet werden.

Aufgrund der Steuerung, die diese serviceklasseninternen Schwellenwerte über Serviceunterklassen bereitstellen, können Sie serviceklasseninterne Schwellenwerte nur in einer Serviceunterklassendomäne definieren. Die serviceklasseninternen Schwellenwerte stellen Steuerelemente ähnlich wie DB2 Governor-Regeln bereit, die auf Monitorelemente für die Prozessorzeit und die gelesenen Zeilen reagieren.

Wenn einem serviceklasseninternen Schwellenwert eine Aktion REMAP ACTIVITY zugeordnet ist, prüfen Agenten, die für die Aktivität arbeiten, in regelmäßigen Abständen in jedem Member, ob gegen den Schwellenwert verstoßen wurde. Wenn ein Agent einen Schwellenwertverstoß in einem Member feststellt, löst der Agent die Aktion REMAP ACTIVITY für die Aktivität in dem member aus und ordnet sich anschließend der Zielserviceunterklasse erneut zu. Alle anderen Agenten, die für die Aktivität in demselben Member arbeiten, ordnen sich der Zielserviceunterklasse zu, wenn sie erkennen, dass die Aktivität neu zugeordnet worden ist. Nur ein Agent erkennt den Schwellenwertverstoß und ordnet die Aktivität neu zu. Die Aktivität gilt als neu zugeordnet, nachdem der Agent den Schwellenwertverstoß erkannt und die Neuordnung durchgeführt hat.

Zwei Monitorelemente stellen Informationen zur Neuordnung von Aktivitäten innerhalb von Serviceunterklassen bereit. Das Monitorelement **act_remapped_in** stellt einen Zähler bereit, der aufzeichnet, wie viele Aktivitäten in einer Serviceunterklasse neu zugeordnet wurden. Er wird für die Zielserviceunterklasse jedesmal erhöht, wenn ihr eine Aktivität neu zugeordnet wird. Analog dazu wird der Zähler des Monitorelements **act_remapped_out** jedesmal für die Quellenserviceunterklasse erhöht, wenn in ihr die Zuordnung einer Aktivität aufgehoben wurde. Ein weiteres Monitorelement, **num_remaps**, zählt, wie oft insgesamt eine Aktivität zwischen Serviceunterklassen neu zugeordnet wurde.

Eine Aktivität kann mehrmals anderen Serviceunterklassen zugeordnet werden. Sie kann auch in ihre ursprüngliche Serviceunterklasse zurückkehren, nachdem sie einer anderen Serviceunterklasse zugeordnet worden war.

Die serviceklasseninternen Schwellenwerte werden für eine Aktivität ohne Koordination in jedem Member separat bewertet. Da es zwischen Members keine Koordination gibt, ist es möglich, dass bei der Neuordnung einer Aktivität in einem Member dieselbe Aktivität in unterschiedlichen Members gleichzeitig in unterschiedlichen Serviceunterklassen enthalten ist.

Wenn Subagentenarbeit für eine Aktivität in einem fernen Member ist und weitere Arbeit für dieselbe Aktivität später an dasselbe Member gesendet wird, wird die Aktivität in derselben Serviceunterklasse erneut gestartet wie der Agent, der die Anforderung an das Member gesendet hat. Wenn Sie einen serviceklasseninternen Schwellenwert für diese Serviceunterklasse definiert haben, startet der Zeitgeber oder Zähler für die Aktivität in dem fernen Member wieder bei null.

Bei verschachtelten Aktivitäten werden über- und untergeordnete Aktivitäten getrennt verfolgt. Wenn daher eine untergeordnete Aktivität zu viele Ressourcen belegt, verstößt nur diese Aktivität und nicht ihre übergeordneten oder gleichgeordneten Aktivitäten gegen einen Schwellenwert.

Verwenden der serviceklasseninternen Schwellenwerte

Auf Datenservern, auf denen die Prozessorzeit die primäre Ressource ist, um die Aktivitäten konkurrieren, verwenden Sie den Schwellenwert CPUTIMEINSC als wichtigste Steuerungsmaßnahme. Auf Datenservern, auf denen Abfragen, die viele Tabellenzeilen lesen, vorwiegend eine Konkurrenzsituation bei Ein-/Ausgabe verursachen, verwenden Sie SQLROWSREADINSC. Auf Systemen, auf denen eine Kombination von starker Prozessor- und E/A-Aktivität auftritt, verwenden Sie eine Kombination der Schwellenwerte CPUTIMEINSC und SQLROWSREADINSC.

Sie sollten die Agentenpriorität der Serviceunterklassen relativ zueinander festlegen, sodass Ihr Datenserver Aktivitäten mit unterschiedlicher Geschäftspriorität

unterschiedlich behandeln kann. Beachten Sie, dass die Agentenpriorität der Standardssystemklasse immer höher als die jeder benutzerdefinierter Serviceklasse sein sollte, um eine negative Auswirkung auf die Leistung zu vermeiden. Die Agentenpriorität der Standardverwaltungsklasse kann niedriger als die Ihrer benutzerdefinierten Serviceklassen festgelegt werden.

Welchen Anteil einer bestimmten Ressource Aktivitäten in einer Serviceunterklasse verbrauchen dürfen, bevor sie einer anderen Serviceunterklasse zugeordnet werden, hängt größtenteils von Ihrer jeweiligen Umgebung ab. Sie müssen überwachen, wie Aktivitäten auf Ihrem Datenserver verarbeitet werden, um den besten Wert für jede Schwellenwertbedingung herauszufinden. Wenn der Maximalwert für die zugeteilte Prozessorzeit oder die maximale Anzahl für gelesene Zeilen in einer Serviceklasse zu hoch eingestellt ist, werden Aktivitäten in derselben Serviceunterklasse gestartet und beendet, unabhängig davon, wie viele Ressourcen jede Aktivität benötigt. Dies ist nicht zweckmäßig. Wenn die maximale Prozessorzeit oder die maximale Anzahl gelesener Zeilen zu niedrig eingestellt ist, wird keine Aktivität in der Serviceklasse beendet, der sie ursprünglich zugeordnet wurde, und jede Aktivität wird schließlich unabhängig von der Geschäftspriorität einer anderen Serviceklasse zugeordnet. In beiden Fällen bietet Ihre mehrschichtige Konfiguration keinen Vorteil für den Gesamtdurchsatz auf Ihrem Datenserver und Aktivitäten werden nicht effektiv entsprechend ihrer Geschäftspriorität behandelt.

Neben der Ermittlung, welchen Anteil einer bestimmten Ressource eine Aktivität verbrauchen darf, erlauben einige Schwellenwerte das Definieren eines Prüfintervalls, in dem der Datenserver auf Schwellenwertverstöße prüft. Diese Funktionalität wird für Schwellenwerte bereitgestellt, bei denen es zu aufwendig ist, den Schwellenwert jedesmal zu prüfen, wenn eine Einheit der gesteuerten Ressource verbraucht wird. Sie bestimmen die Latenzzeit, mit der Verstöße gegen diese Schwellenwerte erkannt werden. Die Schwellenwerte CPUTIME und SQLROWSREAD sowie ihre serviceklasseninternen Entsprechungen CPUTIMEINSC und SQLROWSREADINSC unterstützen Prüfintervalle. Bei seriellen Datenbankinstanzen entspricht das Prüfintervall der Echtzeit, die zwischen Prüfungen auf einen Schwellenwertverstoß verstreichen soll. In Datenbankumgebungen mit mehreren Mitgliedern oder bei SMP-Instanzen (SMP - symmetrischer Multiprozessor) sollte das Prüfintervall auf einen Wert gesetzt werden, der unter der tatsächlich verstrichenen Zeit liegt, um zu berücksichtigen, dass möglicherweise mehr als ein Agent gleichzeitig Prozessorzeit für die Aktivität ansammelt. Zur Berechnung des ungefähren Prüfintervalls in Umgebungen mit Datenbanken mit mehreren Mitgliedern oder bei SMP-Instanzen teilen Sie die Echtzeit, die zwischen Prüfungen verstreichen soll, durch den Grad der Parallelität für die Aktivität und verwenden Sie den Ergebniswert für die Klausel CHECKING EVERY.

Beispiel: Wenn in einer Datenbank mit einem einzigen Member ein Schwellenwert CPUTIMEINSC eine Aktion REMAP ACTIVITY auslösen soll, nachdem 30 Sekunden Prozessorzeit verbraucht wurden, können Sie das Prüfintervall auf 30 Sekunden einstellen und sicher sein, dass die Schwellenwertaktion spätestens ausgelöst wird, nachdem 30 Sekunden Prozessorzeit verbraucht wurden. (Die verbrauchte Prozessorzeit kann die tatsächlich verstrichene Zeit nicht übersteigen.) Wenn Sie in einer Datenbankumgebung mit mehreren Mitgliedern einen Schwellenwert CPUTIMEINSC definieren, der auf 5 Sekunden mit einem Prüfintervall von 5 Sekunden gesetzt ist, und eine Aktivität einen Koordinatorpartitionsagenten und 4 zugehörige Subagenten hat, die für sie arbeiten, ist es möglich, dass die Aktivität 5 Sekunden CPU-Zeit in nur einer Sekunde Echtzeit verbraucht, da 5 Agenten gleichzeitig jeweils eine Sekunde Prozessorzeit ansammeln. Damit die Aktivität nicht ein Mehrfaches von 5 Sekunden der Prozessorzeit verbraucht, sollte das Prüfintervall in diesem Fall auf 1 Sekunde gesetzt werden.

Weitere Informationen zur Verwendung der Schwellenwerte finden Sie in den Beispielschichtenscripts und in den Szenarios zur Prioritätssteuerung nach Verweildauer.

Auswirkung der Neuordnung auf Schwellenwerte

Welche Schwellenwerte nach der Neuordnung durch eine Aktion REMAP ACTIVITY weiter gültig sind, hängt davon ab, ob die Schwellenwerte nur für eine bestimmte Serviceunterklasse oder während der gesamten Laufzeit einer Aktivität gelten.

Wenn Sie eine Aktivität einer neuen Serviceunterklasse zuordnen, ändern sich nur die serviceklasseninternen Schwellenwerte wie CPUTIMEINSC und SQLROWSREADINSC. Diese serviceklasseninternen Schwellenwerte beeinflussen eine Aktivität nicht mehr, nachdem diese die Quellenserviceunterklasse verlässt. Sie werden ersetzt durch die entsprechenden Schwellenwerte für die Zielunterklasse, sofern diese Schwellenwerte definiert sind. Alle anderen Aktivitätsschwellenwerte aus der Serviceunterklasse, der die Aktivität ursprünglich zugeordnet wurde, bleiben unverändert und die entsprechenden Schwellenwertzeitgeber und -zähler werden nicht zurückgesetzt. Die Aktivität wird nicht anhand anderer Schwellenwerte neu bewertet, die Sie für die Zielserviceunterklasse definiert haben.

Nehmen wir z. B. an, dass wie folgt zwei Serviceunterklassen mit Schwellenwerten definiert sind:

- Serviceunterklasse A mit den folgenden Schwellenwerten:
 - Laufzeitschwellenwert TH1 vom Typ ACTIVITYTOTALTIME mit einer Aktion STOP EXECUTION nach 30 Minuten
 - Serviceklasseninterner Schwellenwert TH2 vom Typ SQLROWSREADINSC mit einer Aktion REMAP ACTIVITY für die Neuordnung in Serviceunterklasse B, nachdem mehr als 2000 Zeilen gelesen wurden
- Serviceunterklasse B mit den folgenden Schwellenwerten:
 - Laufzeitschwellenwert TH3 vom Typ ACTIVITYTOTALTIME mit einer Aktion STOP EXECUTION nach 5 Minuten
 - Schwellenwert TH4 vom Typ SQLROWSREADINSC mit einer Aktion STOP EXECUTION, nachdem mehr als 1000 Zeilen gelesen wurden

Wenn eine Aktivität in Serviceunterklasse A in das System eintritt, gelten die beiden Schwellenwerte TH1 und TH2 für die Aktivität. Wenn die Aktivität bei der Abfragebewertung mehr als 2000 Zeilen liest, wird sie dynamisch der Serviceunterklasse B zugeordnet. Durch die Neuordnung der Aktivität zu Unterklasse B ändern sich die serviceklasseninternen Schwellenwerte. Daher gilt nun TH4 anstelle von TH2 für die Aktivität. Die Zähler für beide Schwellenwerte werden auf null zurückgesetzt und obwohl die Aktivität mehr als 2000 Zeilen in der ursprünglichen Serviceunterklasse gelesen hat, wird der Zähler für TH4 mit null erneut gestartet; die Aktivität muss während ihrer Ausführung in Serviceunterklasse B mehr als 1000 Zeilen lesen, bevor gegen den Schwellenwert TH4 verstoßen wird. Der Schwellenwert TH1, der während der gesamten Laufzeit der Aktivität gilt, wird weiterhin angewendet, obwohl die Aktivität jetzt in einer anderen Unterklasse ausgeführt wird. Der Schwellenwert TH3 hat keine Kontrolle über die neu zugeordnete Aktivität, da er nicht für die erste Serviceunterklasse galt, in die die Aktivität zu Beginn der Ausführung eintrat.

Beispielskripts für die Prioritätssteuerung nach Verweildauer

Verwenden Sie die bereitgestellten Beispielskripts, um auf Ihrem Datenserver schnell eine mehrschichtige Serviceklassenkonfiguration zu erstellen. Mit einer mehrschichtigen Konfiguration können Sie bestimmte Leistungsziele durch Herabsetzen der Priorität von länger laufenden Abfragen erreichen. Dieser Vorgang wird auch Prioritätssteuerung nach Verweildauer genannt. Sie können die Skripts auch entsprechend Ihrer eigenen Geschäftsprioritäten ändern, wenn Sie sie an Ihre Umgebung anpassen.

Die beiden Beispielskripts `wlmtiersdefault.db2` und `wlmtierstimerons.db2` sollen zeigen, wie Sie die Prioritätssteuerung nach Verweildauer auf Ihrem Datenserver verwenden könnten, um den Gesamtdurchsatz zu verbessern. Der DB2-Workload-Manager stellt Steuerelemente bereit, die für den Gesamtdurchsatz auf Ihrem Datenserver hilfreich sein können. Um jedoch den vollen Nutzen der Skripts und der Prioritätssteuerung nach Verweildauer im Allgemeinen zu erhalten, müssen Sie Ihren Datenserver über einen längeren Zeitraum ausführen und überwachen, wie diese Arbeit ausgeführt wird, und dann die Einstellungen für die Serviceklassen und die Schwellenwerte entsprechend anpassen.

Die Skripts befinden sich im Verzeichnis `samples/admin_scripts` unter Ihrem Installationsverzeichnis.

Die Skripts `wlmtiersdefault.db2` und `wlmtierstimerons.db2` erstellen drei Serviceunterklassen unter einer gemeinsamen Superklasse mit immer niedrigeren Ressourceneinstellungen (von HIGH (HOCH) bis LOW (NIEDRIG)) zusammen mit Schwellenwerten vom Typ `CPUTIMEINSC`, die Aktivitäten als Reaktion auf den Verbrauch von Prozessorzeit verschieben oder neu zuordnen. Die Skripts unterscheiden sich darin, wie Aktivitäten einer Serviceklasse zugeordnet werden, wenn sie zuerst in den Datenserver eingehen. Ein drittes Skript, `wlmtiersdrop.db2`, löscht die von den beiden anderen Beispielschichtenskripts erstellten WLM-Objekte.

wlmtiersdefault.db2

Alle Aktivitäten, die in den Datenserver eingehen, werden einer Serviceunterklasse `WLM_SHORT` mit hoher Priorität ohne Unterscheidung zwischen Typen von Arbeit zugeordnet. Aktivitäten werden in der Serviceunterklasse mit hoher Priorität ausgeführt, während ihnen die höchste Priorität zugewiesen ist, sofern sie nicht den Maximalwert von Prozessorzeit übersteigen, den sie verbrauchen dürfen. Aktivitäten, die zu viel Prozessorzeit verbrauchen, werden zuerst von einer Schwellenwertaktion `REMAP ACTIVITY` einer Serviceunterklasse `WLM_MEDIUM` mit mittlerer Priorität zugeordnet. Wenn sie dann immer noch nicht ausgeführt werden können, ohne die zugewiesene Prozessorzeit in dieser Unterklasse zu überschreiten, werden sie einer Serviceunterklasse `WLM_LONG` mit niedriger Priorität zugeordnet, in der sie bis zu ihrem Abschluss ausgeführt werden. Aktivitäten, die nicht über einen Schwellenwert `CPUTIMEINSC` neu zugeordnet werden können, werden der Serviceunterklasse `WLM_MEDIUM` direkt zugeordnet, wo sie dann verbleiben.

wlmtierstimerons.db2

DML-Aktivitäten, die in den Datenserver eingehen, werden gemäß ihrem geschätzten Aufwand bewertet und einer der drei Serviceunterklassen zugeordnet. DML-Aktivitäten, die als kurz bewertet werden, werden einer Serviceunterklasse `WLM_SHORT` mit hoher Priorität zugeordnet. DML-Aktivitäten, die auf eine mittlere Länge geschätzt werden, werden einer Serviceunterklasse `WLM_MEDIUM` mit mittlerer Priorität zugeordnet und lange DML-Aktivitäten werden einer Serviceunterklasse `WLM_LONG`

zugeordnet, die die niedrigste Priorität erhält. Andere Aktivitäten (die keine DML-Aktivitäten sind) werden der Serviceunterklasse mit der höchsten Priorität zugeordnet. Während der Verarbeitung ordnet eine Schwellenwertaktion REMAP ACTIVITY die Aktivitäten, die mehr Prozessorzeit verbrauchen, als einer Serviceunterklasse zugewiesen ist, nacheinander jeweils der Serviceunterklasse mit der nächst niedrigeren Priorität zu, bis die Serviceunterklasse mit der niedrigsten Priorität erreicht ist. Hier werden die Aktivitäten dann bis zu ihrem Abschluss ausgeführt. Aktivitäten, die nicht über einen Schwellenwert CPUTIMEINSC neu zugeordnet werden können, werden der Serviceunterklasse WLM_MEDIUM direkt zugeordnet, wo sie dann verbleiben.

wlmtiersdrop.db2

Dieses Script löscht alle DB2-Workload-Manager-Serviceklassen, Schwellenwerte, Workloads, Arbeitsklassensets und Arbeitsaktionssets, die von den Scripts wlmtiersdefault.db2 und wlmtierstimerons.db2 erstellt werden.

Standardmäßig verwenden die Scripts wlmtiersdefault.db2 und wlmtierstimerons.db2 die folgenden Serviceklassen- und Schwellenwertdefinitionen:

Tabelle 40. Serviceklassen mit von den Scripts erstellten Einstellungen für CPU-Anteile und Vorablesezugriffspriorität

Serviceklasse	CPU-Anteile	Vorablesezugriffspriorität
WLM_SHORT (hohe Priorität)	6000	Hoch
WLM_MEDIUM (mittlere Priorität)	3000	Mittel
WLM_LONG (niedrige Priorität)	1000	Niedrig
Standardsystemklasse	Standard	Hoch
Standardverwaltungsklasse	Standard	Niedrig

Tabelle 41. Von den Scripts erstellte Schwellenwerte

Schwellenwert	Maximalwert für die Prozessorzeit, die in der Serviceklasse verwendet werden kann, bevor eine Neuordnung erfolgt
WLM_TIERS_REMAP_SHORT_TO_MEDIUM	30 Sekunden
WLM_TIERS_REMAP_MEDIUM_TO_LONG	30 Sekunden

Das Beispielscript wlmtiersdefault.db2 erstellt das folgende Arbeitsaktionsset und Arbeitsklassenset, das für die Zuordnung von Aktivitäten verwendet wird, die von dem Schwellenwert CPUTIMEINSC nicht direkt der Serviceunterklasse WLM_MEDIUM zugeordnet werden können. Diese Aktivitäten bleiben während ihrer gesamten Ausführung in der Serviceunterklasse WLM_MEDIUM.

Tabelle 42. Vom Beispielscript wlmtiersdefault.db2 erstelltes Arbeitsklassenset

Arbeitsklasse	Arbeitsaktion
WLM_DML_WC	Für DML-Aktivitäten, die anfangs der Serviceklasse WLM_SHORT zugeordnet wurden. Diese Aktivitäten können durch einen Schwellenwert CPUTIMEINSC neu zugeordnet werden.

Tabelle 42. Vom Beispielscript `wlmtiersdefault.db2` erstelltes Arbeitsklassenset (Forts.)

Arbeitsklasse	Arbeitsaktion
WLM_CALL_WC	Für CALL-Aktivitäten, die anfangs der Serviceklasse WLM_SHORT zugeordnet wurden. Diese Aktivitäten können durch einen Schwellenwert CPUTIMEINSC neu zugeordnet werden.
WLM_OTHER_WC	Für Aktivitäten, die nicht über einen Schwellenwert CPUTIMEINSC neu zugeordnet werden können und die der Serviceklasse WLM_MEDIUM zugeordnet sind. Diese Aktivitäten bleiben in der Serviceunterklasse WLM_MEDIUM.

Das Beispielscript `wlmtierstimerons.db2` erstellt auch das folgende Arbeitsaktionsset und Arbeitsklassenset, das für die Zuordnung von Aktivitäten entsprechend ihrem geschätzten Aufwand verwendet wird:

Tabelle 43. Vom Beispielscript `wlmtierstimerons.db2` erstelltes Arbeitsklassenset

Arbeitsklasse	Bereich des geschätzten Aufwands in Timeron und Arbeitsaktion
WLM_SHORT_DML_WC	Für DML-Aktivitäten mit einem geschätzten Aufwand von 0 bis 999 Timeron, die anfangs der Serviceklasse WLM_SHORT zugeordnet sind. Diese Aktivitäten können durch einen Schwellenwert CPUTIMEINSC neu zugeordnet werden.
WLM_MEDIUM_DML_WC	Für DML-Aktivitäten mit einem geschätzten Aufwand von 1000 bis 99.999 Timeron, die anfangs der Serviceklasse WLM_MEDIUM zugeordnet sind. Diese Aktivitäten können durch einen Schwellenwert CPUTIMEINSC neu zugeordnet werden.
WLM_LONG_DML_WC	Für DML-Aktivitäten mit einem geschätzten Aufwand von 100.000 bis unendlich Timeron, die der Serviceklasse WLM_LONG zugeordnet sind.
WLM_CALL_WC	Für CALL-Aktivitäten, die anfangs der Serviceklasse WLM_SHORT zugeordnet wurden. Diese Aktivitäten können durch einen Schwellenwert CPUTIMEINSC neu zugeordnet werden.
WLM_OTHER_WC	Für Aktivitäten, die nicht neu zugeordnet werden können und die der Serviceklasse WLM_MEDIUM zugeordnet sind.

Anpassen der Scripts an Ihre Umgebung

Wenn Sie die Beispielscripts zur Anpassung an Ihre Umgebung ändern, ist die wichtigste Einstellung, die Sie berücksichtigen müssen, der Maximalwert der Prozessorzeit, der in den einzelnen Serviceklassen verwendet werden kann. Wie viel Prozessorzeit Sie für Aktivitäten in jeder Serviceunterklasse zulassen, hängt größtenteils von Ihrer jeweiligen Umgebung ab. Damit Sie die besten Werte herausfinden können, müssen Sie überwachen, wie Aktivitäten auf Ihrem Datenserver verar-

beitet werden. Standardmäßig protokollieren sowohl das Script `wlmtiersdefault.db2` als auch das Script `wlmtierstimerons.db2` Ereignismonitordatensätze im Ereignismonitor für Schwellenwertverstöße, falls ein solcher Ereignismonitor aktiviert ist, und zwar mit der Option, bei höherem Systemaufwand den Aktivitätsereignismonitor zu aktivieren und Aktivitätsdaten zu erfassen. Wenn bei `wlmtiersdefault.db2` der Maximalwert der Prozessorzeit, die in jeder Serviceklasse verwendet werden kann, zu hoch eingestellt ist, werden die meisten Aktivitäten immer in der Klasse mit hoher Priorität gestartet und beendet, unabhängig davon, wie viel Prozessorzeit jede Aktivität tatsächlich benötigt. Wenn der Maximalwert der Prozessorzeit zu niedrig eingestellt ist, wird keine Aktivität in der Serviceklasse mit hoher Priorität beendet und jede Aktivität wird unabhängig von der Geschäftspriorität schließlich einer Serviceklasse mit mittlerer oder niedriger Priorität zugeordnet. In beiden Fällen bietet das Script keinen Vorteil für den Gesamtdurchsatz auf Ihrem Datenserver und Aktivitäten werden nicht effektiv entsprechend ihrer Geschäftspriorität behandelt. Dasselbe gilt in einem geringeren Ausmaß auch für `wlmtierstimerons.db2`, wobei Aktivitäten anfangs unterschieden werden, indem sie Serviceunterklassen basierend auf dem geschätzten Aufwand zugeordnet werden. Wenn der Maximalwert der Prozessorzeit, der in jeder Serviceklasse verwendet werden kann, nicht korrekt eingestellt ist, werden Aktivitäten nicht einer geeigneten Serviceunterklasse zugeordnet, wenn sie zu viel Prozessorzeit verbrauchen, oder sie werden trotz höherer Geschäftspriorität zu schnell neu zugeordnet.

Weitere Informationen zu bestimmten, von den Scripts erstellten DB2-Workload-Manager-Objekten und zu deren Ausführung finden Sie in den Scripts.

Beispielszenarios

In der Dokumentation sind zwei Beispiele enthalten, die zeigen, wie Sie die Beispielschichtenscripts auf Ihrem Datenserver anpassen können, um die Prioritätssteuerung nach Verweildauer zu nutzen.

Szenario: Steuern von ressourcenintensiven Business-Intelligence-Berichten mit Prioritätssteuerung nach Verweildauer

Das folgende Szenario zeigt Ihnen, wie Sie Ihren Datenserver konfigurieren können, um die Priorität aufwendiger Business-Intelligence-Berichte dynamisch zu verringern, die nicht vor Beginn der Ausführung ermittelt werden können, sodass die Systemleistung für andere Abfragen aufrechterhalten bleibt.

Das Problem: Es gibt einen Business-Intelligence-Bericht, den jeder Endbenutzer ausführen kann und der sehr aufwendig ist. Bei jeder Ausführung beeinträchtigt der Bericht die Systemleistung. Das Front-End-Tool, das zum Generieren des Berichts verwendet wird, legt keine Clientinformationen fest, anhand derer der Bericht vorab erkannt werden könnte, wodurch es möglich wäre, ihn mithilfe einer Workload einer Serviceklasse mit niedriger Priorität zuzuordnen.

Die Lösung: Sie können mithilfe der Beispielschichtenscripts `wlmtiersdefault.db2` Ihren Datenserver mit einer mehrschichtigen Konfiguration konfigurieren, die die Priorität von prozessorintensiven Aktivitäten in deren Laufzeit dynamisch herabsetzt bzw. nach Verweildauer steuert, um die Beeinträchtigung der Datenserleistung für alle anderen Benutzer zu verhindern. Nachdem eine Workload anfangs alle Arbeit einer Serviceunterklasse mit hoher Priorität zuordnet, werden die aufwendigen Berichte vom serviceklasseninternen Schwellenwert `CPUTIMEINSC` basierend auf dem verbrauchten Anteil Prozessorzeit erkannt. Wenn eine Aktivität gegen den Schwellenwert `CPUTIMEINSC` verstößt, indem sie den maximalen Anteil zulässiger Prozessorzeit nutzt, verschiebt eine Aktion `REMAP ACTIVITY` die Aktivität in eine Serviceunterklasse mit niedrigerer Priorität. Die Aktivität kann

dann als Reaktion auf die verbrauchte Prozessorzeit immer wieder neu zugeordnet werden, bis sie in der Serviceunterklasse mit der niedrigsten Priorität ausgeführt wird. Hier verbleibt die Aktivität so lange, bis sie abgeschlossen ist oder bis Sie manuell eingreifen. Andere Aktivitäten, die die Schwellenwerte nicht übersteigen, werden weiter in der Serviceunterklasse mit der hohen Priorität ausgeführt, in der sie eine höhere Agentenpriorität erhalten.

Bei jeder Neuuzuordnung einer Aktivität wird ein Ereignismonitordatensatz protokolliert, wenn Sie einen Ereignismonitor für Schwellenwertverstöße erstellt haben. Wenn Sie weitere Informationen zu neu zugeordneten Aktivitäten erfassen wollen, um weiter Untersuchungen anzustellen, können Sie der Anweisung ALTER THRESHOLD im Script wlm_tiersdefault.db2 die Klausel COLLECT ACTIVITY DATA hinzufügen. Führen Sie das Script einfach erneut aus, damit die Änderung wirksam wird.

Nachdem die Workload eine bestimmte Zeit lang ausgeführt wurde, können Sie mithilfe der Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS feststellen, wie viele Aktivitäten zwischen den Serviceunterklassen neu zugeordnet wurden:

```
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 record(s) selected.

Wenn Sie feststellen, dass keine oder nur sehr wenige Aktivitäten den Serviceunterklassen mit niedrigerer Priorität zugeordnet werden, verringern Sie den Schwellenwert CPUTIMEINSC und das von den Anweisungen ALTER THRESHOLD in den Scripts verwendete Prüfintervall, um die Zuordnung von Aktivitäten über die Serviceklassenschichten gemäß der Geschäftspriorität zu verbessern. Wenn die meisten oder fast alle Aktivitäten den Serviceunterklassen mit niedrigerer Priorität zugeordnet werden, erhöhen Sie den Schwellenwert CPUTIMEINSC und das Prüfintervall für die Anweisungen ALTER THRESHOLD, um die Ausführung von mehr Aktivitäten mit höherer Priorität zuzulassen. Nachdem Ihre Änderungen abgeschlossen sind, führen Sie das Script wlm_tiersdefault.db2 erneut aus, damit diese wirksam werden.

Szenario: Neuuzuordnung falsch zugeordneter Abfragen durch Prioritätssteuerung nach Verweildauer

Das folgende Szenario zeigt Ihnen, wie Sie Ihren Datenserver konfigurieren können, um Aktivitäten, die mehr Prozessorzeit als ursprünglich geschätzt verbrauchen, dynamisch neu zuzuordnen oder ihre Priorität nach der Verweildauer zu steuern, sodass die Systemleistung für andere Abfragen aufrechterhalten bleibt.

Das Problem: Sie haben möglicherweise aufwendige Aktivitäten basierend auf dem geschätzten SQL-Aufwand einer Serviceunterklasse mit niedrigerer Priorität zugeordnet, sodass diese Aktivitäten sich nicht auf die Leistung kürzerer Aktivitäten auswirken, die weniger aufwendig sind. Eine solche Zuordnung kann erreicht werden, indem ein Arbeitsaktionsset auf der Ebene der Servicesuperklasse definiert wird. Wenn jedoch der geschätzte SQL-Aufwand beispielsweise aufgrund von veralteten Statistikdaten falsch ist, wird eventuell eine aufwendige Aktivität fälschlicherweise einer Serviceunterklasse mit hoher Priorität zugeordnet, wo sie dann einen übermäßig großen Anteil von Ressourcen zu Lasten aller anderen Aktivitäten mit hoher Priorität verbraucht.

Die Lösung: Sie können mithilfe des Beispielschichtenscripts `wlmtierstimerons.db2` Ihren Datenserver mit einer mehrschichtigen Konfiguration konfigurieren, die eingehende Aktivitäten hinsichtlich ihres geschätzten Aufwands bewertet und sie einer von drei Serviceunterklassen mit jeweils unterschiedlichen Agentenprioritäten zuordnet. Wenn eine Aktivität zu viel Prozessorzeit verbraucht, verringert Ihr Datenserver dynamisch die Priorität der Aktivität während ihrer Laufzeit, indem er sie zwischen leistungsorientierten Schichten neu zuordnet. Dieser dynamische Prozess der Neuordnung von Aktivitäten, um ihre Priorität herabzusetzen, wird auch Prioritätssteuerung nach Verweildauer genannt.

Nachdem eine Aktivität ihrer ursprünglichen Serviceklasse zugeordnet wurde und ihre Ausführung begonnen hat, wird der serviceklasseninterne Schwellenwert `CPUTIMEINSC` vom Script verwendet, um den Anteil der Prozessorzeit zu steuern, die eine Aktivität verbrauchen kann. Wenn die Aktivität gegen den Schwellenwert verstößt, indem sie den maximalen Anteil zulässiger Prozessorzeit verbraucht, wird eine Aktion `REMAP ACTIVITY` ausgelöst, die die Aktivität in eine Serviceunterklasse mit einer niedrigeren Agentenpriorität verschiebt. Die Aktivität kann dann als Reaktion auf die verbrauchte Prozessorzeit neu zugeordnet werden, bis sie in der Serviceunterklasse mit der niedrigsten Priorität ausgeführt wird. Hier verbleibt die Aktivität dann so lange, bis sie abgeschlossen ist oder bis Sie manuell eingreifen.

Bei jeder Neuordnung einer Aktivität wird ein Ereignismonitordatensatz protokolliert. Wenn Sie weitere Informationen zu neu zugeordneten Aktivitäten erfassen wollen, um weitere Untersuchungen anzustellen, können Sie der Anweisung `ALTER THRESHOLD` im Script `wlmtiersdefault.db2` die Klausel `COLLECT ACTIVITY DATA` hinzufügen. Führen Sie das Script einfach erneut aus, damit die Änderung wirksam wird.

Nachdem die Workload eine bestimmte Zeit lang ausgeführt wurde, können Sie mithilfe der Tabellenfunktion `WLM_GET_SERVICE_SUBCLASS_STATS` feststellen, wie viele Aktivitäten zwischen den Serviceunterklassen neu zugeordnet wurden:

```
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 record(s) selected.

In diesem Szenario sollten relativ wenige Aktivitäten zwischen Serviceunterklassen neu zugeordnet werden, da Aktivitäten basierend auf dem geschätzten Aufwand anfangs fast immer der geeigneten Serviceunterklasse zugeordnet werden. Wenn Sie feststellen, dass Aktivitäten typischerweise nur in der Serviceklasse WLM_SHORT oder WLM_LONG ausgeführt werden, können Sie die Werte für den geschätzten Aufwand anpassen, die von der Anweisung ALTER WORK CLASS SET im Script verwendet werden, um die Zuordnung von Aktivitäten über Serviceklassenschichten zu verbessern, sodass kürzere Aktivitäten der Arbeitsklasse WLM_SHORT_DML_WC und längere Aktivitäten der Arbeitsklasse WLM_MEDIUM_DML_WC oder WLM_LONG_DML_WC zugeordnet werden. Wenn Sie bemerken, dass die meisten Aktivitäten neu zugeordnet werden, können Sie die Schwellenwerte erhöhen, die von den ALTER THRESHOLD-Anweisungen verwendet werden, um die anfängliche Zuordnung von Aktivitäten zu Serviceunterklassen zu verbessern. Nachdem Ihre Änderungen abgeschlossen sind, führen Sie das Script wlmstimerons.db2 erneut aus, damit diese wirksam werden.

Neuzuordnung von Aktivitäten zwischen Serviceunterklassen

Sie aktivieren die Neuzuordnung, indem Sie die Aktion REMAP ACTIVITY in den Anweisungen CREATE THRESHOLD und ALTER THRESHOLD angeben. Wenn die Neuzuordnungsaktion durch einen Schwellenwertverstoß ausgelöst wird, verschiebt sie eine Aktivität von einer Serviceunterklasse in eine andere Serviceunterklasse unter derselben Superklasse.

Vorbereitende Schritte

Damit die Neuzuordnung zu einer anderen Serviceunterklasse möglich ist, muss die Zielserviceunterklasse in derselben Servicesuperklasse vorhanden sein wie die ursprüngliche Serviceunterklasse der Aktivität. sowohl die Ziel- als auch die ursprüngliche Serviceunterklasse kann die Standardunterklasse der Superklasse sein. Die Aktion REMAP ACTIVITY kann nicht auf Serviceunterklassen unter der Standardsystemklasse, der Standardverwaltungsklasse oder der Standardbenutzerklasse angewendet werden.

Informationen zu diesem Vorgang

Die Aktion REMAP ACTIVITY verschiebt eine Aktivität in eine andere Serviceunterklasse in derselben Servicesuperklasse. Die Neuzuordnung ist verfügbar mit jedem der serviceklasseninternen Schwellenwerte wie CPUTIMEINSC und SQLROWSREADINSC. Mithilfe dieses dynamischen Prozesses der Neuzuordnung von Aktivitäten können Sie ihre Priorität über die Zeit herabsetzen. Dies wird auch Prioritätssteuerung nach Verweildauer genannt. Das Herabsetzen der Priorität einiger Aktivitäten über die Zeit kann Systemressourcen freigeben, die dann anderen, für das Unternehmen wichtigeren Aktivitäten zur Verfügung gestellt werden.

Sie können die Aktion REMAP ACTIVITY mit dem serviceklasseninternen Schwellenwert DATATAGINSC verwenden, der eine Aktivität auf der Basis der Daten, deren Zugriff erwartet wird, einer anderen DB2-Serviceunterklasse zuordnet.

Agenten, die für die Aktivität arbeiten, prüfen in regelmäßigen Abständen in jedem Member, ob gegen den Schwellenwert verstoßen wurde, ohne Koordination zwischen den Members. Wenn irgendein Agent einen Verstoß gegen einen serviceklasseninternen Schwellenwert in einem Member erkennt, löst dieser Agent die Aktion REMAP ACTIVITY für die Aktivität im Member aus und ordnet sich dann selbst der Zielserviceunterklasse zu. Anschließend gilt die Aktivität als neu zugeordnet. Alle anderen Agenten, die für die Aktivität in demselben Member arbeiten, ordnen sich der Zielserviceunterklasse neu zu, wenn sie erkennen, dass die Aktivität neu zugeordnet worden ist.

Einschränkungen

Die Zielserviceunterklasse darf nicht der ursprünglichen Serviceunterklasse entsprechen. Sie müssen zuerst eine Neuordnung in eine andere Serviceunterklasse durchführen, bevor Sie eine Neuordnung in die ursprüngliche Serviceunterklasse vornehmen können.

Wenn eine Aktivität einer inaktivierten Serviceunterklasse zugeordnet wird, wird die Aktivität behandelt, als wäre sie von der inaktivierten Serviceunterklasse zurückgewiesen worden, und eine Fehlermeldung -4714 wird an den Client zurückgegeben.

Vorgehensweise

1. Geben Sie die Aktivitäten an, die Sie über die Prioritätssteuerung nach Verweildauer steuern möchten. Eine mehrschichtige Konfiguration besteht aus Serviceunterklassen unter derselben Servicesuperklasse, für die serviceklasseninterne Schwellenwerte mit Schwellenwertaktionen REMAP ACTIVITY definiert sind. Sie können das Grundbeispiel für die Prioritätssteuerung nach Verweildauer und die Beispielszenarios als Ausgangspunkt verwenden:
 - a. „Prioritätssteuerung nach Verweildauer für laufende Aktivitäten“ auf Seite 161
 - b. „Szenario: Steuern von ressourcenintensiven Business-Intelligence-Berichten mit Prioritätssteuerung nach Verweildauer“ auf Seite 169
 - c. „Szenario: Neuordnung falsch zugeordneter Abfragen durch Prioritätssteuerung nach Verweildauer“ auf Seite 170
2. Wählen Sie die verschiedenen Serviceunterklassen aus, denen Aktivitäten zugeordnet werden sollen. Dies umfasst sowohl die Serviceunterklasse, der eine Aktivität ursprünglich zu Beginn der Aktivitätsausführung zugeordnet wird, als auch jede andere Serviceunterklasse, der die Aktivität während ihrer Laufzeit neu zugeordnet wird. Weitere Informationen zu Serviceklassen und zu deren Erstellung finden Sie in „Ressourcenzuordnung mit Serviceklassen“ auf Seite 71.
3. Erstellen oder ändern Sie Ihre Schwellenwerte zur Steuerung von Aktivitäten. Weitere Informationen zu Schwellenwerten finden Sie in „Prioritätssteuerung nach Verweildauer für laufende Aktivitäten“ auf Seite 161.
 - a. Definieren Sie Ihre serviceklasseninternen Schwellenwerte so, dass sie eine Aktion REMAP ACTIVITY umfassen, die bei einem Schwellenwertverstoß ausgelöst wird. Beachten Sie, dass ein serviceklasseninterner Schwellenwert sich nur in der Zeit auf eine Aktivität bezieht und von dieser beeinflusst wird, während der die Aktivität der zugehörigen Serviceunterklasse zuge-

ordnet ist. Nach der Neuordnung werden die betroffenen Zähler und Zeitgeber zurückgesetzt. Überlegen Sie, ob ein Schwellenwertverstoßdatensatz bei jeder Neuordnung einer Aktivität protokolliert werden soll. Diese Datensätze stellen Informationen dazu bereit, in welchen Serviceklassen eine Aktivität über eine gewisse Zeit ausgeführt wurde, und können für eine Leistungsanalyse verwendet werden. Beachten Sie aber, dass die Protokollierung von Schwellenwertverstoßdatensätzen beträchtlichen Plattenspeicherplatz verbrauchen kann, wenn die Neuordnung von Aktivitäten zwischen Serviceunterklassen häufig vorkommt.

- b. Sie können auch Schwellenwerte definieren, die Sie auf die Laufzeit der Aktivität anwenden wollen. Beachten Sie jedoch, dass nur Schwellenwerte aus der ersten Serviceunterklasse, der die Aktivität ursprünglich zugeordnet wurde, während der gesamten Laufzeit der Aktivität angewendet werden. Wenn Sie auch Schwellenwerte in einer der Serviceunterklassen definieren, denen eine Aktivität später neu zugeordnet wird, werden diese nicht angewendet.
4. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen werden Ihre Schwellenwerte zur Sicht SYSCAT.THRESHOLDS hinzugefügt.
5. Erlauben Sie Ihrem Datenserver, die gewünschten Aktivitäten mit Ihren serviceklasseninternen Schwellenwerten auszuführen und deren Fortschritt während der Laufzeit zu überwachen. Aktivitäten bleiben während der Ausführung in ihrer ursprünglichen Serviceunterklasse, solange sie nicht gegen einen serviceklasseninternen Schwellenwert verstoßen. Sobald während der Aktivitätsausführung gegen serviceklasseninterne Schwellenwerte verstoßen wird, lösen die Aktivitäten eine Aktion REMAP ACTIVITY aus, die sie dynamisch einer anderen Serviceunterklasse zuordnet. Sobald die Aktivitäten neu zugeordnet worden sind, werden sie weiter ausgeführt und werden jetzt von den Ressourcenbeschränkungen gesteuert, die Sie für diese Zielserviceunterklasse definiert haben.
6. Verbessern Sie bei Bedarf Ihre Methode für die Prioritätssteuerung nach Verweildauer, um die festgelegten Leistungsziele zu erreichen.

Beispiel

Das folgende Beispiel erzeugt eine einfache dreischichtige Konfiguration, die die Priorität von laufenden Aktivitäten über die Zeit herabsetzt bzw. nach Verweildauer steuert). Drei Serviceunterklassen unter einer einzigen Superklasse A stellen die Ausführungsumgebung bereit, in der alle Abfragen ausgeführt werden müssen. Nehmen wir an, dass die Standardbenutzerworkload eingehende Abfragen der Serviceunterklasse A1 zuordnet. Dies ist eine Unterklasse mit hoher Priorität, die die schnelle Ausführung kürzer laufender Abfragen erlaubt. Eine Serviceunterklasse A2 mit mittlerer Priorität ist für die Ausführung länger laufender Abfragen vorgesehen, allerdings mit strikterer Ressourcensteuerung. Die Serviceunterklasse A3 stellt einen Container für sehr große Abfragen bereit, deren Ausführung sehr viel Prozessorzeit beansprucht.

Über drei Schwellenwerte wird die laufende Ressourcenauslastung von Abfragen gesteuert. Eine Abfrage darf in der Serviceunterklasse A1 mit hoher Priorität nur ausgeführt werden, wenn sie in weniger als einer Minute Prozessorzeit abgeschlossen ist. Nachdem eine Minute Prozessorzeit verbraucht wurde, ordnet der Schwellenwert T1 die Aktivität automatisch der Unterklasse A2 zu, in der sie weiter ausgeführt werden kann, solange sie weniger als 10 Minuten Prozessorzeit verbraucht. Wenn die Abfrage auch nach 10 Minuten verbrauchter Prozessorzeit noch nicht abgeschlossen ist, ordnet der Schwellenwert T2 die Aktivität der Serviceunterklasse mit der niedrigsten Priorität, A3, zu. Abfragen in der Unterklasse A3 dürfen unbe-

grenzt ausgeführt werden. Es wird jedoch ein Ereignismonitoreintrag protokolliert und Aktivitätsdaten werden mit Details erfasst, wenn die verbrauchte Prozessorzeit 1 Stunde übersteigt.

```
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
```

Workload-Management-Dispatcher - Übersicht

Der DB2-Workload-Management-Dispatcher ist eine integrierte DB2-Technologie, mit der Sie der auf einem Datenbankserver ausgeführten Arbeit CPU-Ressourcen zuordnen können. CPU-Ressourcenauslastungsrechte können mithilfe von Attributen für CPU-Anteile und den CPU-Grenzwert für DB2-Serviceklassen gesteuert werden.

Übersicht

Der Dispatcher ist besonders unter Workloadbedingungen nützlich, die die System-CPU-Ressourcen stark beanspruchen, und er kann effektiv CPU-Ressourcenauslastungsrechte für die Serviceklasse mit Attributen wie die veränderlichen CPU-Anteile mit unbegrenzter (beinahe uneingeschränkter) Kapazitätsnutzung und die beiden begrenzten Attribute (feste CPU-Anteile und CPU-Grenzwerte) verwalten. Mit den CPU-Anteilen mit unbegrenzter Kapazitätsnutzung können Sie Serviceklassen für Arbeit mit hoher Priorität nicht verwendete CPU-Ressourcen zuteilen. Außerdem können Sie für die CPU-Ressourcenauslastungsrechte Kontrollmechanismen erzwingen, indem Sie begrenzte feste CPU-Anteile oder CPU-Grenzwerte verwenden, die Serviceklassen für Arbeit mit niedriger Priorität zugewiesen sind. Hierdurch begrenzen Sie ihre Auswirkung auf die Arbeit mit hoher Priorität. Unter Bedingungen, wo die System-CPU-Ressourcen weniger beansprucht werden, können Sie mindestens eins der beiden begrenzten Attribute verwenden, ohne Attribute mit unbegrenzter Kapazitätsnutzung zuweisen zu müssen, um die CPU-Ressourcenauslastungsrechte zwischen Serviceklassen, die in einer Standardumgebung mit nicht ausgelasteten CPUs ausgeführt werden, effektiv zu steuern.

Die Infrastruktur des Workload-Management-Dispatchers fungiert auf der Instanzebene des DB2-Datenbankmanagers. Der Dispatcher begrenzt die Anzahl ausführender Agenten, die dem Betriebssystem zugeteilt werden sollen, und die zulässige Ausführungsdauer der einzelnen Agenten zu jedem gegebenen Zeitpunkt. Die Anzahl ausführender Agenten, die gleichzeitig zugeteilt werden können, wird als Dispatcher-Thread-Parallelität bezeichnet.

Der DB2-Workload-Management-Dispatcher hat die folgenden Vorteile:

- Einfach zu implementieren. Seine Implementierung erfordert einen geringeren Zeit- und Arbeitsaufwand als die Implementierung eines Betriebssystem-Workload-Managers wie AIX-WLM oder Linux-WLM.
- Unterstützt veränderliche CPU-Zuordnung im Verlauf der täglichen Systemverwendung unter Berücksichtigung von Auslastungsspitzen. Diese Flexibilität wird durch permanente Zuordnungen erzielt, die immer erzwungen werden (feste CPU-Anteile und CPU-Grenzwerte), oder durch dynamische Zuordnungen, die nur erzwungen werden, wenn der Bedarf die Kapazität überschreitet (veränderliche CPU-Anteile).
- Im DB2-Datenbankmanager eigenständig, sodass Sie über die Festlegung von CPU-Ressourcenauslastungsrechten eine Workloadkontrolle erreichen können, die aufgrund ihrer Unabhängigkeit von einem Betriebssystem-Workload-Manager wie AIX-WLM oder Linux-WLM plattformübergreifend effektiv ist.
- Sie können die Betriebssystem-WLM-Produkte weiterhin als Workloadkontrolle verwenden, sie sind jedoch nicht notwendig, wenn die zusätzliche Implementierungskomplexität (z. B. Einrichten von AIX-WLM auf jedem Member) oder organisatorische Gründe (z. B. Veto des Systemadministrators) ein Hindernis sind. Sie können alternativ die Betriebssystem-WLM-Produkte für Überwachungszwecke verwenden und die Workloadkontrolle dem DB2-Workload-Management-Dispatcher überlassen.

Weitere Informationen

In diesem Abschnitt werden Informationen zu wesentlichen Verwaltungskonzepten, Tasks und Einsatzszenarios für die folgenden Bereiche geliefert:

- DB2-Workload-Management-Dispatcher
- Feste CPU-Anteile
- Veränderliche CPU-Anteile
- CPU-Grenzwert
- Minimale CPU-Auslastungsebene zum Klassifizieren der Serviceklasse als aktiv
- Dispatcher-Thread-Parallelität
- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers

Workload-Management-Dispatcher

Der Workload-Management-Dispatcher verwaltet CPU-Ressourcenauslastungsrechte für im DB2-Datenbankmanager ausgeführte Workloads. In diesem Abschnitt werden Details zu seinen Features und seiner Funktionalität geliefert.

Einführung

Mit dem integrierten DB2-Workload-Management-Dispatcher können Sie DB2-Workloads über die Konfiguration von CPU-Ressourcenauslastungsrechten verwalten, ohne Workload-Management-Software eines Drittherstellers zu benötigen. Der Dispatcher kann die DB2-spezifische CPU-Ressourcenauslastung mithilfe von Einstellungen für CPU-Anteile und den CPU-Grenzwert steuern. Veränderliche CPU-Anteile stellen eine unbegrenzte Ressourcenzuordnung dar und können verwendet werden, um zuzulassen, dass Arbeit mit hoher Priorität mehr als den zugeordneten Anteil verbraucht, sofern ungenutzte CPU-Ressourcen vorhanden sind. Feste CPU-Anteile und CPU-Grenzwerte stellen eine begrenzte CPU-Zuordnung dar, über die verhindert werden kann, dass Arbeit mit niedriger Priorität sich auf die Ausführung der Arbeit mit hoher Priorität auswirkt. CPU-Grenzwerte können verwendet werden, um die von einer bestimmten Workload verbrauchten CPU-Ressourcen

streng zu steuern, damit andere Arbeiten im System isoliert ausgeführt werden können. Feste CPU-Anteile dagegen bieten die zusätzliche Flexibilität, die Nutzung ungenutzter CPU-Ressourcen durch Arbeit mit niedriger Priorität zuzulassen, wenn keine Workload mit hoher Priorität vorhanden ist. Dieses Szenario tritt möglicherweise in Zeiten mit geringer Systemauslastung ein.

In den folgenden Abschnitten werden die Features und die Funktionalität des Workload-Management-Dispatchers ausführlicher beschrieben. Ein Abschnitt mit Szenarios veranschaulicht die Features und die Funktionalität des Workload-Management-Dispatchers anhand von Verwendungsbeispielen.

Features und Funktionalität

Die Infrastruktur des Workload-Management-Dispatchers fungiert auf der Instanzebene des DB2-Datenbankmanagers. Auf der Datenbankebene geben Sie die CPU-Ressourcenauslastungsrechte jedoch mithilfe der CPU-Anteile und CPU-Grenzwertattribute an, die über die Anweisungen CREATE SERVICE CLASS und ALTER SERVICE CLASS in eine Serviceklasse gestellt werden können.

Wie erwähnt, gelten für die Attribute für feste und veränderliche CPU-Anteile unterschiedliche Bedingungen, unter denen nicht verwendete CPU-Ressourcen, die von mindestens einer Serviceklasse freigegeben wurden, von den verbleibenden konkurrierenden Serviceklassen übernommen werden können. Die Attribute für feste und veränderliche CPU-Anteile verhalten sich beim Zurückgeben dieser übernommenen CPU-Ressourcen jedoch gleich. Wenn z. B. eine Serviceklasse ihren Anteil an CPU-Ressourcen vollständig oder teilweise freigibt, müssen die Serviceklassen, die die nicht verwendeten CPU-Ressourcen übernommen haben, die CPU-Ressourcen an die Serviceklasse zurückgeben, die versucht, ihren vollständigen Anteil abzurufen, wenn ihr Bedarf an CPU-Ressourcen steigt. Dadurch werden der abrufenden Serviceklasse die gesamten benutzerdefinierten CPU-Ressourcenauslastungsrechte neu zugewiesen, wenn der entsprechende Bedarf vorhanden ist. Ein Scheduler-Thread des DB2-Datenbankmanagers wertet die CPU-Ressourcenauslastung durch die laufende Arbeit jede Sekunde aus und nimmt Anpassungen vor, um sicherzustellen, dass der Dispatcher zeitgerecht die konfigurierten Zuordnungen für die CPU-Anteile und den CPU-Grenzwert bereitstellt.

Der Workload-Management-Dispatcher begrenzt die Anzahl ausführender Agenten, die dem Betriebssystem gleichzeitig zugeteilt werden sollen, und die zulässige Ausführungsdauer der einzelnen Agenten. Die Anzahl ausgeführter Agenten, die gleichzeitig zugeteilt werden können, wird als Dispatcher-Thread-Parallelität bezeichnet. Sie können sie über den Konfigurationsparameter `wlm_disp_concur` des Datenbankmanagers festlegen.

Der Workload-Management-Dispatcher ist standardmäßig nicht aktiviert. Sie müssen den Workload-Management-Dispatcher aktivieren, um für den DB2-Datenbankmanager spezifische CPU-Ressourcenauslastungsrechte steuern zu können. Sie aktivieren den Workload-Management-Dispatcher, indem Sie den Konfigurationsparameter `wlm_dispatcher` des Datenbankmanagers auf YES setzen. (Dieser Konfigurationsparameter ist standardmäßig auf NO gesetzt.) Vollständige Details zum Aktivieren des Workload-Management-Dispatchers finden Sie in „Aktivieren des Workload-Manager-Dispatchers“ auf Seite 198.

Wenn der Workload-Management-Dispatcher aktiviert ist, wird die gesamte Arbeit, die in den Benutzer- und Wartungserviceklassen im DB2-Datenbankmanager ausgeführt wird, vom Dispatcher gesteuert. In der Systemserviceklasse ausgeführte Arbeit kann nicht für die CPU-Ressourcensteuerung konfiguriert werden, weil den

in dieser Serviceklasse ausgeführten kritischen DB2-Subsystemen die maximale Priorität erteilt wird und diese Subsysteme nicht vom Workload-Management-Dispatcher gesteuert werden. Der Dispatcher kann CPU-Ressourcen nach seiner Aktivierung standardmäßig nur über die Einstellungen der CPU-Grenzwerte verwalten. Sie müssen den Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers auf YES setzen, um dem Dispatcher die Verwaltung von CPU-Ressourcen über CPU-Anteile und CPU-Grenzwerte zu ermöglichen.

Überwachen Sie vor oder nach der ersten Aktivierung des Workload-Management-Dispatchers Ihre Workloads, um die von diesen Workloads verbrauchten relativen CPU-Ressourcen zu ermitteln. Basierend auf der relativen Priorität der Workloads können Sie mithilfe dieser Informationen Entscheidungen hinsichtlich der Erstellung von Serviceklassen, denen Arbeit zugewiesen werden kann, des Typs von CPU-Anteilen (feste oder veränderliche), die bestimmten Serviceklassen zugewiesen werden können, der relative Menge von CPU-Anteilen, die jeder Serviceklasse zugewiesen werden sollen, und der Verwendung von CPU-Grenzwerten treffen.

Eine weitere Methode, die Ihnen hohe Flexibilität beim Steuern des DB2-Datenbankmanagerverhaltens gibt, ist die Festlegung des minimalen Prozentsatzes von CPU-Ressourcenauslastung zum Klassifizieren von Serviceklassen als aktiv. Nach der Festlegung des Konfigurationsparameters **wlm_disp_min_util** des Datenbankmanagers werden Serviceklassen, die CPU-Ressourcen verwenden, die größer gleich diesem Mindestprozentsatz sind, auf dem Host oder auf der logischen Partition als aktiv eingestuft, und die CPU-Anteile der aktiven Serviceklassen werden bei den Berechnungen der CPU-Ressourcenauslastungsrechte berücksichtigt.

Über den Workload-Management-Dispatcher können Sie die Zuordnung von CPU-Ressourcen zwischen Serviceklassen auf Ihrem System anhand des Attributs für CPU-Anteile von Serviceklassen steuern. Das Attribut für Anteile stellt das relative CPU-Zeitnutzungsrecht dar, das jede Serviceklasse im Vergleich zu anderen Serviceklassen auf dem System empfängt. Durch das Zuordnen von mehr CPU-Anteilen zu einer bestimmten Serviceklasse und von weniger CPU-Anteilen zu einer anderen Serviceklasse können Sie steuern, wie viele CPU-Ressourcen jeder Serviceklasse zugeordnet werden, und können Sie einer bestimmten Serviceklasse eine bessere Servicequalität zur Verfügung stellen als einer anderen Serviceklasse.

Nach der Entscheidung, dass der zuvor aktivierte Workload-Management-Dispatcher Ihre stark beanspruchten CPU-Ressourcen am besten über CPU-Anteile und CPU-Grenzwerte verwaltet, müssen Sie die CPU-Anteile aktivieren, indem Sie den Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers auf YES setzen. Die Standardeinstellung für diesen Parameter ist N0. Sie können CPU-Anteile und CPU-Grenzwerte über die Anweisungen CREATE SERVICE CLASS und ALTER SERVICE CLASS festlegen und anpassen.

Der Workload-Management-Dispatcher berechnet auf der Basis der Anzahl CPU-Anteile, die einer Serviceklasse zugewiesen sind, den Anteil der CPU-Ressourcen, die jede Serviceklasse verwenden darf. Sie können den Anteil der CPU-Ressourcen, zu denen jede Servicesuperklasse berechtigt ist, mit der folgenden Formel ermitteln, die die Anzahl CPU-Anteile einer bestimmten Servicesuperklasse in einen Prozentsatz von vom Workload-Management-Dispatcher zugeordneten CPU-Ressourcen konvertiert:

$$\% \text{ CPU(Superklasse)} = (\text{Anzahl Superklassenanteile} / \text{Gesamtzahl Anteile aller aktiven Superklassen}) \times 100$$

Sie können den Anteil der CPU-Ressourcen, zu denen jede Serviceunterklasse berechtigt ist, mit der folgenden Formel ermitteln, die die Anzahl CPU-Anteile einer

bestimmten Serviceunterklasse in einen Prozentsatz von vom Workload-Management-Dispatcher zugeordneten CPU-Ressourcen konvertiert:

$$\% \text{ CPU(Unterklasse)} = \% \text{ CPU(Superklasse)} \times (\text{Anzahl Unterklassenanteile} / \text{Gesamtzahl Anteile aller aktiven Unterklassen in der Superklasse})$$

Anmerkung: Die Gesamtzahl (fester und veränderlicher) CPU-Anteile aller aktiven Superklassen werden in allen Datenbanken und Members auf dem Host bzw. der logischen Partition gezählt.

Anschauliche Beispiele für Berechnungen von CPU-Anteilen finden Sie im Abschnitt „Einsatzszenarios“.

Zusammenfassung der CPU-Ressourcenplanungsfunktionalität

In der folgenden Tabelle werden die Features der verschiedenen CPU-Ressourcensteuerattribute zusammengefasst, die für vom Workload-Management-Dispatcher verwaltete DB2-Serviceklassen verfügbar sind.

Tabelle 44. Zusammenfassung der CPU-Ressourcenplanungsfunktionalität des Workload-Management-Dispatchers

CPU-Ressourcensteuerattribut	Verwendung von inaktiven CPU-Ressourcen im selben Kontext	Begrenzungen von CPU-Ressourcenauslastungsrechten sind relativ zu anderen aktiven Serviceklassen im selben Kontext	Begrenzungen von CPU-Ressourcenauslastungsrechten sind auf der Basis der System-CPU-Gesamtkapazität fixiert	Steuerung der CPU-Belegung bei hoher CPU-Ressourcenauslastung	Steuerung der CPU-Belegung bei niedriger CPU-Ressourcenauslastung	Verwendungshinweise
Veränderliche Anteile	J	J	N	J	N	Ideal für Arbeit mit hoher Priorität zum Nutzen von auf dem Host oder der logischen Partition verfügbaren freien CPU-Ressourcen geeignet, um den Workloadfortschritt zu maximieren
Feste Anteile	N	J	N	J	J	Ideal dafür geeignet, dass Arbeit nicht mit Arbeit mit höherer Priorität kollidiert, die auf dem Host oder der logischen Partition mit unterschiedlicher Kapazität und Intensität ausgeführt wird

Tabelle 44. Zusammenfassung der CPU-Ressourcenplanungsfunktionalität des Workload-Management-Dispatchers (Forts.)

CPU-Ressourcensteuerattribut	Verwendung von inaktiven CPU-Ressourcen im selben Kontext	Begrenzungen von CPU-Ressourcen-auslastungsrechten sind relativ zu anderen aktiven Serviceklassen im selben Kontext	Begrenzungen von CPU-Ressourcen-auslastungsrechten sind auf der Basis der System-CPU-Gesamtkapazität fixiert	Steuerung der CPU-Belegung bei hoher CPU-Ressourcen-auslastung	Steuerung der CPU-Belegung bei niedriger CPU-Ressourcen-auslastung	Verwendungshinweise
CPU-Grenzwerte	N	N	J	J	J	Ideal für Sandboxing und das Erzwingen konsistenter fester CPU-Ressourcenzuordnungen zwischen Serviceklassen oder für das Begrenzen des CPU-Ressourcenverbrauchs der Serviceklasse unabhängig von anderer Arbeit auf dem Host oder der logischen Partition geeignet

Auswirkungen von Arbeit außerhalb der Steuerung durch den Workload-Management-Dispatcher

Für jede DB2-Instanz ist ein Workload-Management-Dispatcher vorhanden. Er kann nur DB2-Arbeit in seiner eigenen Instanz direkt erkennen und verwalten. Jede andere Arbeit, die auf demselben Host oder in derselben logischen Partition ausgeführt wird, untersteht nicht der direkten Kontrolle des Dispatchers. Die willkürliche Natur des CPU-Ressourcenbedarfs an Arbeit, die außerhalb des DB2-Datenbankmanagers ausgeführt wird, bedeutet, dass die vom Dispatcher verwaltete Arbeit jederzeit mit Arbeit konkurrieren muss, die außerhalb der eigenen DB2-Instanz und außerhalb der direkten Kontrolle des Dispatchers ausgeführt wird. Hingegen können die CPU-Grenzwerte und relativen Anteile von CPU-Ressourcen bei in DB2 ausgeführter Arbeit erzwungen werden. Die folgende Arbeit wird vom Workload-Management-Dispatcher weder erkannt noch verwaltet:

- Arbeit, die von Anwendungen oder Middlewareprodukten ausgeführt wird, die einen Teil ihrer Arbeit außerhalb des DB2-Datenbankmanagers ausführen
- Arbeit, die von Entitäten ausgeführt wird, die in der DB2-Systemserviceklasse ausgeführt werden
- Von anderen DB2-Instanzen ausgeführte Arbeit
- Nicht-DB2-Arbeit, die in Prozessen im abgeschirmten Modus wie abgeschirmte gespeicherte Prozeduren ausgeführt wird

- In vertrauenswürdigen Routinen ausgeführte Nicht-DB2-Arbeit. Vertrauenswürdige Prozeduren und benutzerdefinierte Funktionen werden bei der Dispatcher-CPU-Ressourcenplanung berücksichtigt, der Dispatcher kann die CPU-Auslastung jedoch nicht begrenzen, wenn eine vertrauenswürdige Routine über lange Zeiträume im Benutzercode ausgeführt wird.

System

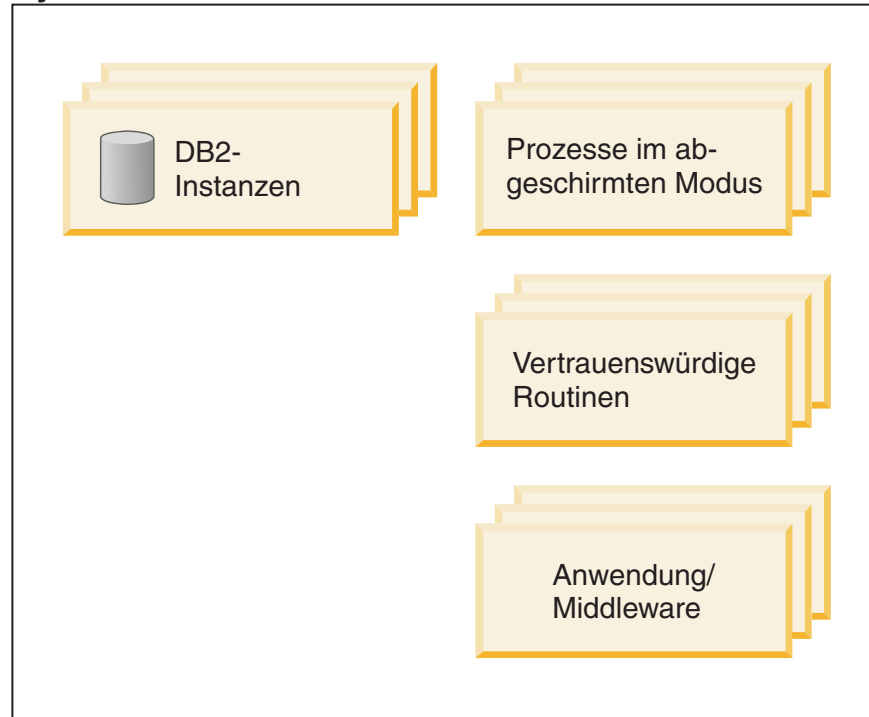


Abbildung 22. Beispielsystemdiagramm

Angenommen, Sie haben einen Host oder eine logische Partition, der bzw. die nur den DB2-Datenbankmanager ausführt. Sie haben eine Datenbank mit den Serviceklassen A und B erstellt und jeder Serviceklasse 50 % der CPU-Ressourcen zugewiesen. Wenn der Host oder die logische Partition die CPU-Ressourcen jederzeit voll nutzt, ähneln Ihre zeitorientierten Messwerte der CPU-Gesamtauslastung (%) den in Anzeige A in Abb. 23 auf Seite 182 angezeigten Messwerten.

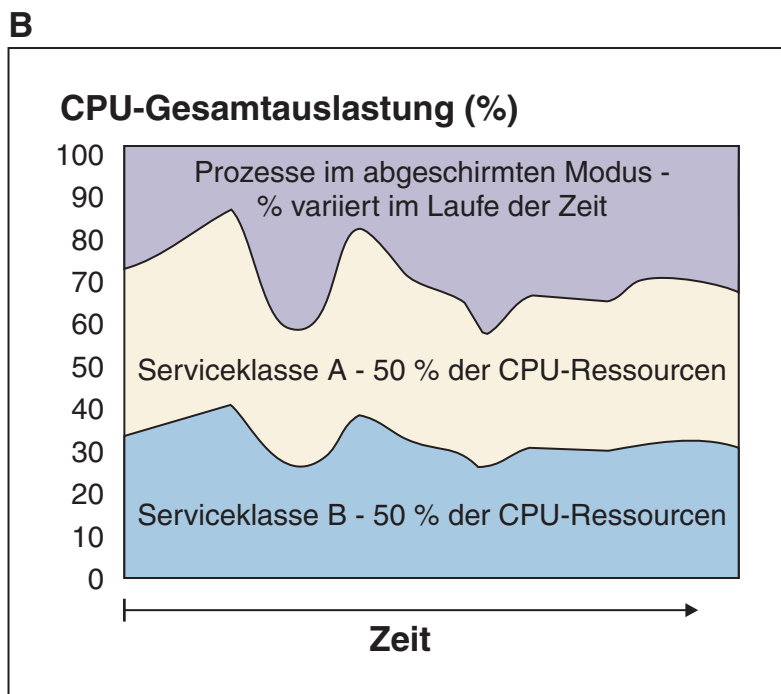
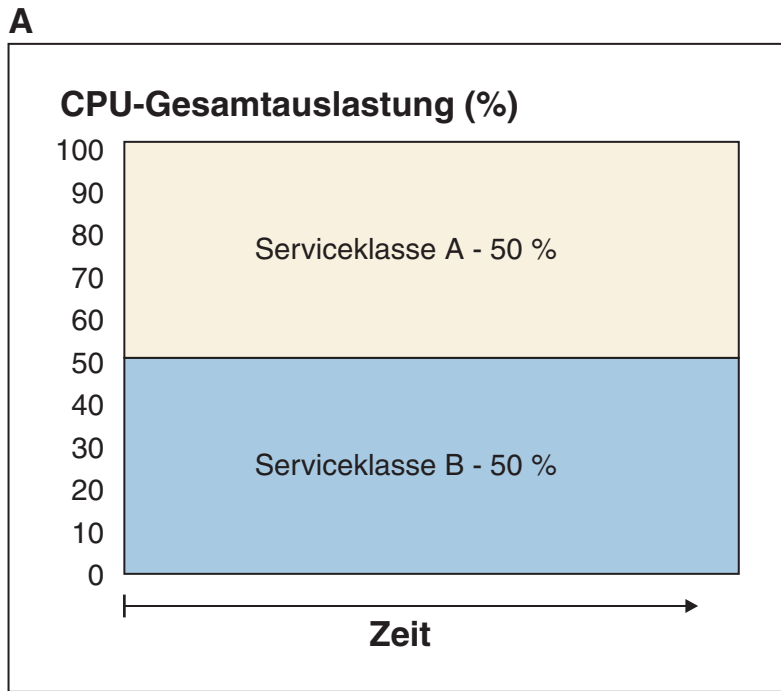


Abbildung 23. Prozentsatz der CPU-Gesamtauslastung

Nun soll betrachtet werden, was passiert, wenn auf Ihrem Host oder Ihrer logischen Partition Prozesse im abgeschirmten Modus ausgeführt werden. Diese Prozesse werden vom Workload-Management-Dispatcher nicht gesteuert. Daher kann der Anteil der verwendeten CPU-Ressourcen willkürlich variieren. Die verbleibenden CPU-Ressourcen, die der Dispatcher verwalten kann, können jedoch weiterhin zu jeweils 50 % auf die Serviceklassen A und B aufgeteilt werden, wie in der Anzeige B gezeigt wird.

Interaktionen zwischen Dispatcher- und anderen Serviceklasseneinstellungen

In der folgenden Tabelle wird zusammengefasst, wie die die Einstellungen des Workload-Management-Dispatchers für eine Serviceklasse (feste und veränderliche CPU-Anteile sowie CPU-Grenzwert) mit anderen Einstellungen für dieselbe Serviceklasse interagieren.

Tabelle 45. Interaktionen zwischen Dispatcher- und anderen Serviceklasseneinstellungen

Dispatcher-Serviceklasseneinstellung	Andere Serviceklasseneinstellung	Interaktion
Feste und veränderliche CPU-Anteile sowie CPU-Grenzwert	Agentenpriorität Anmerkung: Das Zuordnen der einzelnen DB2-Serviceklassen zu einer Agentenpriorität, die die relative Betriebssystempriorität von Agenten in der Serviceklasse steuert, wird nicht weiter unterstützt und wird in einem zukünftigen Release möglicherweise entfernt. Steuern Sie die CPU-Belegung nicht über die Agentenpriorität, sondern über den Workload-Management-Dispatcher.	Wenn der Konfigurationsparameter wlm_dispatcher des Datenbankmanagers auf YES gesetzt ist, wird die Agentenpriorität ignoriert und sind CPU-Grenzwerte aktiviert. Wenn der Parameter wlm_dispatcher auf NO gesetzt ist, werden die CPU-Grenzwerte ignoriert und überschreiben sie nicht mehr die Agentenpriorität.
Feste und veränderliche CPU-Anteile sowie CPU-Grenzwert	Abgehender Korrelator (Betriebssystem-WLM-Integration)	Feste und veränderliche CPU-Anteile, der CPU-Grenzwert und der Betriebssystem-Workload-Manager können zur selben Zeit aktiv sein.
Feste und veränderliche CPU-Anteile sowie CPU-Grenzwert	Pufferpoolpriorität	Feste und veränderliche CPU-Anteile, der CPU-Grenzwert und die Pufferpoolpriorität können zur selben Zeit aktiv sein.
Feste und veränderliche CPU-Anteile sowie CPU-Grenzwert	Vorablesezugriffspriorität	Feste und veränderliche CPU-Anteile, der CPU-Grenzwert und die Vorablesepriorität können zur selben Zeit aktiv sein.

Interaktion zwischen Dispatchereinstellungen und Schwellenwerten mit einer Aktion REMAP

Einige Grenzwerte für den Workload-Management unterstützen eine Aktion REMAP ACTIVITY zum Verschieben ausgeführter Arbeit von einer Serviceunterklasse in eine andere Serviceunterklasse in derselben Servicesuperklasse. Wenn Arbeit in eine Serviceunterklasse mit einem anderen CPU-Anteil oder CPU-Grenzwert als in der Ausgangsserviceunterklasse verschoben wird, werden die Einstellungen für den CPU-Anteil und CPU-Grenzwert der Zielserviceunterklasse angewendet.

Vergleich zwischen Workload-Management-Dispatcher und Workload-Management-Schwellenwerten für gemeinsamen Zugriff

Sie können die Workloadplanung und die Ressourcenauslastung zwar mit dem Workload-Management-Dispatcher und den Schwellenwerten für gemeinsamen Zugriff steuern, jedoch hilft Ihnen die Kenntnis des unterschiedlichen Ausmaßes der Workloadkontrolle und der Vorteile dieser beiden Methoden wahrscheinlich dabei, zu entscheiden, wann sie getrennt oder zusammen in Ihrer Umgebung eingesetzt werden sollen. In der folgenden Tabelle werden ihre Fähigkeiten zum Steuern von CPU-Ressourcen verglichen.

Tabelle 46. Vergleich der Fähigkeiten zum Steuern von CPU-Ressourcen zwischen Workload-Management-Dispatcher und Workload-Management-Schwellenwerten für gemeinsamen Zugriff

Workload-Management-Dispatcher	Schwellenwerte für gemeinsamen Zugriff für das DB2-Workload-Management
Sie können CPU-Zuordnungen im DB2-Datenbankmanager direkt steuern, nachdem eine Aktivität CPU-Ressourcen zu verarbeiten anfängt.	Sie können die CPU-Belegung und den Verbrauch anderer Ressourcen wie z. B. Speicher indirekt steuern, indem Sie die Anzahl der Verbindungen oder Aktivitäten beschränken, die zur selben Zeit ausgeführt werden dürfen.
Sie können relative Zuordnungen von veränderlichen und festen CPU-Ressourcenanteilen zu Serviceklassen zusätzlich zu den absoluten Grenzwerten der CPU-Belegung durch eine einzelne Serviceklasse konfigurieren.	Sie können einen absoluten Grenzwert für den gemeinsamen Zugriff von ausführenden Aktivitäten oder geöffneten Verbindungen für eine Serviceklasse definieren, Sie können jedoch keine Art von relativer CPU-Zuordnung angeben.

Vergleich zwischen Workload-Management-Dispatcher und Betriebssystem-Workload-Managern (AIX-WLM und Linux-WLM)

Der Workload-Management-Dispatcher kann in bestimmten Situationen zusammen mit einem Betriebssystem-Workload-Manager (AIX-WLM oder Linux-WLM) verwendet werden, um für noch größere Steuerung zu sorgen. Es folgen einige Beispiele für Umstände, unter denen der Workload-Management-Dispatcher und ein Betriebssystem-Workload-Manager zusammenarbeiten können:

- Betriebssystem-Workload-Manager verfügen häufig über Überwachungsfunktionen auf der Betriebssystemebene, die die Überwachungsfunktionen des Workload-Management-Dispatchers ergänzen.
- Betriebssystem-Workload-Manager können alle Anwendungen auf dem System, einschließlich das Datenbankmanagementsystem, verwalten. Der Workload-Management-Dispatcher kann die DB2-spezifische Zuteilung übernehmen, während der Betriebssystem-Workload-Manager alle anderen Anwendungen im System verwaltet.
- Versuchen Sie nicht, CPU-Ressourcenzuordnungen zu einzelnen Serviceklassen gleichzeitig mit dem Betriebssystem-Workload-Manager und dem Workload-Management-Dispatcher zu steuern. Sie können für Überwachungszwecke einen Betriebssystem-Workload-Manager auf der Serviceklassenebene integrieren, wenn der Workload-Management-Dispatcher aktiviert ist. Sie können außerdem einen Betriebssystem-Workload-Manager integrieren, um externe Anwendungen und/oder die einer DB2-Instanz zugeordneten CPU-Gesamtressourcen zu steuern.

Tabelle 47. Vergleich der Funktionalität zwischen Workload-Management-Dispatcher und Betriebssystem-Workload-Manager

Workload-Management-Dispatcher	Betriebssystem-Workload-Manager
Eine einzelne Lösung, die plattformübergreifend funktioniert.	Ein Betriebssystem-Workload-Manager ist betriebssystemspezifisch. Jeder Workload-Manager hat seine eigene Benutzerschnittstelle und seine eigenen Einschränkungen.
Sie können CPU-Zuordnungen im DB2-Datenbankmanager ohne betriebssystemspezifische Integration und/oder Einrichtung steuern. Ein Datenbankadministrator kann diese Einstellungen verwalten, ohne Rootzugriff auf das System zu benötigen.	Ein Betriebssystem-Workload-Manager benötigt häufig Rootberechtigungen für die Konfiguration.
Sie können datenbankspezifisches Wissen nutzen, wenn Sie Entscheidungen hinsichtlich der CPU-Ressourcenplanung treffen, die mit Einrichtungen auf der Betriebssystemebene nicht möglich wären.	Ein Betriebssystem-Workload-Manager behandelt den DB2-Datenbankmanager wie eine andere Anwendung und kann daher für Entscheidungen hinsichtlich der CPU-Ressourcenplanung kein datenbankspezifisches Wissen nutzen.
In Umgebungen mit Datenbanken mit mehreren Mitgliedern können alle Mitglieder mit einem Befehl konfiguriert werden. Dies bedeutet allerdings, dass die einzelnen Mitglieder nicht unterschiedlich konfiguriert sein können.	In Umgebungen mit mehreren Mitgliedern muss für jedes Mitglied ein Betriebssystem-Workload-Manager konfiguriert sein. Jedes einzelne Mitglied kann unterschiedlich konfiguriert sein.
Steuert die Verwendung von CPU-Ressourcen nur über DB2-Arbeit, die in den Benutzer- und Wartungsserviceklassen im Bereich einer einzelnen DB2-Instanz ausgeführt wird. Der Dispatcher kann die Verwendung von CPU-Ressourcen durch Anwendungen, die außerhalb der DB2-Instanz ausgeführt werden, nicht steuern.	Ein Betriebssystem-Workload-Manager kann CPU-Ressourcen auf dem gesamten System vollständig steuern.
CPU-Ressourcen werden über veränderliche und feste CPU-Anteile gesteuert.	Betriebssystem-WLM-Lösungen unter AIX und Linux stellen zurzeit kein Äquivalent der festen DB2-CPU-Anteile bereit.

Vergleich zwischen Workload-Management-Dispatcher und Agentenpriorität

Die Workload kann über die Agentenpriorität gesteuert werden. In der folgenden Tabelle wird die CPU-Ressourcensteuerung über die Agentenpriorität mit der CPU-Ressourcensteuerung über den Workload-Management-Dispatcher verglichen.

Tabelle 48. Vergleich der Fähigkeiten zum Steuern von CPU-Ressourcen von Workload-Management-Dispatcher und Agentenpriorität

Workload-Management-Dispatcher	Agentenpriorität
Eine Methode für konsistente und genaue Zuordnung oder die Definition von Grenzwerten für die Verwendung von CPU-Ressourcen zwischen Serviceklassen.	Die Agentenpriorität kann keine Grenzwerte für die Verwendung von CPU-Ressourcen durch eine Serviceklasse definieren. Die Agentenpriorität kann nur eine Serviceklassenpriorität ohne differenzierte Steuerung der CPU-Ressourcenauslastungsrechte zuweisen.

Anmerkung: Das Zuordnen der einzelnen DB2-Serviceklassen zu einer Agentenpriorität, die die relative Betriebssystempriorität von Agenten in der Serviceklasse steuert, wird nicht weiter unterstützt und wird in einem zukünftigen Release möglicherweise entfernt. Steuern Sie die CPU-Belegung nicht über die Agentenpriorität, sondern über den Workload-Management-Dispatcher.

Einsatzszenarios

Szenario 1: Berechnungen der CPU-Nutzungsrechte

Abb. 24 auf Seite 187 zeigt den konfigurierten Datenserver. Es wird zum Veranschaulichen der Basiskonzepte davon ausgegangen, dass in den Standardbenutzer-, Wartungs- und Systemserviceklassen unbedeutende Arbeit ausgeführt wird. Am Anfang des Szenarios wird in den Benutzerservicesuperklassen A, B und C genügend Arbeit ausgeführt, um die vollständigen CPU-Ressourcenauslastungsrechte für diese Serviceklassen zu verwenden. Im restlichen Szenario gibt es nur eine DB2-Instanz mit einer Datenbank und nur einem logischen Member auf diesem Host oder dieser logischen Partition.

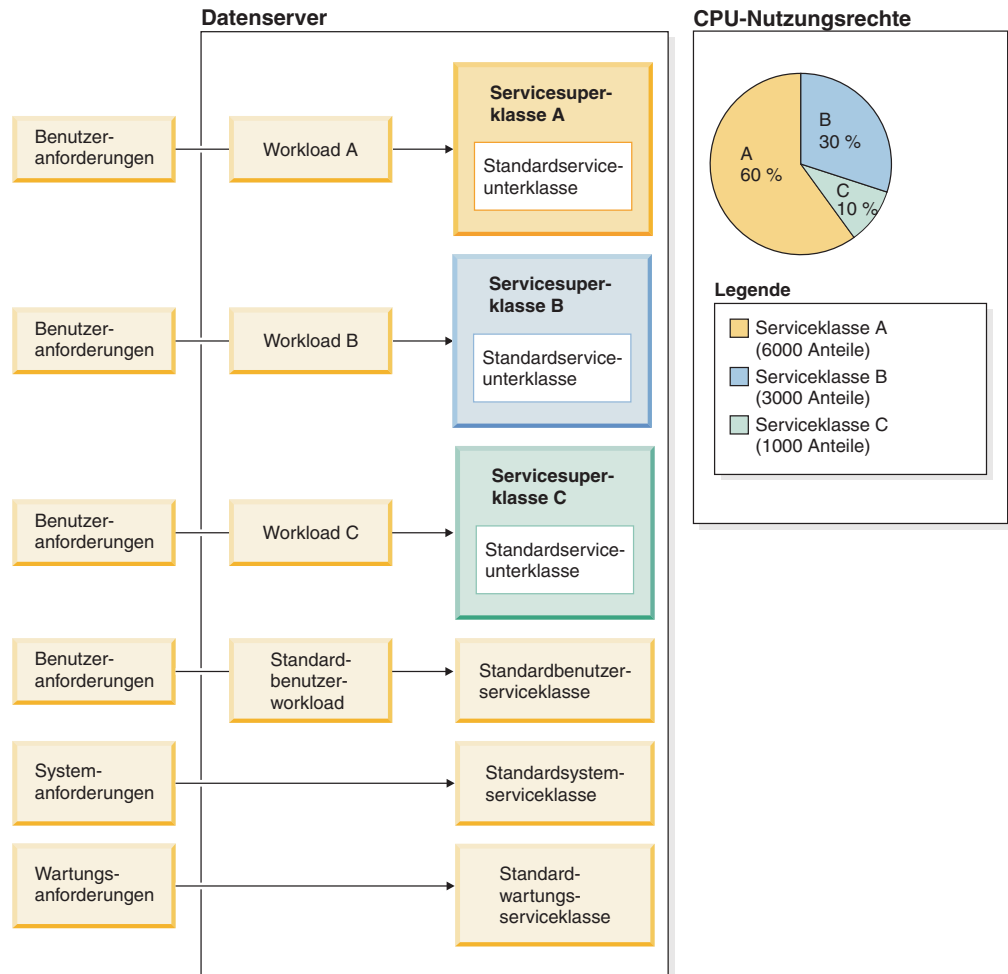


Abbildung 24. Datenserverkonfiguration

Wie in Abb. 24 gezeigt, wird der Prozentsatz der CPU-Nutzungsrechte für die einzelnen Servicesuperklassen wie folgt berechnet:

- Die Servicesuperklasse A hat 60 %.

$$(6000 / (6000 + 3000 + 1000)) \times 100$$
- Die Servicesuperklasse B hat 30 %.

$$(3000 / (6000 + 3000 + 1000)) \times 100$$
- Die Servicesuperklasse C hat 10 %.

$$(1000 / (6000 + 3000 + 1000)) \times 100$$

CPU-Anteile können auf der Ebene der Servicesuperklasse und der Serviceunterklasse zugewiesen werden. Wenn Sie bei der Erstellung einer Servicesuperklasse oder Serviceunterklasse keine CPU-Anteile angeben, weist der DB2-Datenbankmanager diesen Serviceklassen standardmäßig 1.000 feste CPU-Anteile zu. Auf der Superklassenebene stellt der Wert der CPU-Anteile dar, wie die System-CPU-Ressourcen unter DB2-Servicesuperklassen aufgeteilt werden. Auf der Unterklassenebene stellt der Wert der CPU-Anteile die Aufteilung der für eine bestimmte Superklasse verfügbaren CPU-Ressourcen unter den aktiven Unterklassen dar.

Szenario 2: Workload-Management-Dispatcher fungiert auf der Instanzebene; Serviceklassen fungieren auf der Datenbankebene

Der Workload-Management-Dispatcher fungiert auf der Instanzebene, die Konfigurationen (CPU-Anteile und CPU-Grenzwerte), über die der Dispatcher CPU-Ressourcen zuordnet, befinden sich jedoch in einzelnen Servicesuperklassen und Serviceunterklassen in einzelnen Datenbanken. Wenn in derselben Instanz mehrere Datenbanken vorhanden sind, bestimmt die Summe der CPU-Anteilswerte in Servicesuperklassen in jeder Datenbank die relative Zuordnung der CPU-Ressourcen, die jede Datenbank empfängt. Beispiel: In derselben Instanz sind die beiden Datenbanken A und B vorhanden und für jede Datenbank sind zwei Servicesuperklassen (SC1 und SC2) definiert, wie in der Anzeige A in Abb. 25 auf Seite 189 gezeigt wird.

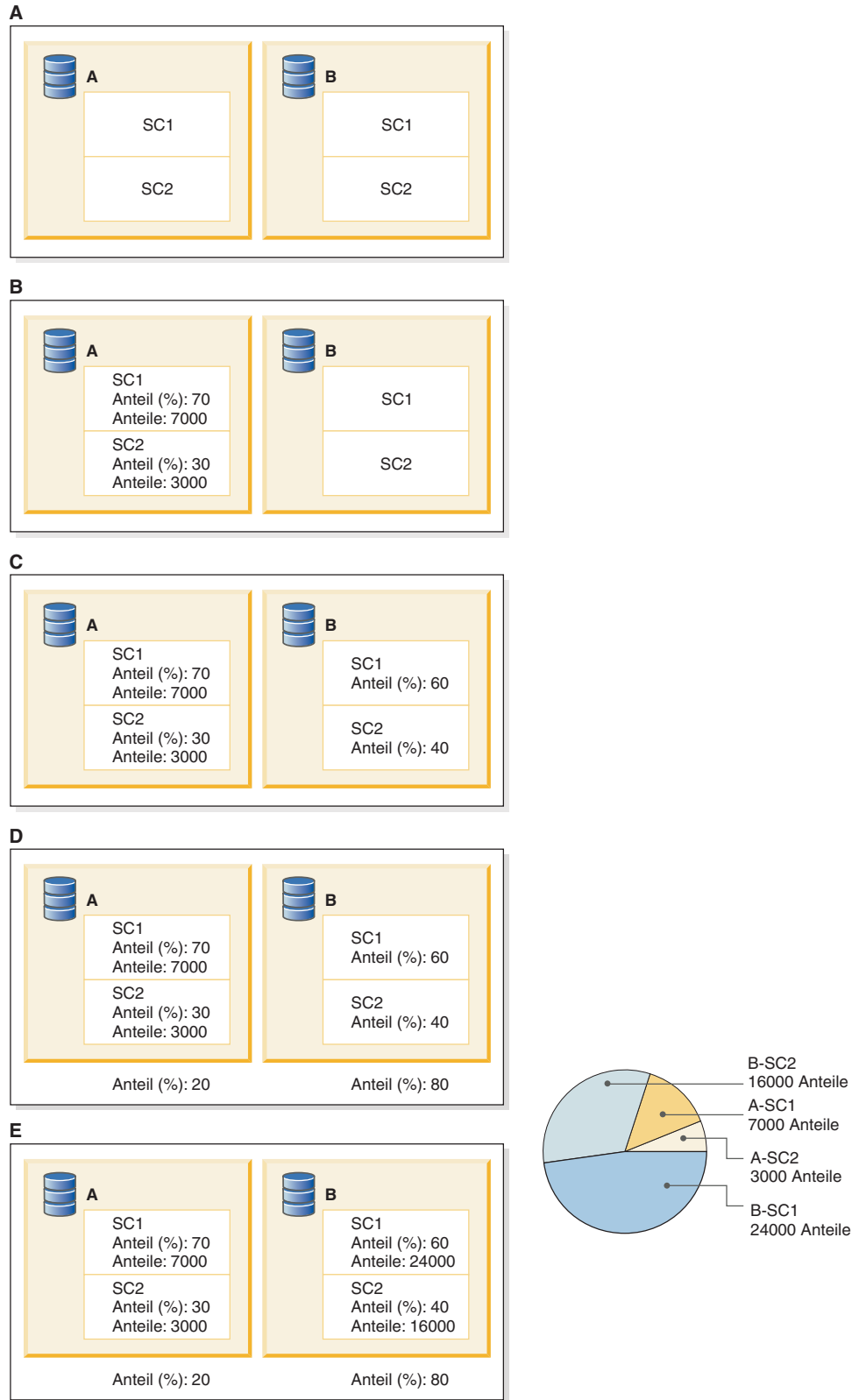


Abbildung 25. Zuordnungen von CPU-Anteilen zu Serviceklassen für zwei Datenbanken in einer Instanz

Bei diesem Beispielszenario sollen die CPU-Ressourcen in der Datenbank A in 70 % und 30 % aufgeteilt werden. Diese Datenbank ist über 7.000 veränderliche CPU-Anteile für SC1 und 3.000 veränderliche CPU-Anteile für SC2 konfiguriert, wie in der Anzeige B gezeigt wird. Die CPU-Ressourcen in der Datenbank B sollen in 60 % für SC1 und 40 % für SC2 aufgeteilt werden, wie in der Anzeige C gezeigt wird.

Vor der Entscheidung, wie diese Aufteilung in 60 % und 40 % in CPU-Anteile in den beiden Servicesuperklassen für die Datenbank B umgesetzt wird, muss entschieden werden, was die erforderlichen relativen CPU-Ressourcenauslastungsrechte zwischen den Datenbanken A und B sind. Die in der Datenbank B ausgeführte Arbeit ist wesentlich wichtiger als die in der Datenbank A ausgeführte Arbeit. Daher soll die Datenbank B ein CPU-Nutzungsrecht von 80 % und die Datenbank A ein CPU-Nutzungsrecht von 20 % haben.

Die Summe der CPU-Anteile für die Datenbank A lautet 10.000. Diese 10.000 CPU-Anteile stellen 20 % der CPU-Gesamtanteile für beide Datenbanken dar, d. h., dass insgesamt 50.000 CPU-Anteile benötigt werden, um 100 % der CPU-Gesamtressourcen darzustellen. Daher müssen der Datenbank B 80 % dieser 50.000 CPU-Anteile zugewiesen werden, also 40.000 CPU-Anteile. Die Aufteilung in 60 % und 40 % zwischen den beiden Superklassen der Datenbank B wird erzielt, indem die Servicesuperklasse SC1 60 % der 40.000 CPU-Gesamtanteile empfängt, die in der Datenbank B zugewiesen sind, also 24.000 CPU-Anteile. Die Servicesuperklasse SC2 empfängt 40 % der 40.000 CPU-Gesamtanteile, die in der Datenbank B zugewiesen sind, also 16.000 CPU-Anteile.

Das Kreisdiagramm zeigt die Zuweisungen der CPU-Anteile, die den gewünschten Prozentsätzen der CPU-Ressourcenzuordnungen entsprechen.

Szenario 3: Bevorzugen einer Workload hoher CPU-Auslastung und Schützen der bevorzugten Workload bei niedriger CPU-Auslastung

Bei diesem Beispieleinsatzszenario ist das Ziel die Verwaltung der Servicequalität für mehrere Workloads bei gleichzeitiger Maximierung der CPU-Auslastung auf dem System. Die Workloads in diesem Szenario konkurrieren jedoch nicht nur um die CPU-Ressourcen, sondern auch um andere Ressourcen. Die Zuordnung von nicht verwendeten CPU-Ressourcen zu einigen Workloads, wenn konkurrierende Workloads, die ihr vollständiges CPU-Nutzungsrecht nicht ausschöpfen, noch vorhanden sind, kann aufgrund von CPU-Cache-Konkurrenzsituationen, Konkurrenzsituationen bei Ein-/Ausgabe oder anderen sekundären Auswirkungen zu einer Leistungsminderung führen .

In diesem Szenario muss die bevorzugte Workload vor nicht bevorzugten Workloads geschützt werden, obwohl es möglich ist, dass die CPU-Ressourcen nicht vollständig verwendet werden. Nicht-ausgelastete CPU-Ressourcen kann es geben, wenn Workloads mit niedriger Priorität begrenzt sind und die bevorzugte Workload aktiv ist, aber nicht ihre gesamten Ressourcenauslastungsrechte nutzt. Die bevorzugte Workload kann geschützt werden, indem den nicht bevorzugten Workloads feste CPU-Anteile und der bevorzugten Workload veränderliche CPU-Anteile zugewiesen werden.

In diesem Beispiel sollen die beiden Workloads 'Favored' und 'Other' verwendet werden. Die Workload 'Favored' soll über 70 % der CPU-Ressourcen verfügen, wenn sie zur selben Zeit ausgeführt wird wie die Workload 'Other', und die Workload 'Favored' soll über 100 % verfügen, wenn die Workload 'Other' nicht vorhanden ist. Die Workload 'Other' soll über 30 % der CPU-Ressourcen verfügen, wenn sie zur selben Zeit ausgeführt wird wie die Workload 'Favored', und die Workload 'Other' soll über 100 % verfügen, wenn die Workload 'Favored' nicht vorhanden ist. Dieses Szenario ähnelt dem Szenario 3 mit der Ausnahme, dass die CPU-Gesamtauslastung gelegentlich unter 100 % fällt.

Die folgenden Schritte zeigen, wie das DB2-Workload-Management konfiguriert werden muss, um die oben genannten Ziele zu erreichen:

1. Erstellen Sie die Servicesuperklassen 'Favored' und 'Other' und weisen Sie ihnen CPU-Anteile zu, sodass ein CPU-Nutzungsverhältnis von 70 zu 30 erzielt wird. CPU-Anteile können als 7 und 3, 14 und 6 oder jede beliebige andere Kombination angegeben werden, die zum CPU-Nutzungsverhältnis von 70 % zu 30 % führt. Im folgenden Beispiel werden der Serviceklasse 'Favored' 7.000 veränderliche CPU-Anteile und der Serviceklasse 'Other' 3.000 feste CPU-Anteile zugewiesen:

```
create service class favored soft cpu shares 7000
create service class other hard cpu shares 3000
```

2. Erstellen Sie die Workloads 'Favored' und 'Other' so wie im folgenden Beispiel gezeigt:

```
create workload favored appl_name('favoredapp') service class favored
create workload other appl_name('otherapp') service class other
```

Das System sieht aufgrund dessen wie folgt aus:

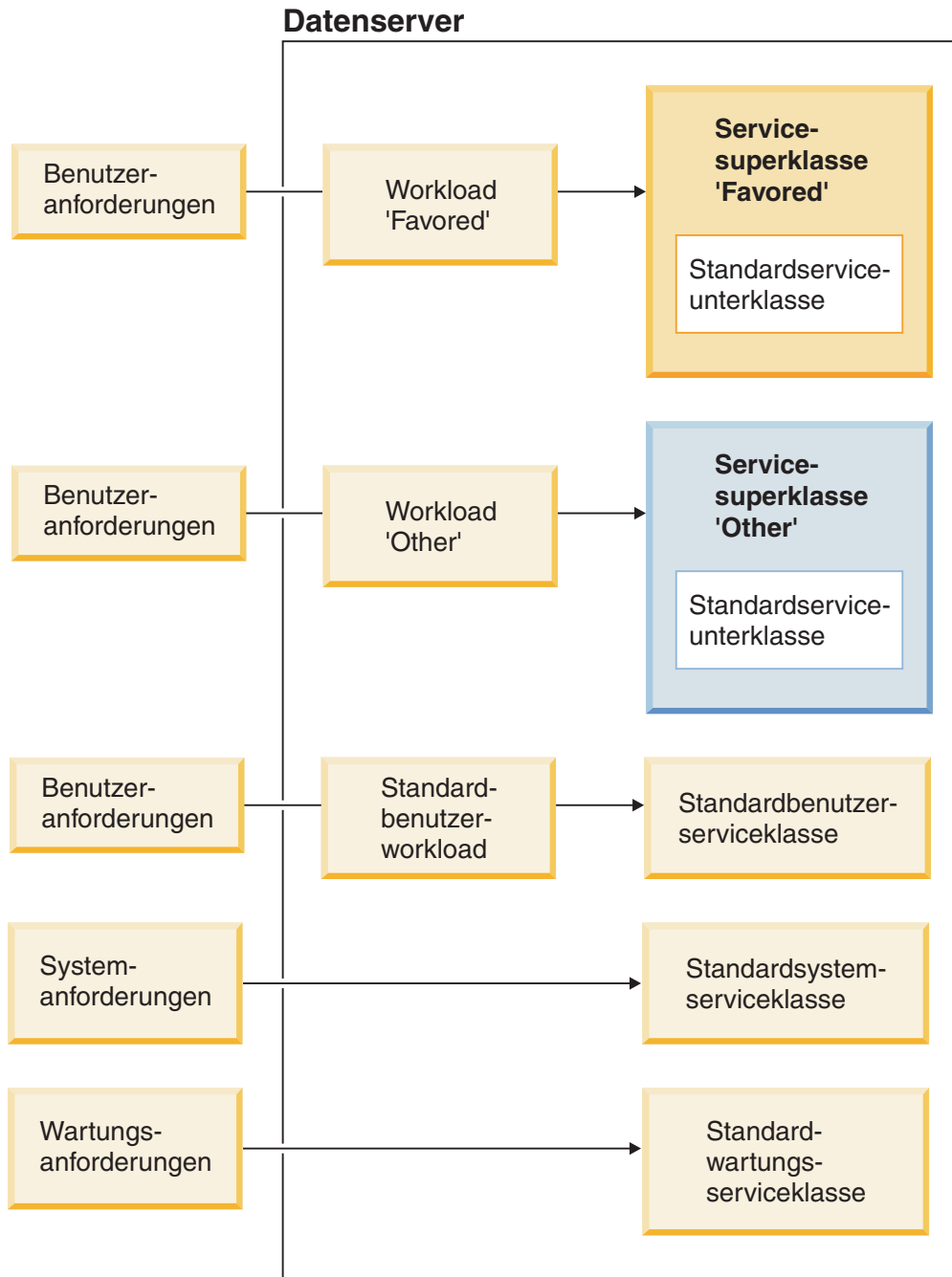


Abbildung 26. Datenserverkonfiguration

Aus Gründen der Vereinfachung wird davon ausgegangen, dass abgesehen von 'Favored' und 'Other' in den Serviceklassen ein unbedeutendes Arbeitsvolumen ausgeführt wird.

3. Messen Sie, ob die gewünschten CPU-Anteilsziele erreicht werden. Das System kann sich hinsichtlich der Workload in einem von mehreren Zuständen befinden:
 - a. 100 % CPU-Auslastung
 - 1) Arbeit wird in den beiden Workloads 'Favored' und 'Other' ausgeführt. Die gemessene CPU-Auslastung beträgt 70 % für die Serviceklasse 'Favored' und 30 % für die Serviceklasse 'Other'.

- 2) Arbeit wird nur in der Workload 'Favored' ausgeführt.
Die gemessene CPU-Auslastung beträgt 100 % für die Serviceklasse 'Favored' und 0 % für die Serviceklasse 'Other'.
 - 3) Arbeit wird nur in der Workload 'Other' ausgeführt.
Die gemessene CPU-Auslastung beträgt 0 % für die Serviceklasse 'Favored' und 100 % für die Serviceklasse 'Other'.
- b. Eine Serviceklasse verwendet weniger als ihr CPU-Nutzungsrecht.
- 1) Arbeit wird in den beiden Workloads 'Favored' und 'Other' ausgeführt.
Die Workload 'Favored' verwendet 50 % anstelle der vollständigen 70 %, die ihr aufgrund ihres CPU-Nutzungsrechts zustehen.
Die gemessene CPU-Auslastung der Workload 'Favored' beträgt 50 %.
Die Workload 'Other' verwendet weiterhin 30 %, weil ihre CPU-Anteile fest sind. Die CPU-Gesamtauslastung beträgt 80 %.
 - 2) Arbeit wird in den beiden Workloads 'Favored' und 'Other' ausgeführt.
Die Workload 'Other' verwendet 10 % anstelle der vollständigen 30 %, die ihr aufgrund ihres CPU-Nutzungsrechts zustehen.
Da ihre CPU-Anteile flexibel sind, erhöht sich die gemessene CPU-Auslastung der Serviceklasse 'Favored' von 70 % auf 90 %, wobei die nicht verwendeten 20 %, die von der Serviceklasse 'Other' freigegeben wurden, übernommen werden. Die CPU-Auslastung der Serviceklasse 'Other' beträgt 10 %. Die CPU-Gesamtauslastung beträgt 100 %.
 - 3) Arbeit wird nur in der Workload 'Favored' ausgeführt und sie verwendet ihr gesamtes CPU-Nutzungsrecht.
Da in der Serviceklasse 'Other' keine Arbeit ausgeführt wird, kann die Serviceklasse 'Favored' ihr eigenes CPU-Nutzungsrecht von 70 % und die nicht verwendeten 30 % der CPU-Ressourcen verwenden, die von der Serviceklasse 'Other' freigegeben wurden. Die CPU-Gesamtauslastung beträgt 100 %.
 - 4) Arbeit wird nur in der Workload 'Other' ausgeführt und sie verwendet ihr gesamtes CPU-Nutzungsrecht.
Da in der Serviceklasse 'Favored' keine Arbeit ausgeführt wird, kann die Serviceklasse 'Other' ihr eigenes CPU-Nutzungsrecht von 30 % und die nicht verwendeten 70 % der CPU-Ressourcen verwenden, die von der Serviceklasse 'Favored' freigegeben wurden. Die CPU-Gesamtauslastung beträgt 100 %.

Szenario 4: Sandbox zum Erzielen von Konsistenz erstellen (selbst auf Kosten einiger inaktiver CPU-Ressourcen)

Bei diesem Beispieleinsatzszenario verbraucht eine bevorzugte Workload unabhängig von der Verfügbarkeit nicht verwendeter CPU-Ressourcen nie mehr als ihren festen Anteil der CPU-Ressourcen.

In diesem Beispiel sollen die beiden Workloads 'Favored' und 'Other' verwendet werden. Die Workload 'Favored' soll zu jedem Zeitpunkt unabhängig von der Verfügbarkeit nicht verwendeter CPU-Ressourcen über ein Maximum von 70 % der CPU-Ressourcen verfügen. Die Workload 'Other' soll zu jedem Zeitpunkt unabhängig von der Verfügbarkeit nicht verwendeter CPU-Ressourcen über ein Maximum von 30 % der CPU-Ressourcen verfügen.

Die folgenden Schritte zeigen, wie das DB2-Workload-Management konfiguriert werden muss, um die oben genannten Ziele zu erreichen:

1. Erstellen Sie die Serviceklassen 'Favored' und 'Other' und weisen Sie jeder Serviceklasse einen CPU-Grenzwert zu, sodass das Maximum von 70 % bzw. 30 % der CPU-Ressourcen für die Workload 'Favored' bzw. 'Other' implementiert wird. Im folgenden Beispiel müssen die CPU-Grenzwerte als eine Zahl zwischen 1 und 100 angegeben werden, die den maximalen Prozentsatz der CPU-Auslastung für diese Serviceklasse darstellt:

```
create service class favored cpu limit 7000
create service class other cpu limit 3000
```

2. Erstellen Sie die Workloads 'Favored' und 'Other' so wie im folgenden Beispiel gezeigt:

```
create workload favored appl_name('favoredapp') service class favored
create workload other appl_name('otherapp') service class other
```

Das System sieht aufgrund dessen wie in Abb. 26 auf Seite 192 aus.

Aus Gründen der Vereinfachung wird davon ausgegangen, dass abgesehen von 'Favored' und 'Other' in den Serviceklassen ein unbedeutendes Arbeitsvolumen ausgeführt wird.

3. Messen Sie, ob die gewünschten CPU-Grenzwerte erfolgreich implementiert werden. Das System kann sich hinsichtlich der Workload in einem von mehreren Zuständen befinden:
 - a. 100 % CPU-Auslastung
 - 1) Arbeit wird in den beiden Workloads 'Favored' und 'Other' ausgeführt. Die gemessene CPU-Auslastung beträgt 70 % für die Serviceklasse 'Favored' und 30 % für die Serviceklasse 'Other'.
 - b. Mindestens eine Serviceklasse verwendet weniger als ihren CPU-Grenzwert.
 - 1) Arbeit wird in den beiden Workloads 'Favored' und 'Other' ausgeführt. Die Workload 'Favored' verwendet 50 % anstelle der vollständigen 70 %, die ihr aufgrund ihres CPU-Grenzwertnutzungsrechts zustehen. Die gemessene CPU-Auslastung der Workload 'Favored' beträgt 50 %. Die Workload 'Other' verwendet weiterhin 30 %, weil dies ihr maximales CPU-Grenzwertnutzungsrecht ist. Die CPU-Gesamtauslastung beträgt 80 %.
 - 2) Arbeit wird in den beiden Workloads 'Favored' und 'Other' ausgeführt. Die Workload 'Other' verwendet 10 % anstelle der vollständigen 30 %, die ihr aufgrund ihres CPU-Grenzwertnutzungsrechts zustehen. Aufgrund des CPU-Grenzwerts, der ein Maximum von 70 % CPU-Auslastung festlegt, bleibt die gemessene CPU-Auslastung der Serviceklasse 'Favored' bei 70 %. Die nicht verwendeten 20 % der CPU-Ressourcen, die von der Serviceklasse 'Other' freigegeben wurden, werden nicht übernommen. Die CPU-Auslastung der Serviceklasse 'Other' beträgt 10 %. Die CPU-Gesamtauslastung beträgt 80 %.
 - 3) Arbeit wird nur in der Workload 'Favored' ausgeführt und sie verwendet das maximale CPU-Ressourcenvolumen, das vom entsprechenden CPU-Grenzwert festgelegt ist. Aufgrund des CPU-Grenzwerts, der ein Maximum von 70 % CPU-Auslastung festlegt, bleibt die gemessene CPU-Auslastung der Serviceklasse 'Favored' bei 70 %. Die nicht verwendeten 30 % der CPU-Ressourcen, die von der Serviceklasse 'Other' freigegeben wurden, werden nicht übernommen. Die CPU-Auslastung der Serviceklasse 'Other' beträgt 0 %. Die CPU-Gesamtauslastung beträgt 70 %.

- 4) Arbeit wird nur in der Workload 'Other' ausgeführt und sie verwendet das maximale CPU-Ressourcenvolumen, das vom entsprechenden CPU-Grenzwert festgelegt ist.

Aufgrund des CPU-Grenzwerts, der ein Maximum von 30 % CPU-Auslastung festlegt, bleibt die gemessene CPU-Auslastung der Serviceklasse 'Other' bei 30 %. Die nicht verwendeten 70 % der CPU-Ressourcen, die von der Serviceklasse 'Favored' freigegeben wurden, werden nicht übernommen. Die CPU-Auslastung der Serviceklasse 'Favored' beträgt 0 %. Die CPU-Gesamtauslastung beträgt 30 %.

Szenario 5: Workloads müssen nach Benutzern aufgeteilt werden, jedoch muss ein Typ von Arbeit für einen angegebenen Benutzer bevorzugt werden

Bei den vorangegangenen Szenarios wurde die Zuordnung von CPU-Ressourcen zwischen verschiedenen Workloads auf der Basis von Geschäftsprioritäten mithilfe von Serviceklassen gesteuert. In einigen Fällen ist es jedoch zweckdienlicher, den Anteil von Ressourcen zwischen Workloads auf der Basis von Geschäftsprioritäten zu steuern und Arbeit mit kurzer Laufzeit eine höhere Priorität zu geben als Arbeit mit langer Laufzeit, um den Gesamtdurchsatz oder die Antwortzeiten zu verbessern.

Bei diesem Beispieleinsatzszenario greifen der Geschäftsführer und alle eintausend anderen Mitarbeiter über die Anwendung mybizapp auf eine Datenbank zu. Das Ziel dieser Task ist es, dem Geschäftsführer 10 % der CPU-Ressourcen zuzuteilen, sodass die verbleibenden 90 % der CPU-Ressourcen von den 1000 anderen Benutzern verwendet werden, die auf die Datenbank zugreifen. Erstellen Sie hierzu eine Servicesuperklasse für den Geschäftsführer (ceo_sc). Definieren Sie anschließend eine Workload, um den Geschäftsführer der Servicesuperklasse zuzuordnen, wie in den folgenden Beispielen gezeigt wird:

```
create service class ceo_sc soft cpu shares 1000
create workload ceo_wl
  applname('mybizapp')
  session_user('ceo')
  service class ceo_sc
```

Erstellen Sie eine Servicesuperklasse für alle anderen Benutzer (mybizapp_sc). Definieren Sie anschließend eine Workload (mybizapp_wl), um alle anderen Benutzer der Anwendung mybizapp der neu erstellten Servicesuperklasse mybizapp_sc zuzuordnen, wie im folgenden Beispiel gezeigt wird:

```
create service class mybizapp_sc hard cpu shares 9000
create workload mybizapp_wl
  applname('mybizapp')
  service class mybizapp_sc
```

In der Workload der Anwendung mybizapp gibt es Abfragen mit hohem und niedrigem Aufwand. Das Unternehmen hat festgelegt, dass Abfragen mit niedrigem Aufwand (weniger als 10.000 Timeron) im Allgemeinen eine höhere Priorität haben als Abfragen mit hohem Aufwand (mehr als 10.000 Timeron). Daher soll Abfragen mit niedrigem Aufwand ein größerer Anteil der CPU-Ressourcen zugeordnet werden als Abfragen mit hohem Aufwand, und zwar im Verhältnis 2:1. Sie erreichen dieses Ziel, indem Sie zwei Serviceunterklassen (lowcost_ssc und highcost_ssc) in der Servicesuperklasse mybizapp_sc erstellen und veränderliche CPU-Anteile im Verhältnis 2:1 zuweisen, wie im folgenden Beispiel gezeigt wird:

```
create service class lowcost_ssc under mybizapp_sc soft cpu shares 2000
create service class highcost_ssc under mybizapp_sc soft cpu shares 1000
```

Das Verhältnis 2:1 von veränderlichen CPU-Anteilen bestimmt, wie viel von den 90 % der CPU-Ressourcen, die der Servicesuperklasse mybizapp_sc zugewiesen sind, Abfragen mit niedrigem Aufwand zugeordnet wird und wie viel Abfragen mit hohem Aufwand zugeordnet wird.

Erstellen Sie ein Arbeitsklassenset (splitbycost_wcs) und ein Arbeitsaktionsset (mybizapp_was), um Abfragen mit niedrigem Aufwand an die Serviceunterklasse lowcost_ssc und Abfragen mit hohem Aufwand an die Serviceunterklasse highcost_ssc weiterzuleiten, wie in den folgenden Beispielen gezeigt wird:

```
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)
```

Wenn die Arbeit in jeder der Serviceklassen ausgeführt wird und ihr vollständiges CPU-Ressourcenauslastungsrecht nutzt, führen die vorherigen Konfigurationen zu einer Division der CPU-Ressourcen, die in der folgenden Abbildung dargestellt ist:

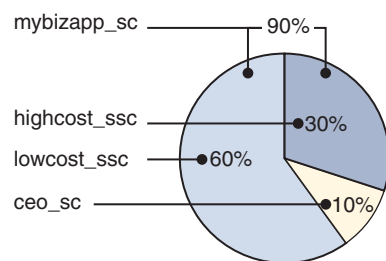


Abbildung 27. CPU-Ressourcenauslastungsrechte

Die Servicesuperklasse ceo_sc verfügt über 1.000 CPU-Anteile von insgesamt 10.000, was ein CPU-Ressourcenauslastungsrecht von 10 % bedeutet. Die Servicesuperklasse mybizapp_sc verfügt über 9.000 CPU-Anteile von insgesamt 10.000, was ein CPU-Ressourcenauslastungsrecht von 90 % bedeutet. In den 90 % des CPU-Ressourcenauslastungsrechts der Servicesuperklasse mybizapp_sc verfügt die Serviceunterklasse lowcost_ssc über 2 Anteile von 3, was zu einem CPU-Ressourcenauslastungsrecht von 66,7 % der CPU-Ressourcen der Servicesuperklasse (d. h. 60 % der CPU-Gesamtrressourcen) führt. Die Serviceunterklasse highcost_ssc verfügt über 1 Anteil von 3, was zu einem CPU-Ressourcenauslastungsrecht von 33,3 % der CPU-Ressourcen der Superklasse (d. h. 30 % der CPU-Gesamtrressourcen) führt.

Weitere Informationen

Zu den folgenden Themen werden an den angegebenen Stellen vollständige Details geliefert:

- Feste CPU-Anteile: „Feste CPU-Anteile“ auf Seite 200.
- Veränderliche CPU-Anteile: „Veränderliche CPU-Anteile“ auf Seite 208.
- CPU-Grenzwert: „CPU-Grenzwert“ auf Seite 214.
- Minimale CPU-Auslastungsebene zum Klassifizieren einer Serviceklasse als aktiv: „Minimale CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 230.

- Dispatcher-Thread-Parallelität: „Dispatcher-Thread-Parallelität“ auf Seite 233.
- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers: „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“ auf Seite 236.

Verhalten des Workload-Management-Dispatchers in Umgebungen mit mehreren Mitgliedern

Wenn der Workload-Management-Dispatcher in DB2-Instanzen mit mehreren Mitgliedern aktiviert ist, funktioniert die CPU-Ressourcenplanung in allen Mitgliedern eines bestimmten Hosts. Beachten Sie, dass der Workload-Management-Dispatcher unter AIX auch Umgebungen mit gemeinsam genutzten logischen Partitionen (Mikropartitionen) unterstützt.

Umgebungen mit Datenbanken mit mehreren Mitgliedern

In DB2-Instanzen mit mehreren Mitgliedern verhält sich der Dispatcher auf folgende Arten:

- Die relative Menge an CPU-Anteilen wird bewertet und CPU-Ressourcen werden allen Mitgliedern in der Instanz auf der Basis der Arbeit zugeordnet, die in dieser Instanz auf diesem Host oder dieser logischen Partition als Ganzes aktiv ist. Angenommen, Sie haben eine Datenbank mit zwei Mitgliedern, für die Arbeit in den beiden Serviceklassen A und B ausgeführt wird. Der Serviceklasse A werden 3.500 veränderliche CPU-Anteile und der Serviceklasse B werden 6.500 veränderliche CPU-Anteile zugewiesen. Die CPU-Auslastung für den gesamten Host oder die gesamte logische Partition beträgt 35 % für die Serviceklasse A und 65 % für die Serviceklasse B, obwohl es möglich ist, dass in einem der beiden Mitglieder relativ mehr Arbeit ausgeführt wird. Der Workload-Management-Dispatcher berücksichtigt beide Mitglieder, wenn er entscheidet, welche Serviceklasse zum gegebenen Zeitpunkt mehr CPU-Ressourcen erhalten soll.

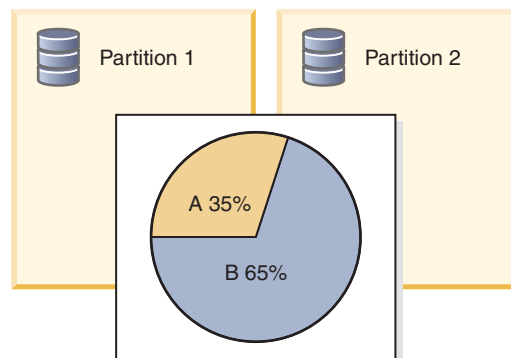


Abbildung 28. Umgebungen mit Datenbanken mit mehreren Mitgliedern: Kreisdiagramm für Zuordnungen von veränderlichen CPU-Anteilen in zwei Datenbankmitgliedern

- Der Wert des Konfigurationsparameters `wlm_disp_concur` des Datenbankmanagers gilt für jeden Host bzw. jede logische Partition. Bei Datenbanken mit mehreren Mitgliedern wird die über den Konfigurationsparameter `wlm_disp_concur` angegebene Dispatcher-Thread-Parallelität auf alle Mitglieder angewendet.
- Bei Datenbanken mit mehreren Mitgliedern werden CPU-Grenzwerte auf den einzelnen Hosts bzw. LPARs erzwungen. Wenn mehrere Mitglieder verwendet werden, darf jede Partition den gesamten angegebenen CPU-Grenzwert verwenden, solange die Summe aller Mitglieder kleiner-gleich dem CPU-Grenzwert ist.

Wenn der CPU-Grenzwert für die einzelne Serviceklasse A 35 % beträgt und wenn zwei Member vorhanden sind, kann das Member 1 bis zu 35 % verwenden, sofern die Summe der CPU-Auslastung in den Member 1 und 2 35 % nicht überschreitet.

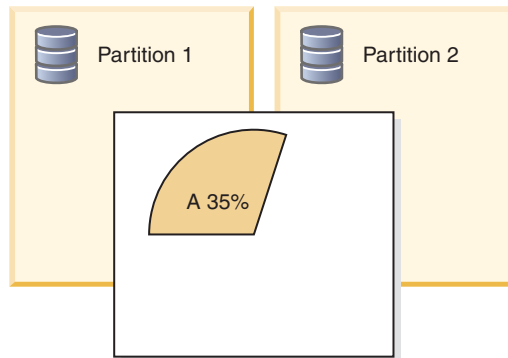


Abbildung 29. Datenbankumgebungen mit mehreren Membern: Kreisdiagramm für CPU-Grenzwert in zwei Datenbankmembern

Mikropartitionsumgebungen (Umgebungen mit gemeinsam genutzten logischen Partitionen)

Der Workload-Management-Dispatcher unterstützt Mikropartitionsumgebungen in AIX. In Mikropartitionsumgebungen wird die prozentuale CPU-Auslastung für Steuerelemente und die Überwachung im Workload-Management-Dispatcher relativ zu den CPU-Ressourcen berechnet, die der logischen Partition vom Betriebssystem oder Hypervisor im Verlauf des letzten Dispatcherzyklus für CPU-Ressourcenplanung zugeordnet werden. Hierbei werden die berechtigten (garantierten) CPU-Ressourcen für die logische Partition als Basis verwendet. In Fällen, wo die der logischen Partition zugeordnete CPU-Ressource größer ist als die berechtigten CPU-Ressourcen, berechnet der Dispatcher die CPU-Auslastung relativ zum größeren Betrag. Wenn die CPU-Ressourcenzuordnung kleiner ist als die berechtigten CPU-Ressourcen, berechnet der Dispatcher die CPU-Auslastung relativ zu den berechtigten CPU-Ressourcen.

Anmerkung: Das Verhalten des Workload-Management-Dispatchers ist mit den Berechnungen durch AIX-WLM konsistent. Bei dedizierten logischen Partitionen unter AIX und bei allen Nicht-AIX-Umgebungen wird die CPU-Auslastung relativ zur vollen CPU-Kapazität der physischen Kerne berechnet, die dem DB2-Datenbankmanager zur Verfügung stehen.

Aktivieren des Workload-Manager-Dispatchers

Setzen Sie den Konfigurationsparameter `wlm_dispatcher` des Datenbankmanagers auf YES, um den DB2-Workload-Management-Dispatcher zu aktivieren. Mit dem Workload-Management-Dispatcher werden die CPU-Ressourcen verwaltet, die DB2-Benutzer- und Wartungsserviceklassen zugeordnet sind.

Vorgehensweise

Gehen Sie wie folgt vor, um den Workload-Management-Dispatcher mit dem Befehlszeilenprozessor (Command Line Processor, CLP) zu aktivieren:

1. Stellen Sie eine Verbindung zu Ihrer DB2-Instanz her.

```
ATTACH TO instanzname
```


2. Setzen Sie den Befehl **UPDATE DATABASE MANAGER CONFIGURATION** (oder **UPDATE DBM CFG**) ab. Setzen Sie den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers auf YES.

```
UPDATE DBM CFG USING wlm_dispatcher yes
```

3. Optional: Heben Sie die Zuordnung für Ihre DB2-Instanz auf.

```
DETACH
```

Ergebnisse

Der Workload-Management-Dispatcher ist aktiviert. Sie können nun CPU-Grenzwertzuordnungen von CPU-Ressourcen zu DB2-Serviceklassen angeben.

Nächste Schritte

Nach der Aktivierung des Workload-Management-Dispatchers zum Verwalten von CPU-Ressourcen müssen Sie die folgenden Tasks ausführen:

- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie den CPU-Grenzwert über die Anweisung **CREATE SERVICE CLASS** oder **ALTER SERVICE CLASS**. Weitere Informationen finden Sie in „Festlegen eines CPU-Grenzwerts“ auf Seite 228.
- Aktivieren Sie feste und veränderliche CPU-Anteile, indem Sie den Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers konfigurieren. Weitere Informationen finden Sie in Schritt 1 in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie einen festen oder veränderlichen CPU-Anteil über die Anweisung **CREATE SERVICE CLASS** oder **ALTER SERVICE CLASS**. Weitere Informationen finden Sie in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Legen Sie den minimalen CPU-Auslastungsprozentsatz für den Workload-Management-Dispatcher zur Aktivierung von Serviceklassen durch Konfigurieren des Konfigurationsparameters **wlm_disp_min_util** des Datenbankmanagers fest. Weitere Informationen finden Sie in „Festlegen der minimalen CPU-Ressourcen-auslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 232.
- Legen Sie die Dispatcher-Thread-Parallelität durch Konfigurieren des Konfigurationsparameters **wlm_disp_concur** des Datenbankmanagers fest. Weitere Informationen finden Sie in „Festlegen der Dispatcher-Thread-Parallelität“ auf Seite 235.

Sicherstellen maximaler Planungsgenauigkeit des DB2-Workload-Management-Dispatchers

Der DB2-Datenbankmanager versucht, dem Zeitgeberthread **db2w1mt** und dem Planungsthread **db2w1mtm** Echtzeitprioritäten zuzuweisen, um sicherzustellen, dass für den Workload-Management-Dispatcher maximale Planungsgenauigkeit besteht. Dem DB2-Datenbankmanager müssen bestimmte betriebssystemspezifische Berechtigungen erteilt werden, damit die Zuweisung derartiger Prioritäten erfolgreich ist.

Vorgehensweise

Auch ohne die Fähigkeit, den Zeitgeber- und Planungsthreads Echtzeitprioritäten zuweisen zu können, kann der DB2-Workload-Management-Dispatcher Arbeit weiterhin priorisieren, jedoch nicht so genau oder optimal wie sonst. Die folgenden Tasks zum Sicherstellen maximaler Planungsgenauigkeit für den Workload-Management-Dispatcher sind nach dem Betriebssystem aufgelistet. Wählen Sie das entsprechende Betriebssystem aus und befolgen Sie die Anweisungen.

- Unter AIX-Betriebssystemen muss der Instanzeigner über die Funktionen CAP_NUMA_ATTACH und CAP_PROPAGATE verfügen, um mit AGENT PRIORITY eine höhere relative Priorität für Agenten in einer Serviceklasse festzulegen. Melden Sie sich als root an und führen Sie den folgenden Befehl aus, um diese Funktionen zu aktivieren:

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

- Unter Solaris 10 oder höher muss der Instanzeigner über das Zugriffsrecht proc_prioctl verfügen, um mit AGENT PRIORITY eine höhere relative Priorität für Agenten in einer Serviceklasse festzulegen. Melden Sie sich zur Erteilung dieses Zugriffsrechts als root an und führen Sie den folgenden Befehl aus:

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

In diesem Beispiel wird proc_prioctl der Standardgruppe der Zugriffsrechte des Benutzers db2user hinzugefügt.

Wenn der DB2-Datenbankmanager in einer nichtglobalen Zone von Solaris ausgeführt wird, muss das Zugriffsrecht proc_prioctl darüber hinaus der Zugriffsrechtsgruppe 'Limit' (L) für die Zone hinzugefügt werden. Melden Sie sich zur Erteilung dieses Zugriffsrechts an der Zone als root an und führen Sie den folgenden Befehl aus:

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

In diesem Beispiel wird proc_prioctl der Zugriffsrechtsgruppe 'Limit' (L) für die Zone db2zone hinzugefügt.

- Unter Solaris 9 gibt es keine Funktion für den DB2-Datenbankmanager, mit der die relative Priorität von Threads erhöht werden kann. Führen Sie ein Upgrade auf Solaris 10 oder höher durch, um dieses Feature nutzen zu können.

Feste CPU-Anteile

Der DB2-Workload-Management-Dispatcher kann CPU-Ressourcen über Nutzungsrechte auf der Basis von Anteilen verwalten, die Serviceklassen zugewiesen sind. Werden einer Serviceklasse mit Arbeit, die vom Administrator als Arbeit mit hoher Auswirkung oder niedrigerer Priorität eingestuft wird, feste CPU-Anteile zugewiesen, verbraucht diese Serviceklasse nicht mehr als ihren Anteil der CPU-Ressourcen, solange Arbeit in anderen Serviceklassen vorhanden ist, die auf dem System ausgeführt werden.

Einführung

Feste CPU-Anteile können jeder Benutzer- und Wartungserviceklasse, jedoch nicht der Systemserviceklasse zugewiesen werden. Nach der Aktivierung des Workload-Management-Dispatchers, der Überwachung Ihrer vorhandenen Workloads zur Ermittlung des Ausmaßes der CPU-Ressourcenauslastung und der Aktivierung des Attributs für CPU-Anteile für Serviceklassen können Sie den Serviceklassen, die Ihrer Meinung nach Arbeit mit niedrigerer Priorität oder hoher Auswirkung ausführen, feste CPU-Anteile zuweisen. Durch die Verwendung von festen CPU-Anteilen wird sichergestellt, dass die CPU-Belegung dieser Serviceklassen eingeschränkt wird, wenn andere Workloads vorhanden sind. Dadurch werden die Auswirkungen auf das System begrenzt und es wird sichergestellt, dass die restliche CPU für andere Arbeit mit höherer Priorität reserviert bleibt.

In den folgenden Abschnitten werden die Features und Funktionalität der festen CPU-Anteile ausführlicher beschrieben. Ein Einsatzszenarioabschnitt veranschaulicht die Features und Funktionalität der festen CPU-Anteile mit Einsatzbeispielen.

Features und Funktionalität

Wenn der Host oder die logische Partition mit 100 % CPU-Auslastung ausgeführt wird, spiegelt die Zuordnung von CPU-Ressourcen zwischen Serviceklassen lediglich ihre relativen Anteilsprozentsätze wider. Wenn der Host oder die logische Partition jedoch unter der vollständigen CPU-Auslastung ausgeführt werden, ist die erneute Zuordnung von CPU-Ressourcen komplex und hängt davon ab, ob das Attribut für CPU-Anteile für jede aktive Serviceklasse auf veränderliche oder feste CPU-Anteile gesetzt ist.

Eine Serviceklasse mit zugewiesenen festen CPU-Anteilen kann ihr CPU-Ressourcenauslastungsrecht nicht überschreiten (über die Konfiguration ihrer CPU-Anteile angegeben), um nicht verwendete CPU-Ressourcen zu verbrauchen, die auf dem Host oder der logischen Partition verfügbar werden. Der Workload-Management-Dispatcher respektiert immer das CPU-Ressourcenauslastungsrecht, das von der relativen Menge der zugewiesenen festen CPU-Anteile bestimmt wird, wenn Arbeit weiterhin in konkurrierenden Servicesuperklassen oder in konkurrierenden Serviceunterklassen in derselben Servicesuperklasse ausgeführt wird. Wenn konkurrierende Workloads nicht vorhanden sind oder wenn eine konkurrierende Workload vorübergehend völlig inaktiv wird, können Serviceklassen mit festen CPU-Anteilen nicht verwendete CPU-Ressourcen übernehmen.

Die Einstellung für feste CPU-Anteile ist am effektivsten, wenn Sie in Fällen verwendet wird, bei denen Sie das CPU-Ressourcenauslastungsrecht für eine Serviceklasse erzwingen wollen, um in dieser Serviceklasse ausgeführte Arbeit am Unterbrechen der Ausführung wichtigerer Arbeit auf dem Host bzw. in der logischen Partition zu hindern. Weisen Sie Serviceklassen, die komplexe oder intensive Abfragen ausführen, die andernfalls die Leistung von Arbeit mit höherer Priorität aufgrund der Konkurrenzsituation bei Ressourcen wie Ein-/Ausgabe, der Pufferpool oder CPU-Caches vermindert, feste CPU-Anteile zu.

Sie aktivieren das Attribut für CPU-Anteile, indem Sie den Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers auf YES setzen. Die Standardeinstellung für diesen Parameter ist NO. Nach der Aktivierung dieses Parameters werden alle vorhandenen und neu erstellten Serviceklassen standardmäßig 1.000 feste CPU-Anteile zugewiesen, um eine anfängliche gleiche Verteilung von CPU-Ressourcen sicherzustellen. Sie können feste CPU-Anteile über die Anweisungen CREATE SERVICE CLASS und ALTER SERVICE CLASS zuweisen und anpassen. Vollständige Details zum Aktivieren und Festlegen von festen CPU-Anteilen finden Sie in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.

Der Workload-Management-Dispatcher berechnet auf der Basis der Anzahl CPU-Anteile, die einer Serviceklasse zugewiesen sind, den Anteil der CPU-Ressourcen, die jede Serviceklasse verwenden darf. Sie können den Anteil der CPU-Ressourcen, zu denen jede Servicesuperklasse berechtigt ist, mit der folgenden Formel ermitteln, die die Anzahl CPU-Anteile einer bestimmten Servicesuperklasse in einen Prozentsatz von vom Workload-Management-Dispatcher zugeordneten CPU-Ressourcen konvertiert:

$$\% \text{ CPU(Superklasse)} = (\text{Anzahl Superklassenanteile} / \text{Gesamtzahl Anteile aller aktiven Superklassen}) \times 100$$

Sie können den Anteil der CPU-Ressourcen, zu denen jede Serviceunterklasse berechtigt ist, mit der folgenden Formel ermitteln, die die Anzahl CPU-Anteile einer bestimmten Serviceunterklasse in einen Prozentsatz von vom Workload-Management-Dispatcher zugeordneten CPU-Ressourcen konvertiert:

$$\% \text{ CPU(Unterklasse)} = \% \text{ CPU(Superklasse)} \times (\text{Anzahl Unterklassenanteile} / \text{Gesamtzahl Anteile aller aktiven Unterklassen in der Superklasse})$$

Anmerkung: Die Gesamtzahl (fester und veränderlicher) CPU-Anteile aller aktiven Superklassen werden in allen Datenbanken und Mitgliedern auf dem Host bzw. der logischen Partition gezählt.

Einsatzszenarios

Szenario 1

In Abb. 30 auf Seite 203 in der Anzeige A sind der Serviceklasse B feste CPU-Anteile und sind den Serviceklassen A und C veränderliche CPU-Anteile zugewiesen, deren Beträge in der Abbildungslegende beschrieben sind. Das Kreisdiagramm stellt den Anteil zugeordneter CPU-Ressourcen dar, zu denen jede dieser aktiven Serviceklassen berechtigt ist, und jede Serviceklasse verwendet ihren vollständigen Anteil der CPU-Ressourcen, was in diesem Beispiel 100 % CPU-Auslastung bedeutet. In der Anzeige B verfügt die Serviceklasse A nicht über genügend Arbeit, um ihr CPU-Nutzungsrecht vollständig verwenden zu können. Hierdurch sinkt die CPU-Auslastung von 60 % auf 50 %. Die nicht verwendeten 10 % der CPU-Ressourcen, die von der Serviceklasse A vorübergehend freigegeben werden, können nur von der konkurrierenden Serviceklasse C auf der Basis ihrer Zuweisung von veränderlichen CPU-Anteilen übernommen werden. Die Serviceklasse B kann ihre CPU-Ressourcenzuordnung von 30 % in diesem Beispiel nicht überschreiten, weil ihr feste CPU-Anteile zugewiesen sind und in den Serviceklassen A und C genügend Arbeit ausgeführt wird, um vom Dispatcher als aktiv eingestuft zu werden. In der Anzeige C wird die gesamte erneute Zuordnung von CPU-Ressourcen zur Serviceklasse C mit einer Erhöhung von 10 % auf 20 % der insgesamt verfügbaren CPU-Ressourcen dargestellt.

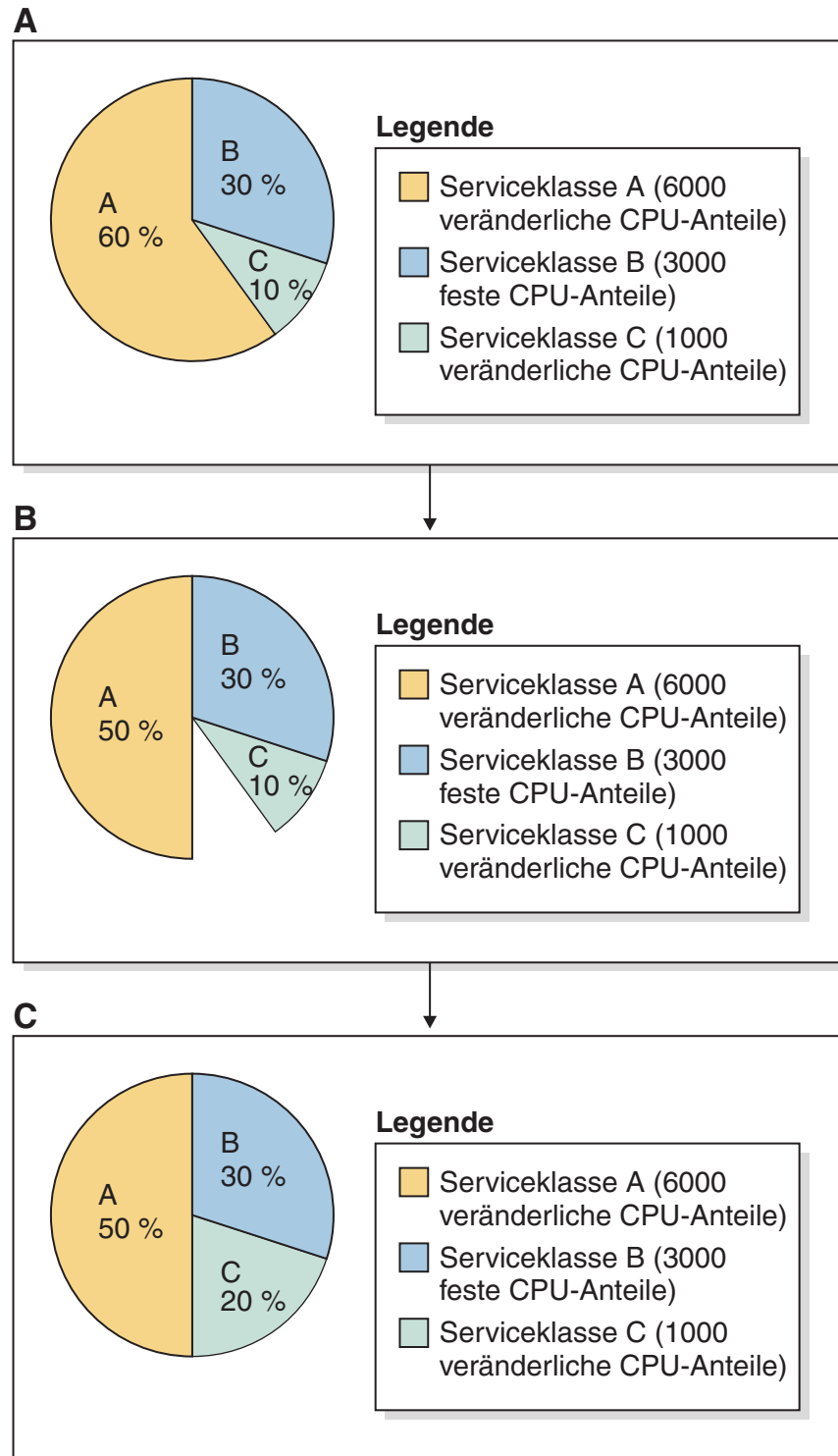


Abbildung 30. Kreisdiagramme für feste und veränderliche CPU-Anteile: Szenario 1

Wenn sich die Workload der Serviceklasse A erhöht, erhöht sich ihr Bedarf an CPU-Ressourcen. Unter diesem Umstand übergibt die Serviceklasse C alle übernommenen CPU-Ressourcen unverzüglich an die Serviceklasse A. Hierdurch werden die im Kreisdiagramm in der Anzeige A dargestellten CPU-Ressourcenanordnungen wiederhergestellt.

Anmerkung: Wenn Serviceklassen beim Verbrauchen von CPU-Ressourcen konkurrieren, werden einzelne Serviceklassenanforderungen von CPU-Ressourcen vom Workload-Management-Dispatcher nacheinander verarbeitet. Aufgrund von gewöhnlich häufigen und kurzfristigen Anforderungen von CPU-Ressourcen auf einem ausgelasteten Host bzw. einer ausgelasteten logischen Partition führt die erneute Zuordnung von nicht verwendeten CPU-Ressourcen im Verlauf der Zeit zu einer gleichmäßigen Neuverteilung von CPU-Ressourcen, die zu den relativen Zuweisungen von CPU-Anteilen proportional ist.

Szenario 2

Die Verwendung von festen Anteilen kann dazu führen, dass einige Teile der CPU-Ressourcen auf dem Datenbankserver nicht ausgelastet sind. Nicht ausgelastete CPU-Ressourcen können in Fällen auftreten, wo die anderen Serviceklassen keine Workload ausführen, die groß genug war, um ihre CPU-Ressourcenzuordnungen vollständig zu verwenden. Nicht ausgelastete CPU-Ressourcen sind in Fällen wünschenswert, wo mithilfe der festen Anteile eine intensive Workload begrenzt wird, die andernfalls den Fortschritt anderer Serviceklassen behindern kann, selbst wenn die CPU-Ressourcen nicht vollständig ausgelastet sind. Dieser Umstand tritt im Allgemeinen aufgrund der Konkurrenzsituation bei Ressourcen wie Ein-/Ausgabe oder der CPU-Cache auf.

In Abb. 31 auf Seite 205 in der Anzeige A sind den Serviceklassen B und C feste CPU-Anteile und sind der Serviceklasse A veränderliche CPU-Anteile zugewiesen, deren Beträge in der Abbildungslegende beschrieben sind. Das Kreisdiagramm stellt den Anteil zugeordneter CPU-Ressourcen dar, zu denen jede dieser aktiven Serviceklassen berechtigt ist, und jede Serviceklasse verwendet ihren vollständigen Anteil der CPU-Ressourcen, was in diesem Beispiel 100 % CPU-Auslastung bedeutet. In der Anzeige B verfügt die Serviceklasse A nicht über genügend Arbeit, um ihr CPU-Nutzungsrecht vollständig verwenden zu können. Hierdurch sinkt die CPU-Auslastung von 60 % auf 50 %. Die nicht verwendeten 10 % der CPU-Ressourcen, die von der Serviceklasse A vorübergehend freigegeben werden, können von den konkurrierenden Serviceklassen B und C auf der Basis ihrer Zuweisung von festen CPU-Anteilen nicht übernommen werden. Die Serviceklassen B und C können ihre CPU-Ressourcenzuordnung von 30 % bzw. 10 % in diesem Beispiel nicht überschreiten, weil ihnen feste CPU-Anteile zugewiesen sind und in der Serviceklasse A genügend Arbeit ausgeführt wird, um vom Dispatcher als aktiv eingestuft zu werden. (Die CPU-Auslastung fällt unter die für den Konfigurationsparameter `wlm_disp_min_util` des Datenbankmanagers konfigurierte Ebene; der Standardwert ist 5 %.) Die Anzeige C zeigt, dass die nicht verwendeten CPU-Ressourcen nicht erneut zugeordnet werden und dass sie in diesem Szenario nicht ausgelastet sind.

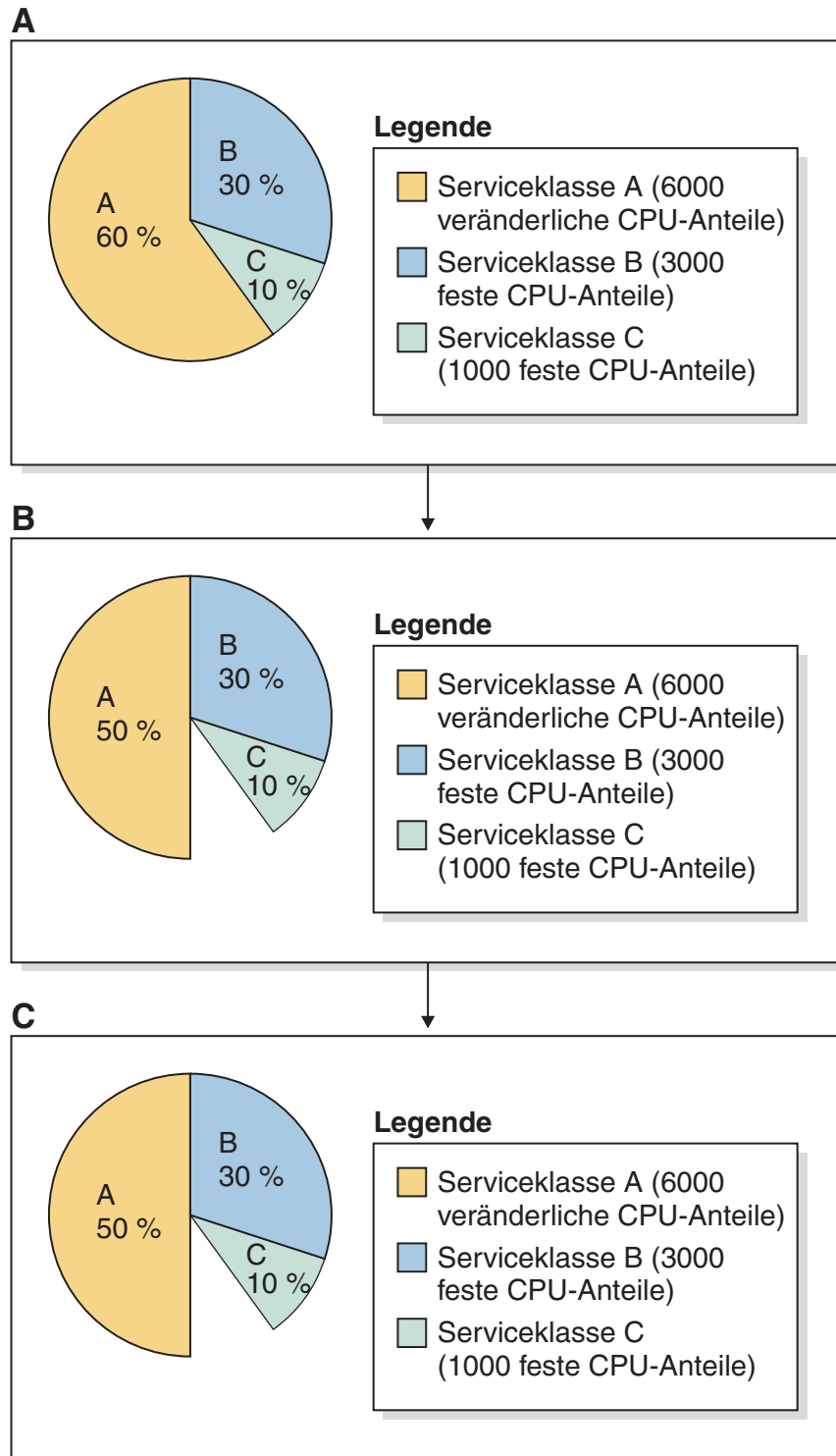


Abbildung 31. Kreisdiagramme für feste und veränderliche CPU-Anteile: Szenario 2

Dieses Szenario zeigt, dass Sie den Fortschritt von Arbeit, die in einer Serviceklasse mit hoher Priorität ausgeführt wird, vor Unterbrechungen durch Arbeit, die in Serviceklassen mit niedriger Priorität ausgeführt wird, schützen können.

Szenario 3

Feste CPU-Anteile haben im Vergleich zu traditionellen prozentualen CPU-Grenzwerten Vorteile. Da andere Workloads fehlen, kann die Serviceklasse mit festen CPU-Anteilen nicht verwendete CPU-Ressourcen übernehmen. Daher wird eine Serviceklasse mit einer Zuweisung von festen CPU-Anteilen im Gegensatz zu einer Serviceklasse mit festen CPU-Grenzwerten nicht künstlich begrenzt, wenn auf dem Host oder der logischen Partition keine andere Arbeit vorhanden ist.

In Abb. 32 auf Seite 207 in der Anzeige A sind den Serviceklassen B und C feste CPU-Anteile und sind der Serviceklasse A veränderliche CPU-Anteile zugewiesen, deren Beträge in der Abbildungslegende beschrieben sind. Das Kreisdiagramm stellt den Anteil zugeordneter CPU-Ressourcen dar, zu denen jede dieser aktiven Serviceklassen berechtigt ist, und jede Serviceklasse verwendet ihren vollständigen Anteil der CPU-Ressourcen, was in diesem Beispiel 100 % CPU-Auslastung bedeutet. In der Anzeige B verfügt die Serviceklasse A über keine Arbeit, um ihr CPU-Nutzungsrecht vollständig verwenden zu können. Hierdurch sinkt die CPU-Auslastung von 60 % auf 0 %. Der Dispatcher stuft die Serviceklasse als inaktiv ein. Die nicht verwendeten 60 % der CPU-Ressourcen, die von der Serviceklasse A vorübergehend freigegeben werden, können nun von den konkurrierenden Serviceklassen B und C auf der Basis ihrer Zuweisung von festen CPU-Anteilen und einer inaktiven Serviceklasse übernommen werden. Die Serviceklassen B und C können ihre CPU-Ressourcenzuordnungen von 30 % bzw. 10 % unter diesem Umstand überschreiten, weil ihnen feste CPU-Anteile zugewiesen sind und in der Serviceklasse A nicht genügend Arbeit ausgeführt wird, um vom Dispatcher als aktiv eingestuft zu werden. (Die CPU-Auslastung fällt unter die für den Konfigurationsparameter `wlm_disp_min_util` des Datenbankmanagers konfigurierte Ebene; der Standardwert ist 1 %.) Die Anzeige C zeigt, dass der Serviceklasse B 75 % $((3000 / (3000 + 1000)) \times 100)$ der CPU-Ressourcen und der Serviceklasse C 25 % $((1000 / (3000 + 1000)) \times 100)$ zugeordnet sind.

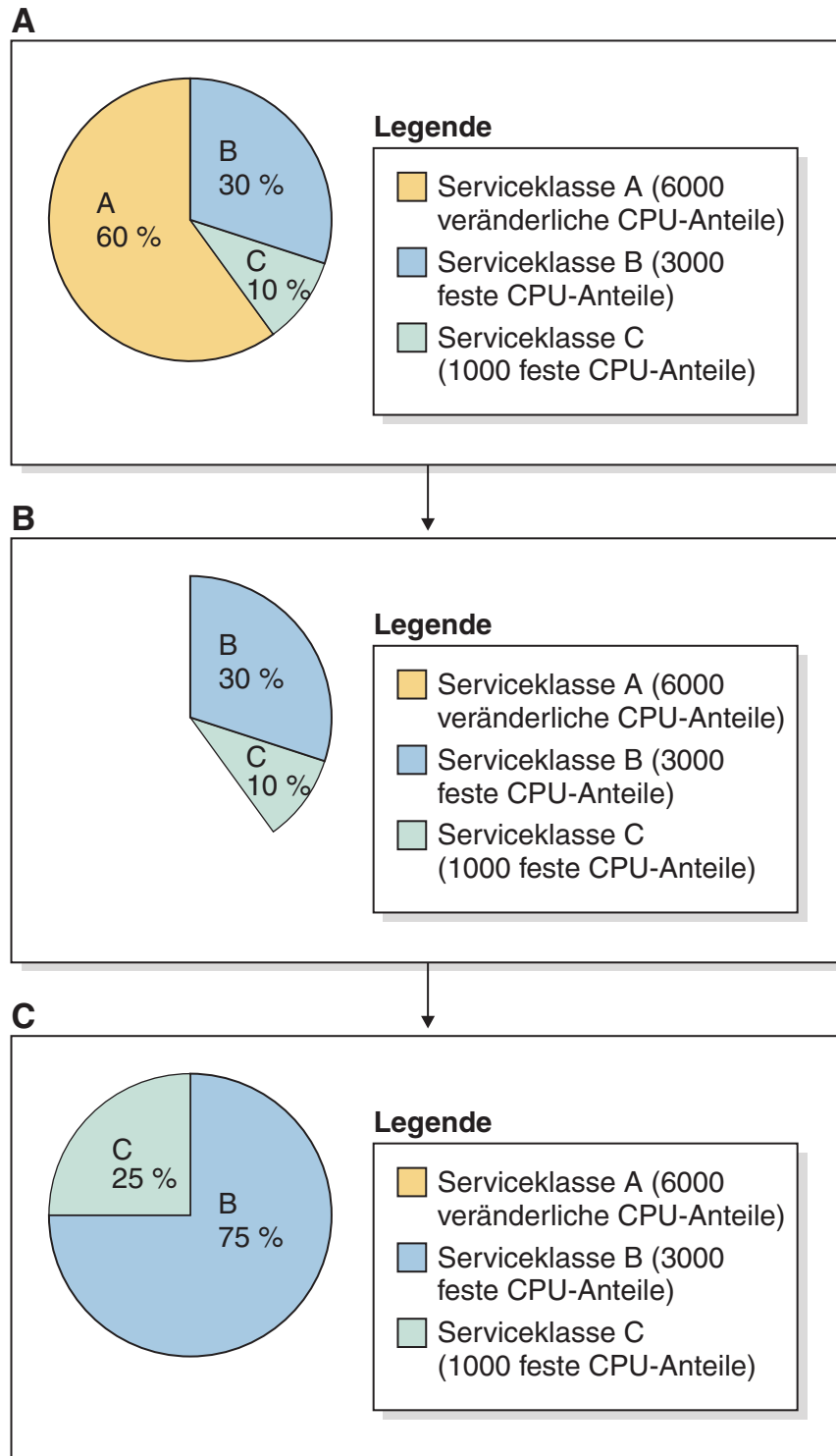


Abbildung 32. Kreisdiagramme für feste und veränderliche CPU-Anteile: Szenario 3

Wenn sich die Workload der Serviceklasse A erhöht, erhöht sich ihr Bedarf an CPU-Ressourcen. Unter diesem Umstand übergeben die Serviceklassen B und C alle übernommenen CPU-Ressourcen unverzüglich an die Serviceklasse A. Hierdurch werden die im Kreisdiagramm in der Anzeige A dargestellten CPU-Ressourcenzuordnungen wiederhergestellt.

Dieses Szenario zeigt, dass Sie zwar den Fortschritt von Arbeit, die in Serviceklassen mit hoher Priorität ausgeführt wird, vor Unterbrechungen durch Arbeit, die in Serviceklassen mit niedriger Priorität ausgeführt wird, schützen können. Wenn die Arbeit mit hoher Priorität jedoch nicht mehr vorhanden ist (z. B. in den Geschäftszeiten mit geringer Systemauslastung), können Serviceklassen mit niedriger Priorität mit festen CPU-Anteilen die nicht verwendeten CPU-Ressourcen weiterhin übernehmen.

Anmerkung: Wenn die Serviceklasse B oder C ihre vollständige CPU-Ressourcenzuordnung nicht verwendet und vom Dispatcher als Arbeit ausführend eingestuft wird, kann die andere Serviceklasse mit festen CPU-Anteilen von diesem Umstand nicht profitieren und nicht mehr als ihre Zuordnung verwenden.

Weitere Informationen

Zu den folgenden Workload-Management-Dispatcher-Themen finden Sie an den angegebenen Stellen vollständige Details:

- Workload-Management-Dispatcher: „Workload-Management-Dispatcher“ auf Seite 176.
- Veränderliche CPU-Anteile: „Veränderliche CPU-Anteile“.
- CPU-Grenzwert: „CPU-Grenzwert“ auf Seite 214.
- Minimale CPU-Auslastungsebene zum Klassifizieren einer Serviceklasse als aktiv: „Minimale CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 230.
- Dispatcher-Thread-Parallelität: „Dispatcher-Thread-Parallelität“ auf Seite 233.
- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers: „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“ auf Seite 236.

Veränderliche CPU-Anteile

Der DB2-Workload-Management-Dispatcher kann CPU-Ressourcen über Nutzungsrechte auf der Basis von Anteilen verwalten, die Serviceklassen zugewiesen sind. Werden einer Serviceklasse durch den Administrator veränderliche CPU-Anteile zugewiesen, kann diese Serviceklasse mehr als ihren Anteil von CPU-Ressourcen verbrauchen, wenn sie nicht verwendet werden. Wenn diese Serviceklasse zusammen mit anderen Serviceklassen verwendet wird, die durch feste CPU-Anteile gebunden sind, bedeutet dies für diese Serviceklasse hinsichtlich der CPU-Ressourcen bevorzugte Behandlung.

Einführung

Veränderliche CPU-Anteile können jeder Benutzer- und Wartungserviceklasse, jedoch nicht der Systemserviceklasse zugewiesen werden. Nachdem Sie den Workload-Management-Dispatcher aktiviert und Ihre vorhandenen Workloads überwacht haben, um das Ausmaß der CPU-Ressourcenauslastung zu ermitteln, können Sie den Serviceklassen, die eine hohe Priorität besitzen, veränderliche CPU-Anteile zuweisen. Veränderliche CPU-Anteile sind am effektivsten, wenn sie für Workloads mit hoher Priorität eingesetzt werden, weil sie zulassen, dass eine Workload mehr als das angegebene Nutzungsrecht verbraucht, wenn inaktive CPU-Ressourcen im System vorhanden sind. Die Verwendung von veränderlichen CPU-Anteilen wird nicht empfohlen, wenn Sie die CPU-Belegung von Arbeit mit niedriger Priorität oder mit starken Auswirkungen beschränken möchten. Verwenden Sie in diesem Fall stattdessen feste CPU-Anteile.

In den folgenden Abschnitten werden die Features und Funktionalität der veränderlichen CPU-Anteile ausführlicher beschrieben. Ein Einsatzszenarioabschnitt veranschaulicht die Features und Funktionalität der veränderlichen CPU-Anteile mit einem Einsatzbeispiel.

Features und Funktionalität

Wenn der Host oder die logische Partition mit 100 % CPU-Auslastung ausgeführt wird, spiegelt die Zuordnung von CPU-Ressourcen zwischen Serviceklassen lediglich ihre relativen Anteilsprozentsätze wider. Wenn der Host oder die logische Partition jedoch unter der vollständigen CPU-Auslastung ausgeführt werden, ist die erneute Zuordnung von CPU-Ressourcen komplex und hängt davon ab, ob das Attribut für CPU-Anteile für jede aktive Serviceklasse auf veränderliche oder feste CPU-Anteile gesetzt ist.

Eine Serviceklasse mit zugewiesenen veränderlichen CPU-Anteilen kann ihr CPU-Ressourcenauslastungsrecht überschreiten (über die Konfiguration ihrer CPU-Anteile angegeben), um nicht verwendete CPU-Ressourcen zu verbrauchen, die auf dem Host oder der logischen Partition verfügbar werden. Wenn mindestens zwei Serviceklassen über veränderliche Anteile verfügen und nicht verwendete CPU-Ressourcen bei genügend CPU-Ressourcenbedarf von jeder Serviceklasse zum Verbrauchen freier Kapazität verfügbar werden, werden die CPU-Ressourcen den konkurrierenden Serviceklassen je nach dem relativen Anteil jeder aktiven Serviceklasse proportional zugeordnet. Die Einstellung für veränderliche CPU-Anteile ist bei Arbeit mit hoher Priorität, die verfügbar werdende freie CPU-Ressourcen vorübergehend übernehmen soll, am effektivsten. Außerdem ist die Einstellung für veränderliche CPU-Anteile am effektivsten für Workloads, die aus kurzen Abfragen bestehen, die sich abgesehen von ihrer unmittelbaren CPU-Belegung relativ geringfügig auf die Datenbankressourcen auswirken.

Sie aktivieren das Attribut für CPU-Anteile, indem Sie den Konfigurationsparameter `wlm_disp_cpu_shares` des Datenbankmanagers auf YES setzen. Die Standardeinstellung für diesen Parameter ist NO. Sie können veränderliche CPU-Anteile über die Anweisungen CREATE SERVICE CLASS und ALTER SERVICE CLASS zuweisen und anpassen. Vollständige Details zum Aktivieren und Festlegen von veränderlichen CPU-Anteilen finden Sie in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.

Der Workload-Management-Dispatcher berechnet auf der Basis der Anzahl CPU-Anteile, die einer Serviceklasse zugewiesen sind, den Anteil der CPU-Ressourcen, die jede Serviceklasse verwenden darf. Sie können den Anteil der CPU-Ressourcen, zu denen jede Servicesuperklasse berechtigt ist, mit der folgenden Formel ermitteln, die die Anzahl CPU-Anteile einer bestimmten Servicesuperklasse in einen Prozentsatz von vom Workload-Management-Dispatcher zugeordneten CPU-Ressourcen konvertiert:

$$\% \text{ CPU(Superklasse)} = (\text{Anzahl Superklassenanteile} / \text{Gesamtzahl Anteile aller aktiven Superklassen}) \times 100$$

Sie können den Anteil der CPU-Ressourcen, zu denen jede Serviceunterklasse berechtigt ist, mit der folgenden Formel ermitteln, die die Anzahl CPU-Anteile einer bestimmten Serviceunterklasse in einen Prozentsatz von vom Workload-Management-Dispatcher zugeordneten CPU-Ressourcen konvertiert:

$$\% \text{ CPU(Unterklasse)} = \% \text{ CPU(Superklasse)} \times (\text{Anzahl Unterklassenanteile} / \text{Gesamtzahl Anteile aller aktiven Unterklassen in der Superklasse})$$

Anmerkung: Die Gesamtzahl (fester und veränderlicher) CPU-Anteile aller aktiven Superklassen werden in allen Datenbanken und Mitgliedern auf dem Host bzw. der logischen Partition gezählt.

Einsatzszenario

In Abb. 33 auf Seite 211 in der Anzeige A sind den Serviceklassen A, B und C veränderliche CPU-Anteile zugewiesen, deren Beträge in der Abbildungslegende beschrieben sind. Das Kreisdiagramm stellt den Anteil zugeordneter CPU-Ressourcen dar, zu denen jede dieser aktiven Serviceklassen berechtigt ist, und jede Serviceklasse verwendet ihren vollständigen Anteil der CPU-Ressourcen, was in diesem Beispiel 100 % CPU-Auslastung bedeutet. In der Anzeige B verfügt die Serviceklasse A nicht über genügend Arbeit, um ihr CPU-Nutzungsrecht vollständig verwenden zu können. Hierdurch sinkt die CPU-Auslastung von 60 % auf 50 %. Die nicht verwendeten 10 % der CPU-Ressourcen, die von der Serviceklasse A vorübergehend freigegeben werden, können von den konkurrierenden Serviceklassen B und C auf der Basis ihrer relativen Zuweisungen von veränderlichen CPU-Anteilen proportional übernommen werden. In der Anzeige C ist die proportionale erneute Zuordnung von CPU-Ressourcen zwischen den Serviceklassen B und C dargestellt. Die Serviceklasse B erhält 7,5 % ($10 \% \times (3000/4000)$) und die Serviceklasse C erhält 2,5 % ($10 \% \times (1000/4000)$) der nicht verwendeten 10 %, die von der Serviceklasse A freigegeben wurden.

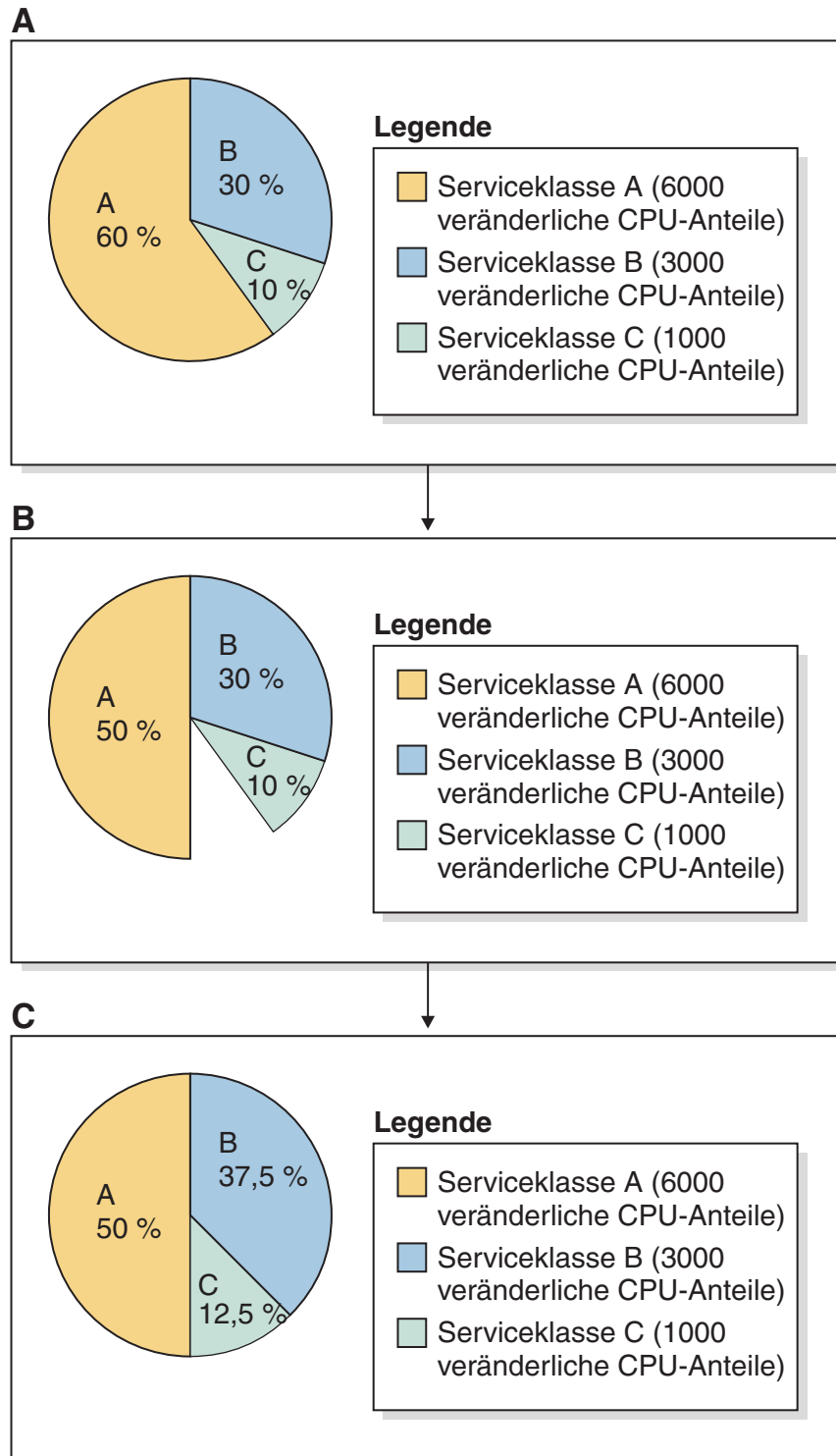


Abbildung 33. Kreisdiagramme für veränderliche CPU-Anteile

Wenn sich die Workload der Serviceklasse A erhöht, erhöht sich ihr Bedarf an CPU-Ressourcen. Unter diesem Umstand übergeben die Serviceklassen B und C alle übernommenen CPU-Ressourcen unverzüglich an die Serviceklasse A. Hierdurch werden die im Kreisdiagramm in der Anzeige A dargestellten CPU-Ressourcenzuordnungen wiederhergestellt.

Anmerkung: Wenn Serviceklassen beim Verbrauchen von CPU-Ressourcen konkurrieren, werden einzelne Serviceklassenanforderungen von CPU-Ressourcen vom Workload-Management-Dispatcher nacheinander verarbeitet. Aufgrund von gewöhnlich häufigen und kurzfristigen Anforderungen von CPU-Ressourcen auf einem ausgelasteten Host bzw. einer ausgelasteten logischen Partition führt die erneute Zuordnung von nicht verwendeten CPU-Ressourcen im Verlauf der Zeit zu einer gleichmäßigen Neuverteilung von CPU-Ressourcen, die zu den relativen Zuweisungen von CPU-Anteilen proportional ist.

Weitere Informationen

Zu den folgenden Workload-Management-Dispatcher-Themen finden Sie an den angegebenen Stellen vollständige Details:

- Workload-Management-Dispatcher: „Workload-Management-Dispatcher“ auf Seite 176.
- Feste CPU-Anteile: „Feste CPU-Anteile“ auf Seite 200.
- CPU-Grenzwert: „CPU-Grenzwert“ auf Seite 214.
- Minimale CPU-Auslastungsebene zum Klassifizieren einer Serviceklasse als aktiv: „Minimale CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 230.
- Dispatcher-Thread-Parallelität: „Dispatcher-Thread-Parallelität“ auf Seite 233.
- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers: „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“ auf Seite 236.

Aktivieren und Festlegen von CPU-Anteilen

Sie aktivieren die Attribute für CPU-Anteile, indem Sie den Konfigurationsparameter `wlm_disp_cpu_shares` des Datenbankmanagers auf YES setzen. Sie legen das Attribut für feste oder veränderliche CPU-Anteile mit der Anweisung CREATE SERVICE CLASS oder ALTER SERVICE CLASS fest. Normalerweise ordnen Sie Serviceklassen, die Arbeit mit niedriger Priorität oder starken Auswirkungen ausführen, feste CPU-Anteile zu und legen die veränderlichen CPU-Anteile für Serviceklassen mit hoher Priorität fest, wenn CPU-Ressourcen stark beansprucht werden.

Vorbereitende Schritte

Anmerkung: Sie müssen den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter `wlm_dispatcher` des Datenbankmanagers konfigurieren, damit die Einstellung der CPU-Anteile wirksam wird. Sie können diese Task vor oder nach der Aktivierung und Festlegung der CPU-Anteile ausführen. Weitere Informationen finden Sie in „Aktivieren des Workload-Manager-Dispatchers“ auf Seite 198.

Informationen zu diesem Vorgang

Die Attribute für CPU-Anteile werden in dieser Task über den Befehlszeilenprozessor aktiviert und das Attribut für CPU-Anteile wird mit der Anweisung CREATE SERVICE CLASS oder ALTER SERVICE CLASS für eine Serviceklasse festgelegt.

Einschränkungen

Sie können veränderliche CPU-Anteile nur Serviceklassen zuweisen, die über den Workload-Management-Dispatcher gesteuert werden können, d. h., Benutzer- und

Wartungserviceklassen. Die CPU-Ressourcenzuordnungen für Systemserviceklassen können vom Workload-Management-Dispatcher nicht gesteuert werden.

Vorgehensweise

Fahren Sie mit Schritt 2 fort, wenn die Attribute für CPU-Anteile bereits aktiviert wurden.

Gehen Sie wie folgt vor, um die Attribute für CPU-Anteile zu aktivieren:

1. Setzen Sie den Befehl **UPDATE DATABASE MANAGER CONFIGURATION** (oder **UPDATE DBM CFG**) über den Befehlszeilenprozessor ab und setzen Sie den Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers auf YES. Der Parameterwert wird sofort aktualisiert, indem zuerst eine Verbindung zur DB2-Instanz hergestellt wird:

```
attach to instanzname
update dbm cfg using wlm_disp_cpu_shares yes
detach
```

Fahren Sie mit Schritt 3 fort, wenn eine vorhandene Einstellung für feste oder veränderliche CPU-Anteile einer Serviceklasse geändert werden soll.

Gehen Sie wie folgt vor, um eine neue Serviceklasse zu erstellen und das Attribut für feste oder veränderliche CPU-Anteile festzulegen:

2. Setzen Sie die folgende Anweisung **CREATE SERVICE CLASS** ab, um eine neue Serviceklasse zu erstellen und die festen oder veränderlichen CPU-Anteile auf 5.000 zu setzen:
 - create service class *serviceklassenname* hard cpu shares 5000
 - create service class *serviceklassenname* soft cpu shares 5000

Gehen Sie wie folgt vor, um die Einstellung der festen oder veränderlichen CPU-Anteile für eine vorhandene Serviceklasse zu ändern:

3. Setzen Sie die folgende Anweisung **ALTER SERVICE CLASS** ab, um den Wert der festen oder veränderlichen CPU-Anteile in 15.000 zu ändern:
 - alter service class *serviceklassenname* hard cpu shares 15000
 - alter service class *serviceklassenname* soft cpu shares 15000

Ergebnisse

Die Attribute für CPU-Anteile wurden aktiviert und Sie haben die Einstellung der festen oder veränderlichen CPU-Anteile für die angegebene Serviceklasse festgelegt oder geändert. Feste CPU-Anteile, die Serviceklassen zugewiesen sind, können unter eingeschränkten Zuständen CPU-Ressourcen übernehmen, die anderen Serviceklassen aufgrund ihres gesenkten Workloadbedarfs freigegeben haben. Veränderliche CPU-Anteile, die Serviceklassen zugewiesen sind, können dagegen immer CPU-Ressourcen übernehmen, die andere Serviceklassen aufgrund ihres gesenkten Workloadbedarfs freigegeben haben.

Nächste Schritte

Führen Sie die folgenden Tasks aus, nachdem Sie den Workload-Management-Dispatcher zum Verwalten von CPU-Ressourcen aktiviert, die Attribute für die CPU-Anteile aktiviert und das Attribut für feste oder veränderliche CPU-Anteile festgelegt haben:

- Sie müssen den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers konfigurieren, damit die Einstellung der festen oder veränderlichen CPU-Anteile wirksam

wird. Weitere Informationen finden Sie in „Aktivieren des Workload-Manager-Dispatchers“ auf Seite 198.

- Legen Sie die Dispatcher-Thread-Parallelität durch Konfigurieren des Konfigurationsparameters **wlm_disp_concur** des Datenbankmanagers fest. Weitere Informationen finden Sie in „Festlegen der Dispatcher-Thread-Parallelität“ auf Seite 235.
- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie den CPU-Grenzwert über die Anweisung CREATE SERVICE CLASS oder ALTER SERVICE CLASS. Weitere Informationen finden Sie in „Festlegen eines CPU-Grenzwerts“ auf Seite 228.
- Legen Sie den minimalen CPU-Auslastungsprozentsatz für den Workload-Management-Dispatcher zur Aktivierung von Serviceklassen fest, indem Sie den Konfigurationsparameter **wlm_disp_min_util** des Datenbankmanagers konfigurieren. Weitere Informationen finden Sie in „Festlegen der minimalen CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 232.

CPU-Grenzwert

Der DB2-Workload-Management-Dispatcher kann Nutzungsrechte mit festem CPU-Grenzwert erzwingen, die Servicesuperklassen und -unterklassen zugewiesen werden können. Durch die Anwendung eines CPU-Grenzwerts können Sie die CPU-Ressourcen, die von einer Serviceklasse verbraucht werden, ungeachtet anderer im DB2-Datenbankmanager ausgeführter Arbeit auf eine feste Menge begrenzen. Der restliche Teil der CPU-Ressourcen bleibt so für die Verwendung durch andere Konsumenten verfügbar. Wenn CPU-Grenzwerte zusammen mit CPU-Anteilen verwendet werden, wird immer die am meisten einschränkende Bedingung berücksichtigt.

Einführung

Ein CPU-Grenzwert kann jeder Benutzer- und Wartungserviceklasse, jedoch nicht der Systemserviceklasse zugewiesen werden. Nachdem Sie den Workload-Management-Dispatcher aktiviert und Ihre vorhandenen Workloads überwacht haben, um das Ausmaß der CPU-Ressourcenauslastung zu ermitteln, können Sie den Serviceklassen, die in ihrer CPU-Belegung streng eingeschränkt werden sollen, CPU-Grenzwerte zuweisen.

Mit CPU-Anteilen können Sie das CPU-Ressourcenauslastungsrecht einzelner Workloads steuern, wenn die Gesamtworkload des Hosts oder der logischen Partition hoch ist, allerdings CPU-Ressourcen nicht verschwenden, wenn die Gesamtworkload niedrig ist. Es gibt jedoch Workloads, deren CPU-Ressourcenauslastungsrecht Sie trotz der niedrigen Gesamtworkload des Hosts oder der logischen Partition immer begrenzen wollen. Wenn z. B. mehrere Abteilungen die Kosten für den Kauf eines Datenbankservers teilen, will jede Abteilung sicherstellen, dass die anderen Abteilungen ihre zugeordneten CPU-Ressourcenauslastungsrechte nicht überschreiten, obwohl die ausgewählte Konfiguration dazu führen kann, dass die CPU-Ressourcen des Hosts oder der logischen Partition nicht ausgelastet sind. CPU-Anteile geben Ihnen nicht diesen Grad an Kontrolle, der über einen CPU-Grenzwert erzielt werden kann.

In den folgenden Abschnitten werden die Features und Funktionalität des CPU-Grenzwerts ausführlicher beschrieben. Ein Einsatzszenarioabschnitt veranschaulicht die Features und Funktionalität des CPU-Grenzwerts mit Einsatzbeispielen.

Features und Funktionalität

Mit dem CPU-Grenzwert können Sie nach der Arbeit in einer Serviceklasse für das CPU-Ressourcenauslastungsrecht einen festen Grenzwert festlegen. Wenn für alle

Serviceklassen ein CPU-Grenzwert festgelegt wird, können Sie einen Teil der CPU-Ressourcen für die Durchführung von Arbeit ungeachtet anderer im DB2-Datenbankmanager ausgeführter Arbeit reservieren. Das Konfigurieren eines CPU-Grenzwerts für eine Serviceklasse liefert eine erzwungene Sandbox für Ihre Workloads, in der die CPU-Ressourcenauslastung zwischen Workloads ausgeglichen ist, allerdings sind die CPU-Ressourcen gelegentlich nicht vollständig ausgelastet.

Der CPU-Grenzwert ist außerdem nützlich, wenn Sie über mehrere DB2-Instanzen auf Ihrem Host oder ihrer logischen Partition verfügen. Da der Workload-Management-Dispatcher auf der Instanzebene fungiert, wird die CPU-Ressourcenzuordnung einer Serviceklasse aus den Anteilen dieser Serviceklasse relativ zu den Anteilen aller anderen Serviceklassen in der Instanz berechnet. Ein CPU-Grenzwert wird jedoch als Prozentsatz der CPU-Ressourcen des Hosts oder der logischen Partition ausgedrückt, unabhängig davon, wie viele DB2-Instanzen auf einem derartigen Host oder einer derartigen logischen Partition vorhanden sind. Durch die Anwendung von CPU-Grenzwerten auf Ihre Superklassen und von Anteilen auf Ihre Unterklassen können Sie mit dem CPU-Grenzwert das absolute CPU-Ressourcenauslastungsrecht jeder Superklasse und somit auch die Instanz steuern und dann mit Anteilen die relativen CPU-Ressourcenauslastungsrechte von Serviceunterklassen steuern, die in diesen Superklassen ausgeführt werden.

Ein CPU-Grenzwert kann auf der Ebene der Servicesuperklasse konfiguriert werden, auf der er einen prozentualen Grenzwert des CPU-Ressourcenauslastungsrechts auf dem Host oder der logischen Partition aller Unterklassen in dieser Superklasse darstellt. Er kann auch auf der Unterklassenebene konfiguriert werden, auf der er einen prozentualen Grenzwert des CPU-Ressourcenauslastungsrechts auf dem Host oder der logischen Partition dieser bestimmten Unterklasse darstellt.

Zum Aktivieren des Attributs für den CPU-Grenzwert müssen Sie den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers auf ON setzen. Die Standardeinstellung für diesen Parameter ist OFF. Durch die Aktivierung des Workload-Management-Dispatchers wird die CPU-Ressourcensteuerung über das Attribut für den CPU-Grenzwert standardmäßig zur Verfügung gestellt. Sie können CPU-Grenzwerte über die Anweisungen CREATE SERVICE CLASS und ALTER SERVICE CLASS zuweisen und anpassen. Vollständige Details zum Festlegen von CPU-Grenzwerten finden Sie in „Festlegen eines CPU-Grenzwerts“ auf Seite 228.

Der Workload-Management-Dispatcher respektiert beim Zuordnen von CPU-Ressourcen zu Serviceklassen immer den restriktivsten CPU-Grenzwert oder die restriktivsten Zuweisungen von CPU-Anteilen. Wenn z. B. ein CPU-Grenzwert auf der Superklassen- und Unterklassenebene festgelegt ist, wird der restriktivere CPU-Grenzwert berücksichtigt. Wenn eine Serviceklasse ihren CPU-Grenzwert erreicht, bevor sie ihr anteilsbasiertes CPU-Ressourcenauslastungsrecht vollständig verwendet hat, respektiert der Dispatcher den CPU-Grenzwert.

Einsatzszenarios

CPU-Grenzwert und mehrere Superklassen

Diese Gruppe von Verwendungsbeispielen veranschaulicht das CPU-Grenzwertverhalten in einer Umgebung mit mehreren Superklassen.

Abb. 34 auf Seite 216 zeigt einen Host oder eine logische Partition, der bzw. die mit den beiden Superklassen A und B konfiguriert ist. Es wird zum Veranschaulichen der Basiskonzepte davon ausgegangen, dass in den Standardbenutzer-, Wartungs-

und Systemserviceklassen unbedeutende Arbeit ausgeführt wird. In den folgenden Szenarios gibt es nur eine DB2-Instanz mit einer Datenbank und nur einem Member auf diesem Host oder dieser logischen Partition.

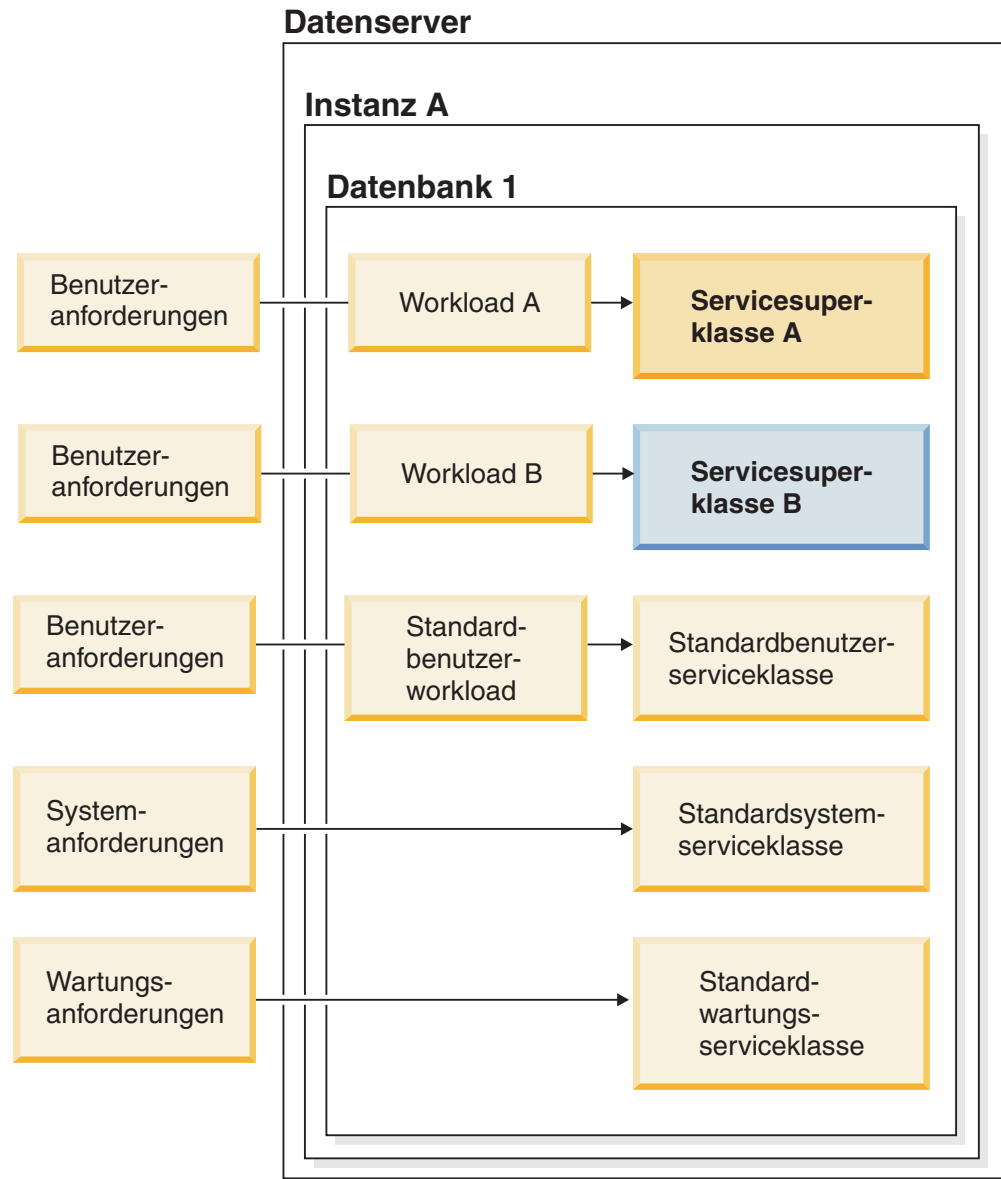


Abbildung 34. Datenserverkonfiguration: Mehrere Superklassen

CPU-Grenzwert und mehrere Superklassen: Szenario 1

In diesem Beispiel zeigt die Anzeige A in Abb. 35 auf Seite 217, dass für die Serviceklasse A ein CPU-Grenzwert von 30 % und für die Serviceklasse B kein CPU-Grenzwert definiert ist. Am Anfang dieses Szenarios verfügt die Serviceklasse A über genügend Arbeit, um eine CPU-Ressourcenauslastung von 30 % zu erzielen, und die Serviceklasse B verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 70 % zu erzielen. Für die Serviceklassen A und B sind 1.000 veränderliche CPU-Anteile konfiguriert.

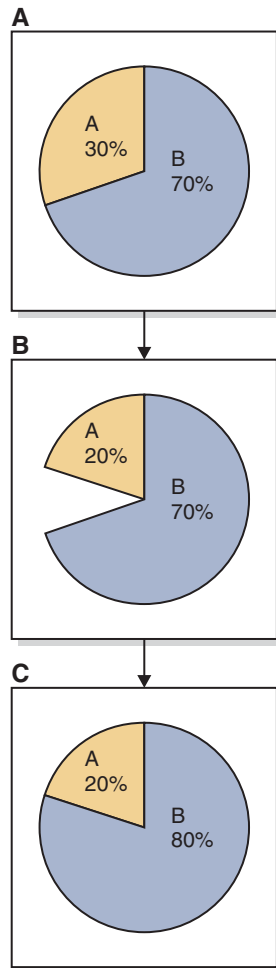


Abbildung 35. CPU-Grenzwert und mehrere Superklassen: Szenario 1

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceklasse A von 30 % auf 20 % gesunken ist. Die Serviceklasse B verfügt über genügend CPU-Ressourcenbedarf, um die von der Serviceklasse A vorübergehend freigegebenen CPU-Ressourcen zu übernehmen, wie in der Anzeige C gezeigt wird.

CPU-Grenzwert und mehrere Superklassen: Szenario 2

In diesem Beispiel zeigt die Anzeige A in Abb. 36 auf Seite 218 erneut, dass für die Serviceklasse A ein CPU-Grenzwert von 30 % und für die Serviceklasse B kein CPU-Grenzwert definiert ist. Am Anfang dieses Szenarios verfügt die Serviceklasse A über genügend Arbeit, um eine CPU-Ressourcenauslastung von 30 % zu erzielen, und die Serviceklasse B verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 70 % zu erzielen. Für die Serviceklassen A und B sind 1.000 veränderliche CPU-Anteile konfiguriert.

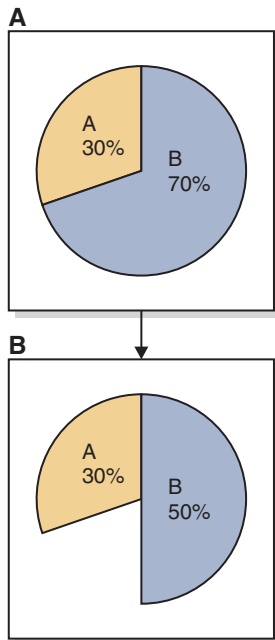


Abbildung 36. CPU-Grenzwert und mehrere Superklassen: Szenario 2

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceklasse B von 70 % auf 50 % gesunken ist. Aufgrund ihres CPU-Grenzwerts kann die Serviceklasse A nicht die 20 % der CPU-Ressourcen verbrauchen, die die Serviceklasse B vorübergehend freigegeben hat. Die CPU-Gesamtauslastung für den Host oder die logische Partition bleibt bei 80 %.

CPU-Grenzwert und mehrere Superklassen: Szenario 3

In diesem Beispiel zeigt die Anzeige A in Abb. 37 auf Seite 219 erneut, dass für die Serviceklasse A ein CPU-Grenzwert von 30 % und für die Serviceklasse B kein CPU-Grenzwert definiert ist. Am Anfang dieses Szenarios verfügt die Serviceklasse A über genügend Arbeit, um eine CPU-Ressourcenauslastung von 30 % zu erzielen, und die Serviceklasse B verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 70 % zu erzielen. Für die Serviceklassen A und B sind 1.000 veränderliche CPU-Anteile konfiguriert.

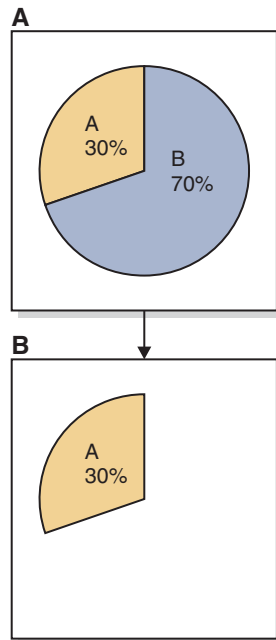


Abbildung 37. CPU-Grenzwert und mehrere Superklassen: Szenario 3

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceklasse B von 70 % auf 0 % gesunken ist. Aufgrund ihres CPU-Grenzwerts kann die Serviceklasse A nicht die 70 % der CPU-Ressourcen verbrauchen, die die Serviceklasse B vorübergehend freigegeben hat. Die CPU-Gesamtauslastung für den Host oder die logische Partition bleibt bei 30 %.

CPU-Grenzwert und mehrere Unterklassen

Diese nächste Gruppe von Verwendungsbeispielen veranschaulicht das CPU-Grenzwertverhalten in einer Umgebung mit mehreren Unterklassen.

Abb. 38 auf Seite 220 zeigt einen Host oder eine logische Partition, der bzw. die mit den beiden Superklassen A und B konfiguriert ist. Die Servicesuperklasse A enthält die Serviceunterklassen A1 und A2. Es wird zum Veranschaulichen der Basiskonzepte davon ausgegangen, dass in den Standardbenutzer-, Wartungs- und System-serviceklassen unbedeutende Arbeit ausgeführt wird. In den folgenden Szenarios gibt es nur eine DB2-Instanz mit einer Datenbank und nur einer logischen Partition auf diesem Host oder dieser logischen Partition.

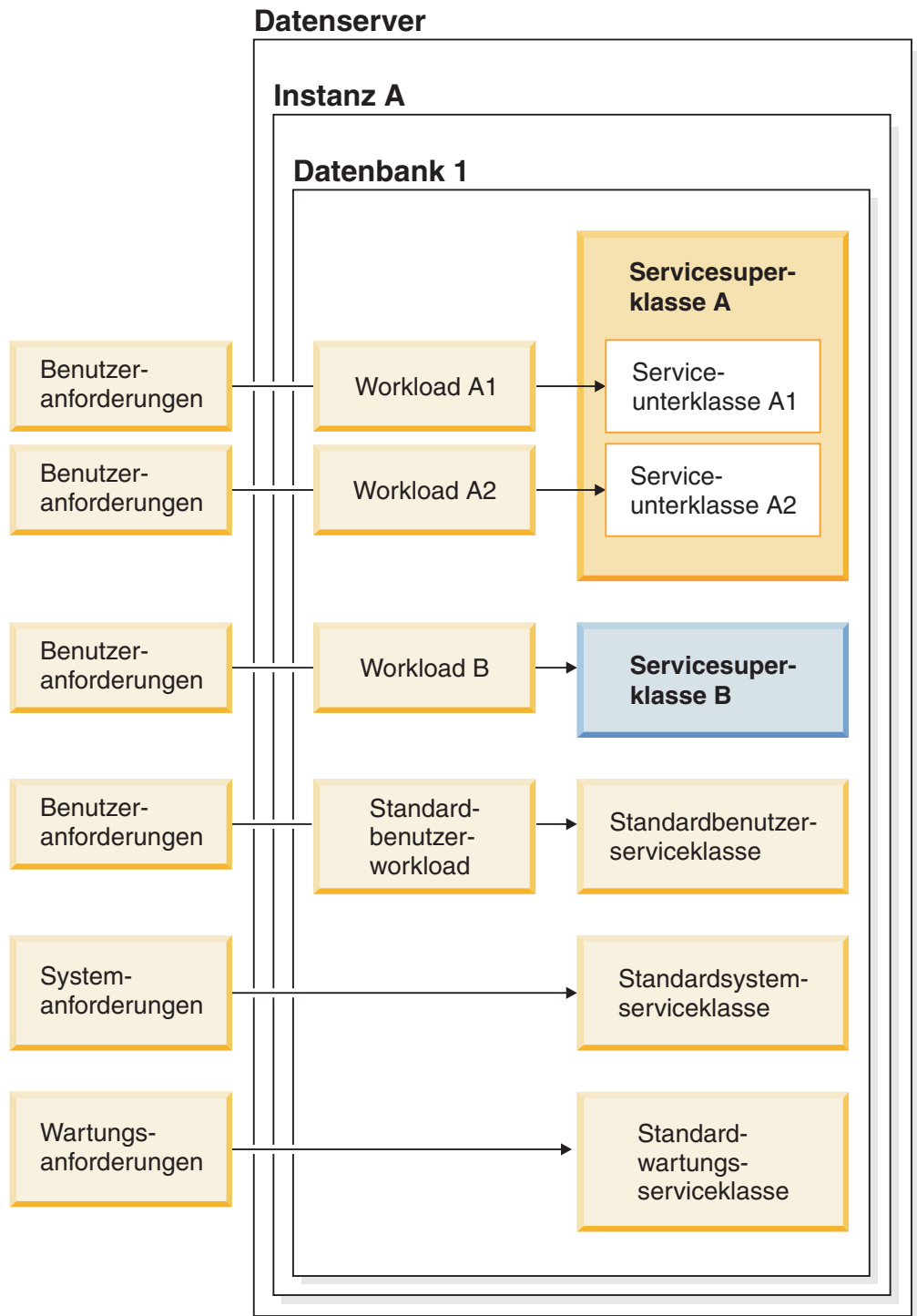


Abbildung 38. Datenserverkonfiguration: Mehrere Unterklassen

CPU-Grenzwert und mehrere Unterklassen: Szenario 1

In diesem Beispiel zeigt die Anzeige A in Abb. 39 auf Seite 221, dass für die Servicesuperklasse A ein CPU-Grenzwert von 50 % und für die Serviceunterklasse A1 ein CPU-Grenzwert von 20 % definiert ist. Für die Servicesuperklasse B ist kein CPU-Grenzwert definiert. Am Anfang dieses Szenarios verfügt die Serviceunterklasse A1 über genügend Arbeit, um eine CPU-Ressourcenauslastung von 20 % zu erzielen, und die Serviceunterklasse A2 verfügt über genügend Arbeit, um eine

CPU-Ressourcenauslastung von 30 % zu erzielen, was für die Servicesuperklasse A eine CPU-Gesamtr ressourcenauslastung von 50 % ergibt. Die Servicesuperklasse B verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 50 % zu erzielen. Für die Servicesuperklassen A und B sind 1.000 veränderliche CPU-Anteile konfiguriert.

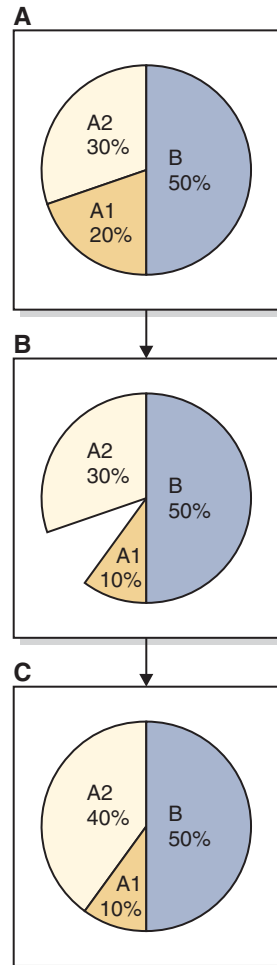


Abbildung 39. CPU-Grenzwert und mehrere Unterklassen: Szenario 1

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceunterklasse A1 von 20 % auf 10 % gesunken ist. Die Serviceunterklasse A2 und die Servicesuperklasse B verfügen jeweils über genügend CPU-Bedarf, um die gesamten nicht verwendeten CPU-Ressourcen zu übernehmen, die die Serviceklasse A1 vorübergehend freigegeben hat. Der Serviceunterklasse A2 sind jedoch alle freigegebenen CPU-Ressourcen zugeordnet und sie erhöht ihre CPU-Auslastung von 30 % auf 40 %, wie in der Anzeige C dargestellt wird. Dieses CPU-Ressourcenzuordnungsergebnis ergibt sich daraus, dass sich die Servicesuperklassen A und B mit jeweils 50 % bereits die CPU-Gesamtr ressourcen teilen, weil jeder Superklasse 1.000 veränderliche CPU-Anteile zugewiesen werden.

CPU-Grenzwert und mehrere Unterklassen: Szenario 2

In diesem Beispiel zeigt die Anzeige A in Abb. 40 auf Seite 222 erneut, dass für die Servicesuperklasse A ein CPU-Grenzwert von 50 % und für die Serviceunterklasse A1 ein CPU-Grenzwert von 20 % definiert ist. Für die Servicesuperklasse B ist kein

CPU-Grenzwert definiert. Am Anfang dieses Szenarios verfügt die Serviceunterklasse A1 über genügend Arbeit, um eine CPU-Ressourcenauslastung von 20 % zu erzielen, und die Serviceunterklasse A2 verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 30 % zu erzielen, was für die Servicesuperklasse A eine CPU-Gesamtr ressourcenauslastung von 50 % ergibt. Die Servicesuperklasse B verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 50 % zu erzielen. Für die Servicesuperklassen A und B sind 1.000 veränderliche CPU-Anteile konfiguriert.

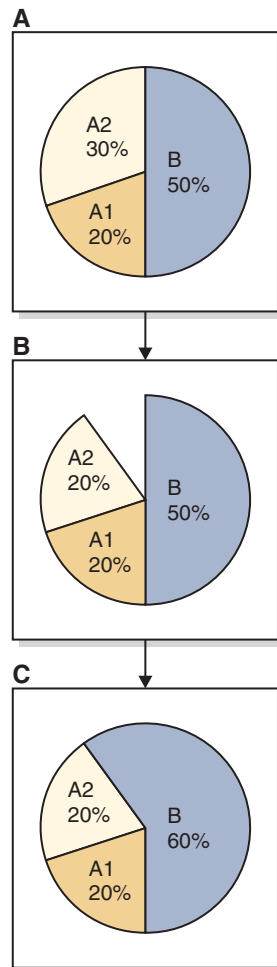


Abbildung 40. CPU-Grenzwert und mehrere Unterklassen: Szenario 2

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceunterklasse A2 von 30 % auf 20 % gesunken ist. Da für die Serviceunterklasse A1 ein CPU-Grenzwert von 20 % definiert ist, kann sie ihre aktuelle CPU-Auslastung von 20 % nicht überschreiten. Die Servicesuperklasse A senkt ihre CPU-Auslastung auf einen Gesamtbetrag von 40 %. Die Servicesuperklasse B verfügt über genügend CPU-Ressourcenbedarf, um die von der Serviceunterklasse A2 vorübergehend freigegebene CPU-Ressource zu übernehmen. Hierdurch wird die CPU-Auslastung der Servicesuperklasse B von 50 % auf 60 % erhöht, wie in der Anzeige C gezeigt wird.

CPU-Grenzwert und mehrere DB2-Instanzen

Diese nächste Gruppe von Verwendungsbeispielen veranschaulicht das CPU-Grenzwertverhalten in einer Umgebung mit DB2-Instanzen.

Abb. 41 zeigt einen Host oder eine logische Partition, der bzw. die mit zwei DB2-Instanzen konfiguriert ist, Instanz A und Instanz B. Jede Instanz enthält eine Datenbank, Datenbank 1 oder Datenbank 2. Jede Datenbank enthält die Servicesuperklasse A oder B. Die Servicesuperklasse A enthält die Serviceunterklassen A1 und A2. Die Servicesuperklasse B enthält die Serviceunterklassen B1 und B2. Benutzeranforderungen für jede Datenbank werden nur an die beiden Serviceunterklassen weitergeleitet und an die Standardserviceunterklasse wird keine Arbeit weitergeleitet, um das Szenario einfach zu halten. Außerdem wird zum Veranschaulichen der Basiskonzepte davon ausgegangen, dass in den Standardbenutzer-, Wartungs- und Systemserviceklassen unbedeutende Arbeit ausgeführt wird. In den folgenden Szenarios gibt es zwei DB2-Instanzen mit einer Datenbank in jeder Instanz und nur einem Member auf diesem Host oder dieser logischen Partition.

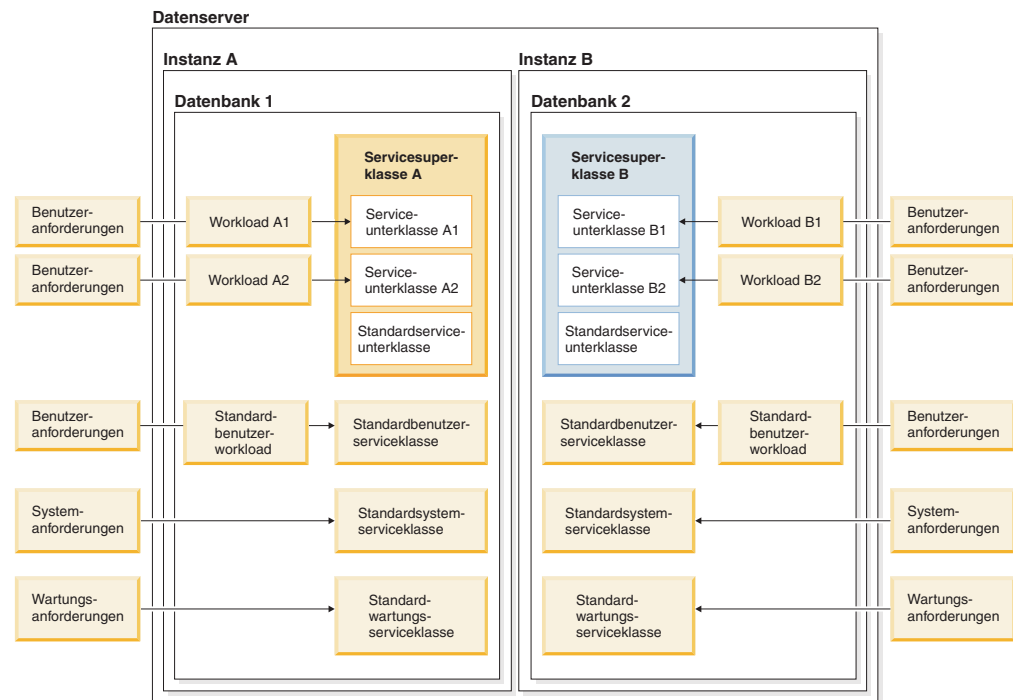


Abbildung 41. Datenserverkonfiguration: Mehrere DB2-Instanzen

CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 1

In diesem Beispiel sollen die Instanzen A und B jeweils nicht mehr als 50 % der CPU-Gesamtressourcen des Hosts oder der logischen Partition empfangen. In der Instanz A sollen 80 % der Instanz-CPU-Ressourcen der Serviceunterklasse A1 und 20 % der Serviceunterklasse A2 zugeordnet werden, indem A1 8.000 veränderliche CPU-Anteile und A2 2.000 veränderliche CPU-Anteile zugewiesen werden. Wenn die Serviceunterklasse A1 ihre vollständigen 80 % nicht verbraucht, soll die Serviceunterklasse A2 nicht verwendete CPU-Ressourcen übernehmen (und umgekehrt), um die CPU-Auslastung in der Instanz A zu maximieren. In der Instanz B sollen 60 % der Instanz-CPU-Ressourcen der Serviceunterklasse B1 und 40 % der Serviceunterklasse B2 zugeordnet werden, indem B1 6.000 veränderliche CPU-Anteile und B2 4.000 veränderliche CPU-Anteile zugewiesen werden. Wie bei den Serviceunterklassen A1 und A2 soll die CPU-Auslastung in der Instanz B maximiert werden, indem die Serviceunterklasse B1 nicht verwendete CPU-Ressourcen übernimmt, wenn die Serviceunterklasse B2 ihr vollständiges CPU-Ressourcenauslastungsrecht nicht verwendet (und umgekehrt).

Sie müssen die entsprechenden Bedingungen für dieses Szenario so wie im vorangehenden Absatz beschrieben konfigurieren. Begrenzen Sie hierzu die CPU-Ressourcen für jede Instanz, indem Sie einen CPU-Grenzwert von 50 % für die Servicesuperklasse A in der Datenbank 1 in der Instanz A und einen CPU-Grenzwert von 50 % für die Servicesuperklasse B in der Datenbank 2 in der Instanz B erstellen. Wenn beide Instanzen ihre gesamten CPU-Ressourcenauslastungsrechte verwenden, wird für den Host oder die logische Partition als Ganzes eine CPU-Auslastung von 100 % angenommen. Wenn eine der beiden Instanzen ihr vollständiges CPU-Nutzungsrecht nicht verwendet, kann die andere Instanz die nicht verwendeten CPU-Ressourcen nicht übernehmen.

Für jede Servicesuperklasse in jeder Instanz teilt der Workload-Management-Dispatcher die verfügbaren CPU-Ressourcen über ihre relativen Zuweisungen von veränderlichen CPU-Anteilen unter den Unterklassen auf. Durch die Verwendung von veränderlichen CPU-Anteilen, die Serviceunterklassen zugewiesen sind, verwendet eine Instanz ihr CPU-Ressourcenauslastungsrecht nur dann nicht vollständig, wenn in jeder Unterklasse in der Instanz nicht genügend Arbeit ausgeführt wird, um die vollständige CPU-Auslastung ihres Nutzungsrechts zu erzielen.

Die Anzeige A in Abb. 42 auf Seite 225 zeigt, dass für die Servicesuperklassen A und B ein CPU-Grenzwert von jeweils 50 % definiert ist. Die Serviceunterklasse A1 verfügt über 80 % der CPU-Ressourcen der Instanz A (40 % der CPU-Ressourcen des Hosts oder der logischen Partition), während die Serviceunterklasse A2 über die verbleibenden 20 % (10 % der CPU-Ressourcen des Hosts oder der logischen Partition) verfügt. Die Serviceunterklasse B1 verfügt über 60 % der CPU-Ressourcen der Instanz B (30 % der CPU-Ressourcen des Hosts oder der logischen Partition), während die Serviceunterklasse B2 über die verbleibenden 40 % (20 % der CPU-Ressourcen des Hosts oder der logischen Partition) verfügt. Am Anfang dieses Szenarios verfügt die Serviceunterklasse A1 über genügend Arbeit, um eine CPU-Ressourcenauslastung von 40 % für den Host oder die logische Partition zu erzielen, und die Serviceunterklasse A2 verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 10 % für den Host oder die logische Partition zu erzielen, was für die Servicesuperklasse A eine CPU-Gesamtr ressourcenauslastung von 50 % ergibt. Die Serviceunterklasse B1 verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 30 % für den Host oder die logische Partition zu erzielen, und die Serviceunterklasse B2 verfügt über genügend Arbeit, um eine CPU-Ressourcenauslastung von 20 % für den Host oder die logische Partition zu erzielen, was für die Servicesuperklasse B eine CPU-Gesamtr ressourcenauslastung von 50 % ergibt.

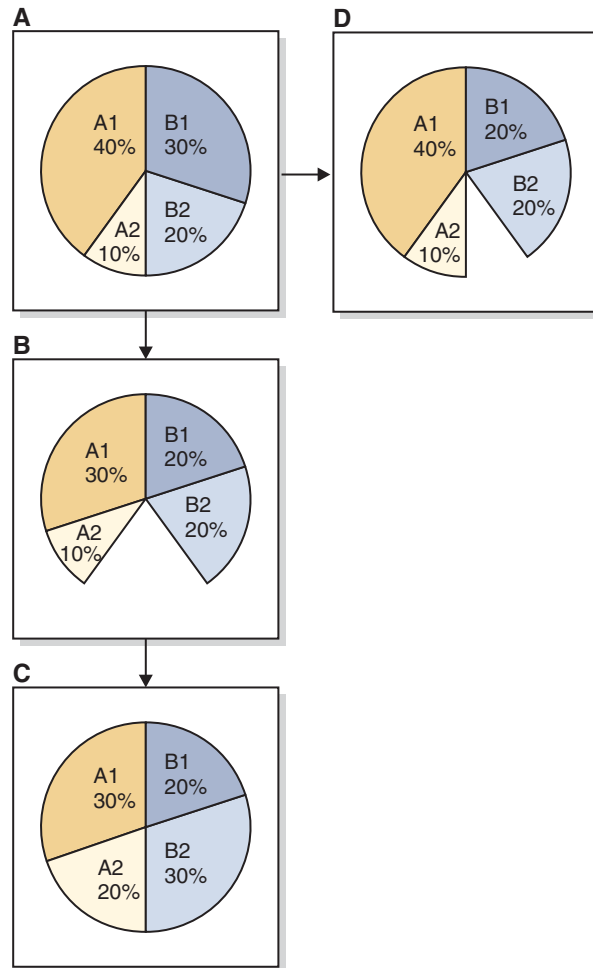


Abbildung 42. CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 1

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceunterklasse A1 von 40 % auf 30 % gesunken ist, und der CPU-Ressourcenbedarf der Serviceunterklasse B1 ist von 30 % auf 20 % gesunken. Wenn in der Serviceunterklasse A2 genügend Arbeit ausgeführt wird, um nicht verwendete CPU-Ressourcen zu übernehmen, die verfügbar werden, erhöht A2 ihre CPU-Auslastung von 10 % auf 20 %, weil A2 veränderliche CPU-Anteile zugewiesen sind, wie in der Anzeige C gezeigt wird. Dies gilt auch für die Serviceunterklasse B2, die ihre CPU-Auslastung von 20 % auf 30 % erhöht, wie in der Anzeige C gezeigt wird.

Nun soll die ursprüngliche Startbedingung wie in der Anzeige A beschrieben für ein anderes Beispiel betrachtet werden. Wenn die Serviceunterklasse B1 ihren CPU-Bedarf von 30 % auf 20 % reduziert und wenn in der Serviceunterklasse B2 nicht genügend Arbeit ausgeführt wird, um einen CPU-Bedarf von 20 % zu überschreiten, verwendet die Servicesuperklasse B in der Instanz B ihr vollständiges CPU-Nutzungsrecht von 50 % nicht und behält eine CPU-Auslastung von 40 % bei, wie in der Anzeige D gezeigt wird. Deshalb beträgt die CPU-Auslastung des Hosts oder der logischen Partition lediglich 90 % der CPU-Gesamtresourcen.

CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 2

Dieses Szenariobeispiel veranschaulicht, was passiert, wenn die Summe der CPU-Grenzwerte der Serviceunterklassen den CPU-Grenzwert der übergeordneten Servicesuperklasse nicht überschreitet.

Nun sollen unter ähnlichen Bedingungen wie in „CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 1“ nur die veränderlichen CPU-Anteile, die den Serviceunterklassen A1 und A2 zugewiesen sind, in CPU-Grenzwerte von 40 % bzw. 10 % der CPU-Gesamtressourcen für den Host oder die logische Partition geändert werden, wie in der Anzeige A in Abb. 43 gezeigt wird. Wenn beide Serviceunterklassen die CPU-Ressourcen bis zu ihren zugewiesenen CPU-Grenzwerten verwenden, beträgt die CPU-Gesamtauslastung für die Servicesuperklasse A 50 %, wodurch die zusätzliche Einschränkung eines CPU-Grenzwerts von 50 % für die Servicesuperklasse A entfällt.

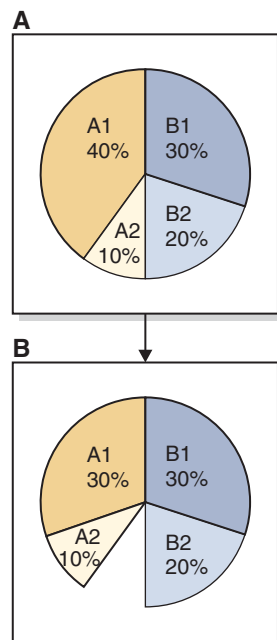


Abbildung 43. CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 2

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceunterklasse A1 aufgrund einer Abnahme der in der Serviceunterklasse ausgeführten Arbeit von 40 % auf 30 % gesunken ist. Unter diesen Umständen kann der Dispatcher die nicht verwendete CPU-Ressource, die von der Serviceunterklasse A1 vorübergehend freigegeben wurde, der Serviceunterklasse A2 nicht zuordnen. Die Serviceunterklasse A2 führt Workloads mit dem CPU-Grenzwert von 10 % der CPU-Ressourcen des Hosts oder der logischen Partition weiter aus. Daher kann die Instanz A ihr vollständiges CPU-Ressourcenauslastungsrecht von 50 % nicht verwenden.

CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 3

Dieses Szenariobeispiel veranschaulicht, was passiert, wenn die Summe der CPU-Grenzwerte der Serviceunterklassen den CPU-Grenzwert der übergeordneten Servicesuperklasse überschreitet.

Nun sollen unter ähnlichen Bedingungen wie in „CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 1“ nur die veränderlichen CPU-Anteile, die den Serviceunterklassen A1 und A2 zugewiesen sind, in CPU-Grenzwerte von jeweils 40 % der CPU-Gesamtressourcen für den Host oder die logische Partition geändert werden. In diesem Beispiel beträgt die Gesamtsumme der zugewiesenen CPU-Grenzwerte für die Serviceunterklassen A1 und A2 80 %, was den CPU-Grenzwert von 50 % überschreitet, der der Servicesuperklasse A zugewiesen ist. Der Workload-Management-Dispatcher hindert die Servicesuperklasse A am Überschreiten ihres CPU-Grenzwert von 50 %. Das CPU-Ressourcenvolumen, das jeder Serviceunterklasse in der Superklasse A zugeordnet wird, wird von den CPU-Anteilen bestimmt, die den Unterklassen zugewiesen wurden. CPU-Anteile wurden den Serviceunterklassen A1 und A2 nicht explizit zugewiesen, jeder dieser Unterklassen wurden bei ihrer Erstellung jedoch 1.000 veränderliche CPU-Anteile zugewiesen, wodurch jede Unterklasse ein gleiches CPU-Nutzungsrecht erhält. Der Dispatcher ordnet eine gleiche Aufteilung der 50 % der CPU-Gesamtressourcen für den Host oder die logische Partition zu, zu denen die Servicesuperklasse A berechtigt ist. Somit werden den Serviceunterklassen A1 und A2 jeweils 25 % zugeordnet, wie in der Anzeige A in Abb. 44 gezeigt wird.

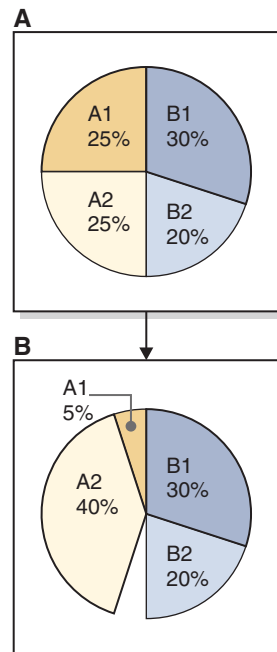


Abbildung 44. CPU-Grenzwert und mehrere DB2-Instanzen: Szenario 3

Die Anzeige B zeigt, dass der CPU-Ressourcenbedarf der Serviceunterklasse A1 aufgrund einer Abnahme der in der Serviceunterklasse ausgeführten Arbeit von 25 % auf 5 % gesunken ist. Aufgrund der Zuweisung von veränderlichen CPU-Anteilen kann der Dispatcher die nicht verwendete CPU-Ressource, die von der Serviceunterklasse A1 vorübergehend freigegeben wurde, bis zu 40 % der Serviceunterklasse A2 zuordnen. Die Serviceunterklasse A2 kann ihren CPU-Grenzwert von 40 % der CPU-Ressourcen des Hosts oder der logischen Partition nicht überschreiten. Daher kann die Instanz A ihr vollständiges CPU-Ressourcenauslastungsrecht von 50 % nicht verwenden.

Anmerkung: Diese Methode für die Verwaltung von Arbeit in mehreren Instanzen auf demselben Host oder in derselben logischen Partition beschränkt Sie auf eine

Servicesuperklasse in jeder Instanz. Bei Betriebssystemen mit einem Workload-Manager, in den das DB2-Workload-Management integriert werden kann, können Sie die DB2-Serviceklassen jeder Instanz einer Serviceklasse in einem Betriebssystem-Workload-Manager (wie AIX-WLM und Linux-WLM) zuordnen und die festen Grenzwerte des Betriebssystem-Workload-Managers jeder Betriebssystemserviceklasse zuweisen, um für die CPU-Ressourcenauslastung jeder Instanz Obergrenzen festzulegen.

Weitere Informationen

Zu den folgenden Workload-Management-Dispatcher-Themen finden Sie an den angegebenen Stellen vollständige Details:

- Workload-Management-Dispatcher: „Workload-Management-Dispatcher“ auf Seite 176.
- Dispatcher-Thread-Parallelität: „Dispatcher-Thread-Parallelität“ auf Seite 233.
- Minimale CPU-Auslastungsebene zum Klassifizieren einer Serviceklasse als aktiv: „Minimale CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 230.
- Feste CPU-Anteile: „Feste CPU-Anteile“ auf Seite 200.
- Veränderliche CPU-Anteile: „Veränderliche CPU-Anteile“ auf Seite 208.
- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers: „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“ auf Seite 236.

Festlegen eines CPU-Grenzwerts

Legen Sie das Attribut für den CPU-Grenzwert über die Anweisung `CREATE SERVICE CLASS` fest, um eine neue Serviceklasse und einen neuen CPU-Grenzwert zu erstellen. Mit der Anweisung `ALTER SERVICE CLASS` können Sie die Einstellung des CPU-Grenzwerts für eine vorhandene Serviceklasse ändern. Sie legen den CPU-Grenzwert fest, um die CPU-Auslastung durch eine Serviceklasse während der Steuerung durch den Workload-Management-Dispatcher unter allen Umständen zu beschränken.

Vorbereitende Schritte

Sie müssen den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter `wlm_dispatcher` des Datenbankmanagers konfigurieren, sofern noch nicht geschehen. Durch die Aktivierung des Workload-Management-Dispatchers wird die CPU-Ressourcensteuerung über das Attribut für den CPU-Grenzwert standardmäßig zur Verfügung gestellt. Weitere Informationen finden Sie in „Aktivieren des Workload-Manager-Dispatchers“ auf Seite 198.

Informationen zu diesem Vorgang

Der CPU-Grenzwert wird in dieser Task mit der Anweisung `CREATE SERVICE CLASS` oder `ALTER SERVICE CLASS` über den Befehlszeilenprozessor für eine Serviceklasse festgelegt.

Einschränkungen

Sie können CPU-Grenzwerte nur Serviceklassen zuweisen, die über den Workload-Management-Dispatcher gesteuert werden können, d. h. Benutzer- und Wartungserviceklassen. Die CPU-Ressourcenzuordnungen für Systemserviceklassen können vom Workload-Management-Dispatcher nicht gesteuert werden.

Vorgehensweise

Gehen Sie wie folgt vor, um eine neue Serviceklasse zu erstellen und den CPU-Grenzwert festzulegen (fahren Sie mit Schritt 2 fort, wenn eine vorhandene Einstellung für den CPU-Grenzwert der Serviceklasse geändert werden soll):

1. Setzen Sie die folgende Anweisung `CREATE SERVICE CLASS` ab, um eine neue Serviceklasse zu erstellen und den CPU-Grenzwert auf 25 % zu setzen:

```
create service class serviceklassenname cpu limit 25
```

Gehen Sie wie folgt vor, um die Einstellung des CPU-Grenzwerts für eine vorhandene Serviceklasse zu ändern:

2. Setzen Sie die folgende Anweisung `ALTER SERVICE CLASS` ab, um den CPU-Grenzwert in 50 % zu ändern:

```
alter service class serviceklassenname cpu limit 50
```

Ergebnisse

Sie haben die Einstellung des CPU-Grenzwerts für die angegebene Serviceklasse festgelegt oder geändert. Der Workload-Management-Dispatcher kann dieser Serviceklasse über den festgelegten Grenzwert hinaus keine CPU-Ressourcen zuordnen.

Nächste Schritte

Führen Sie nach der Aktivierung des Workload-Management-Dispatchers zum Verwalten von CPU-Ressourcen und nach der Festlegung des CPU-Grenzwerts die folgenden Tasks aus:

- Legen Sie die Dispatcher-Thread-Parallelität durch Konfigurieren des Konfigurationsparameters `wlm_disp_concur` des Datenbankmanagers fest. Weitere Informationen finden Sie in „Festlegen der Dispatcher-Thread-Parallelität“ auf Seite 235.
- Aktivieren Sie feste und veränderliche CPU-Anteile, indem Sie den Konfigurationsparameter `wlm_disp_cpu_shares` des Datenbankmanagers konfigurieren. Weitere Informationen finden Sie in Schritt 1 in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie einen festen oder veränderlichen CPU-Anteil über die Anweisung `CREATE SERVICE CLASS` oder `ALTER SERVICE CLASS`. Weitere Informationen finden Sie in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Legen Sie den minimalen CPU-Auslastungsprozentsatz für den Workload-Management-Dispatcher zur Aktivierung von Serviceklassen fest, indem Sie den Konfigurationsparameter `wlm_disp_min_util` des Datenbankmanagers festlegen. Weitere Informationen finden Sie in „Festlegen der minimalen CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 232.

Minimale CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv

Über den Konfigurationsparameter `wlm_disp_min_util` des Datenbankmanagers können Sie die minimale Ebene der CPU-Ressourcenauslastung steuern, auf der der Workload-Management-Dispatcher die Serviceklasse als aktiv an der Ausführung von Arbeit beteiligt klassifiziert. Nur die CPU-Anteile der aktiven Serviceklassen werden bei der vom Dispatcher ausgeführten CPU-Ressourcenzuordnungsplanung berücksichtigt.

Bei der Verwaltung von CPU-Ressourcenzuordnungen auf der Basis von Anteilen klassifiziert der Workload-Management-Dispatcher eine Serviceklasse als aktiv und berücksichtigt ihre gesamte Zuweisung von CPU-Anteilen bei der Gesamtzuordnung der CPU-Ressourcenplanung, wenn ein Teil der CPU-Ressource von Datenbankankorderungen verwendet wird, die in dieser Serviceklasse ausgeführt werden. In bestimmten Fällen wollen Sie steuern, wieviel der von der Serviceklasse generierten CPU-Aktivität für den Workload-Management-Dispatcher benötigt wird, um die CPU-Anteile der Serviceklasse während der CPU-Ressourcenplanung einzuschließen.

Im Beispielszenario, das in der Anzeige A in Abb. 45 auf Seite 231 dargestellt ist, enthält die Serviceklasse A eine transaktionsorientierte Workload mit hoher Priorität, die nur während des Arbeitstags ausgeführt wird, während die Serviceklassen B und C kontinuierlich ausgeführte Stapeljobs mit niedriger Priorität enthalten. Die Serviceklasse A wird durch das Zuweisen von festen CPU-Anteilen zu den Serviceklassen B und C vor Unterbrechungen geschützt. Nachts, wenn die transaktionsorientierte Workload der Serviceklasse A nicht vorhanden ist (wie in der Anzeige B gezeigt), können die Serviceklassen B und C die CPU-Ressourcen vollständig verwenden und schnellere Fortschritte machen (wie in der Anzeige C gezeigt).

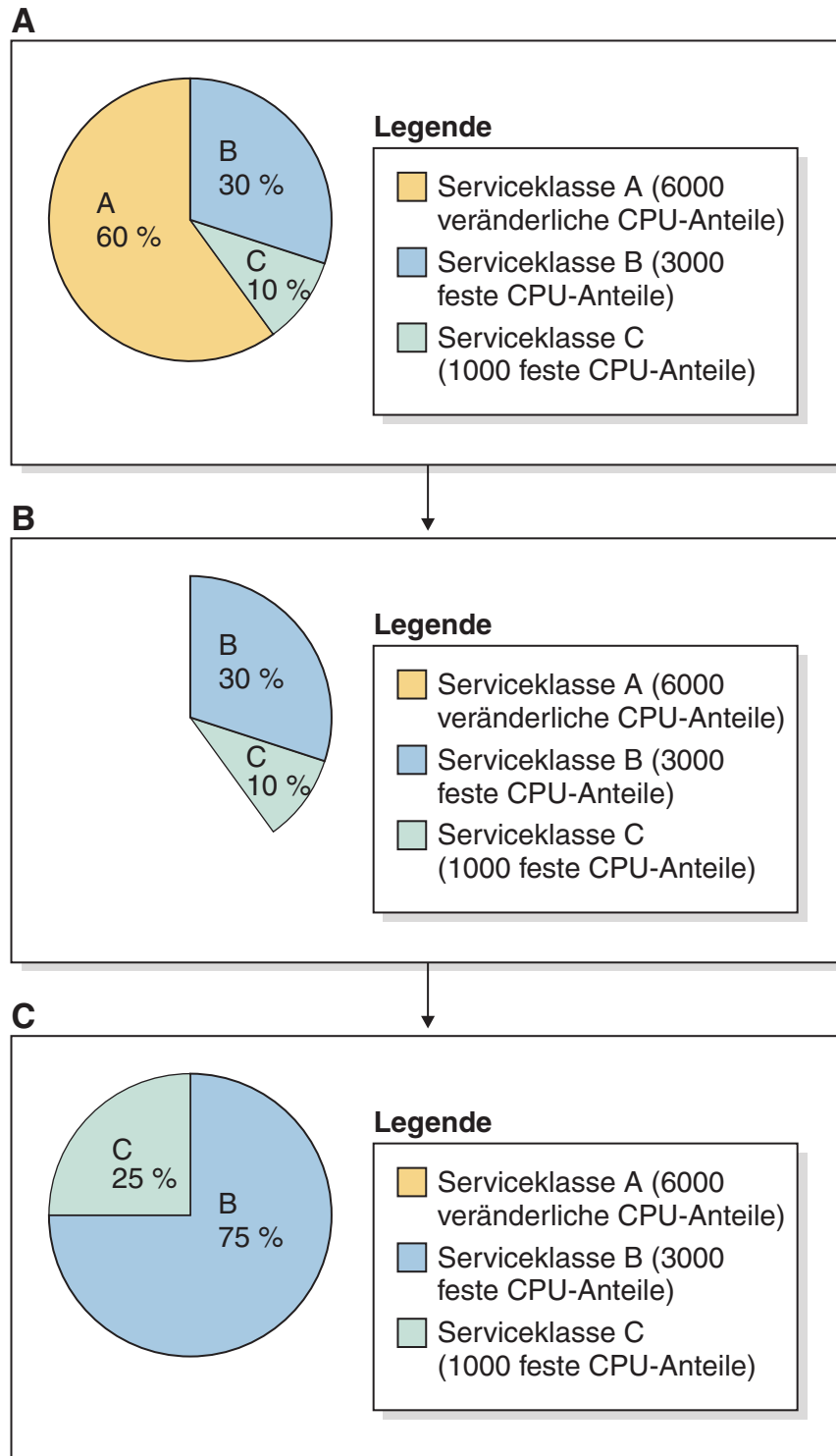


Abbildung 45. Beispiel für minimale CPU-Auslastung: Feste und veränderlich CPU-Anteile

Angenommen, nachts wird für die Serviceklasse A eine kleine Menge transaktionsorientierter Arbeitsvorgänge ausgeführt. In diesem Fall wird die Serviceklasse A vom Workload-Management-Dispatcher als aktiv klassifiziert, und die nicht verwendeten CPU-Ressourcen, die von der Serviceklasse A vorübergehend freigegeben werden, stehen den Serviceklassen B und C nicht zur Verfügung. Die CPU-Ressourcenauslastungsrechte für die Serviceklassen B und C werden in der Anzeige B

als kleine Scheiben im Kreisdiagramm für die CPU-Auslastung angezeigt, das die Aktivität der Serviceklasse A darstellt (im Kreisdiagramm der Anzeige B nicht angezeigt). Für die Serviceklassen B und C ist das Ergebnis wesentlich langsamer nächtlicher Fortschritt im Vergleich zum ursprünglichen Szenario, wie in der Anzeige C gezeigt wird.

Maximale Benutzerflexibilität zum Verwalten der DB2-Workloads wird über die Option zum Festlegen eines Prozentsatzes der CPU-Auslastung bereitgestellt, an oder über dem eine Serviceklasse als auf dem Host oder der logischen Partition aktiv klassifiziert wird. Wenn eine Serviceklasse als inaktiv eingestuft wird, wird ihre Zuweisung von CPU-Anteilen bei den Berechnungen der CPU-Ressourcenauslastungsrechte nicht berücksichtigt. Hierdurch können Serviceklassen, vor allem jene mit zugewiesenen festen CPU-Anteilen, die nicht verwendeten CPU-Ressourcen übernehmen. Dieser minimale Prozentsatz der CPU-Auslastung wird über das Konfigurieren des Prozentsatzwerts für den Konfigurationsparameter **wlm_disp_min_util** des Datenbankmanagers angegeben. Dieser Konfigurationsparameter kann auf einen Prozentsatzwert zwischen 0 und 100 gesetzt werden, wobei 5 der Standardwert ist. Diese Konfigurationsparametereinstellung ist nur wirksam, wenn Sie den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers auf YES setzen.

Das weiter oben beschriebene Szenario mit der kleinen Menge transaktionsorientierter Arbeitsvorgänge soll nun erneut betrachtet werden. Da der Prozentsatz des Konfigurationsparameters **wlm_disp_min_util** des Datenbankmanagers nun auf einen Wert gesetzt ist, der etwas höher ist als der kleine Anteil der nächtlichen CPU-Auslastung für die Serviceklasse A, kann davon ausgegangen werden, dass der Fortschritt der nächtlichen Stapeljobs der Serviceklassen B und C wesentlich verbessert wurde und dem des ursprünglichen Szenarios ähnelt, bei dem die CPU-Auslastung so wie in der Anzeige C dargestellt war.

Weitere Informationen

Zu den folgenden Workload-Management-Dispatcher-Themen finden Sie an den angegebenen Stellen vollständige Details:

- Workload-Management-Dispatcher: „Workload-Management-Dispatcher“ auf Seite 176.
- Dispatcher-Thread-Parallelität: „Dispatcher-Thread-Parallelität“ auf Seite 233.
- Feste CPU-Anteile: „Feste CPU-Anteile“ auf Seite 200.
- Veränderliche CPU-Anteile: „Veränderliche CPU-Anteile“ auf Seite 208.
- CPU-Grenzwert: „CPU-Grenzwert“ auf Seite 214.
- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers: „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“ auf Seite 236.

Festlegen der minimalen CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv

Sie legen die minimale CPU-Auslastung für eine Serviceklasse fest, indem Sie den Konfigurationsparameter **wlm_disp_min_util** des Datenbankmanagers auf einen Prozentsatz setzen. Serviceklassen, die CPU-Ressourcen verwenden, die größer-gleich dem Prozentsatz sind, werden auf dem Host oder auf der logischen Partition als aktiv eingestuft, und die CPU-Anteile der aktiven Serviceklassen werden bei den CPU-Zuordnungsberechnungen berücksichtigt.

Vorgehensweise

Gehen Sie wie folgt vor, um den Prozentsatz minimaler CPU-Auslastung zum Klassifizieren von Serviceklassen als aktiv festzulegen:

Setzen Sie den Befehl **UPDATE DATABASE MANAGER CONFIGURATION** (oder **UPDATE DBM CFG**) über den Befehlszeilenprozessor ab und setzen Sie den Konfigurationsparameter **wlm_disp_min_util** des Datenbankmanagers auf 6 %. Der Parameterwert wird sofort aktualisiert, indem zuerst eine Verbindung zur DB2-Instanz hergestellt wird:

```
attach to instanzname
update dbm cfg using wlm_disp_min_util 6
detach
```

Ergebnisse

Der Prozentsatz minimaler CPU-Auslastung zum Klassifizieren von Serviceklassen als aktiv wurde festgelegt. Serviceklassen mit einem kleineren CPU-Auslastungsprozentsatz als dieses Minimum werden als inaktiv eingestuft und ihre CPU-Zuordnungen können Serviceklassen, die freie CPU-Ressourcen übernehmen können, proportional erneut zugeordnet werden.

Nächste Schritte

Führen Sie nach der Festlegung des Prozentsatzes minimaler CPU-Auslastung zum Klassifizieren von Serviceklassen als aktiv durch den Workload-Management-Dispatcher die folgenden Tasks aus:

- Sie müssen den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers konfigurieren, damit die Einstellung der minimalen CPU-Auslastung wirksam wird. Weitere Informationen finden Sie in „Aktivieren des Workload-Manager-Dispatchers“ auf Seite 198.
- Legen Sie die Dispatcher-Thread-Parallelität durch Konfigurieren des Konfigurationsparameters **wlm_disp_concur** des Datenbankmanagers fest. Weitere Informationen finden Sie in „Festlegen der Dispatcher-Thread-Parallelität“ auf Seite 235.
- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie den CPU-Grenzwert über die Anweisung **CREATE SERVICE CLASS** oder **ALTER SERVICE CLASS**. Weitere Informationen finden Sie in „Festlegen eines CPU-Grenzwerts“ auf Seite 228.
- Aktivieren Sie feste und veränderliche CPU-Anteile, indem Sie den Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers konfigurieren. Weitere Informationen finden Sie in Schritt 1 in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie einen festen oder veränderlichen CPU-Anteil über die Anweisung **CREATE SERVICE CLASS** oder **ALTER SERVICE CLASS**. Weitere Informationen finden Sie in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.

Dispatcher-Thread-Parallelität

Der Workload-Management-Dispatcher steuert die Anzahl der Threads, die dem Betriebssystem vom Datenbankmanager gleichzeitig zugeteilt werden dürfen. Diese Anzahl wird als Dispatcher-Thread-Parallelität bezeichnet. Sie können die Dispatcher-Thread-Parallelität über den Konfigurationsparameter **wlm_disp_concur** des Datenbankmanagers festlegen. In diesem Abschnitt werden entsprechende Details geliefert.

Features und Funktionalität

Die Anzahl der gleichzeitig ausgeführten Agenten, die vom Workload-Management-Dispatcher verwaltet werden, sollte ein niedriges Vielfaches der Anzahl CPUs im System sein. Das Ziel ist es, eine Ebene für den gemeinsamen Zugriff zu erstellen, die gerade groß genug ist, um sicherzustellen, dass die CPU-Ressourcen auf dem Host oder der logischen Partition vollständig verwendet werden, und um Betriebssystemplanungs-Latenzen aufgrund des Umschaltens von Threads von aktiv in inaktiv und umgekehrt Rechnung zu tragen. Dieser optimale Wert stellt maximale Effizienz sicher und gibt dem Workload-Management-Dispatcher maximale Kontrolle über die CPU-Ressourcenzuordnung. Wenn der konfigurierte Wert für den gemeinsamen Zugriff zu klein ist, ist nicht genügend Arbeit vorhanden, um die verfügbaren CPU-Ressourcen vollständig zu verwenden. Wenn der konfigurierte Wert für den gemeinsamen Zugriff zu groß ist, hat der Dispatcher weniger Kontrolle über die auf dem System ausgeführte Arbeit, was den Ressourcenkonflikt verstärken kann.

Über den Konfigurationsparameter `wlm_disp_concur` des Datenbankmanagers können Sie angeben, wie der DB2-Datenbankmanager die Ebene für den gemeinsamen Zugriff festlegt (COMPUTED), oder Sie können die Ebene für den gemeinsamen Zugriff manuell auf einen festen Wert setzen.

Als allgemeine Richtlinie gilt, dass die optimale Dispatcher-Thread-Parallelität vier Mal die Anzahl der auf der Server-Hardware verfügbaren CPUs ist. Diese Richtlinie reicht in der Regel für die meisten Plattformen aus und ist der Standardwert (COMPUTED ist die Standardeinstellung für den Konfigurationsparameter). Vollständige Details zum Festlegen der Dispatcher-Thread-Parallelität finden Sie in „Festlegen der Dispatcher-Thread-Parallelität“ auf Seite 235.

Der angegebene Wert des Datenbankkonfigurationsparameters `wlm_disp_concur` gilt für jeden Host bzw. jede logische Partition. In Umgebungen mit Datenbanken mit mehreren Mitgliedern wird die angegebene Ebene für den gemeinsamen Zugriff zwischen den Mitgliedern gemeinsam genutzt.

Optimierung

Überwachen Sie Ihr System nach der Aktivierung des Workload-Management-Dispatchers auf eine Senkung des Durchsatzes und der CPU-Auslastung. Sinkt einer dieser beiden Messwerte, erhöhen Sie den Wert des Konfigurationsparameters in Inkrementen der Anzahl verfügbarer Prozessoren.

Weitere Informationen

Zu den folgenden Themen werden an den angegebenen Stellen vollständige Details geliefert:

- Workload-Management-Dispatcher: „Workload-Management-Dispatcher“ auf Seite 176.
- Veränderliche CPU-Anteile: „Veränderliche CPU-Anteile“ auf Seite 208.
- Feste CPU-Anteile: „Feste CPU-Anteile“ auf Seite 200.
- CPU-Grenzwert: „CPU-Grenzwert“ auf Seite 214.
- Minimale CPU-Auslastungsebene zum Klassifizieren einer Serviceklasse als aktiv: „Minimale CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 230.

- Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers: „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“ auf Seite 236.

Festlegen der Dispatcher-Thread-Parallelität

Sie legen die Dispatcher-Thread-Parallelität fest, indem Sie den Konfigurationsparameter **wlm_disp_concur** des Datenbankmanagers auf COMPUTED oder einen manuell festgelegten Wert setzen. Sie setzen die Dispatcher-Thread-Parallelität auf einen optimalen Wert, der maximale Effizienz sicherstellt und dem Workload-Management-Dispatcher die maximale Steuerung der CPU-Ressourcenzuordnungen ermöglicht.

Vorgehensweise

Gehen Sie wie folgt vor, um die Dispatcher-Thread-Parallelität festzulegen:

Setzen Sie über den Befehlszeilenprozessor den Befehl **UPDATE DATABASE MANAGER CONFIGURATION** (oder **UPDATE DBM CFG**) ab und setzen Sie den Konfigurationsparameter **wlm_disp_concur** des Datenbankmanagers auf einen der folgenden möglichen Werte:

- COMPUTED (Standardwert)
- *manuell_festgelegter_wert*

Das folgende Beispiel setzt die Dispatcher-Thread-Parallelität manuell auf 16, was 4 Mal die 4 auf dem System verfügbaren CPUs ist. Die Zuteilungsebene wird sofort aktualisiert, indem zuerst eine Verbindung zur DB2-Instanz hergestellt wird:

```
attach to instanzname
update dbm cfg using wlm_disp_concur 16
detach
```

Ergebnisse

Die Dispatcher-Thread-Parallelität wurde festgelegt. Wurde die Dispatcher-Thread-Parallelität auf COMPUTED gesetzt, wird sie vom DB2-Datenbankmanager bestimmt. Wurde die Dispatcher-Thread-Parallelität manuell festgelegt, müssen Sie sie möglicherweise auf einen optimalen Wert setzen, der maximale Effizienz sicherstellt und dem Workload-Management-Dispatcher die maximale Steuerung der CPU-Ressourcenzuordnungen ermöglicht.

Nächste Schritte

Führen Sie nach der Festlegung der Dispatcher-Thread-Parallelität die folgenden Tasks aus:

- Sie müssen den Workload-Management-Dispatcher aktivieren, indem Sie den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers konfigurieren, damit die Einstellung der Dispatcher-Thread-Parallelität wirksam wird. Weitere Informationen finden Sie in „Aktivieren des Workload-Manager-Dispatchers“ auf Seite 198.
- Erstellen oder ändern Sie eine Serviceklasse und konfigurieren Sie den CPU-Grenzwert über die Anweisung **CREATE SERVICE CLASS** oder **ALTER SERVICE CLASS**. Weitere Informationen finden Sie in „Festlegen eines CPU-Grenzwerts“ auf Seite 228.

- Aktivieren Sie feste und veränderliche CPU-Anteile, indem Sie den Konfigurationsparameter `wlm_disp_cpu_shares` des Datenbankmanagers konfigurieren. Weitere Informationen finden Sie in Schritt 1 in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Erstellen oder Ändern einer Serviceklasse und Konfigurieren eines festen oder veränderlichen CPU-Anteils über die Anweisung `CREATE SERVICE CLASS` oder `ALTER SERVICE CLASS`. Weitere Informationen finden Sie in „Aktivieren und Festlegen von CPU-Anteilen“ auf Seite 212.
- Legen Sie den minimalen CPU-Auslastungsprozentsatz für den Workload-Management-Dispatcher zur Aktivierung von Serviceklassen fest, indem Sie den Konfigurationsparameter `wlm_disp_min_util` des Datenbankmanagers festlegen. Weitere Informationen finden Sie in „Festlegen der minimalen CPU-Ressourcenauslastung zum Klassifizieren der Serviceklasse als aktiv“ auf Seite 232.

Überwachen Sie Ihr System nach der Aktivierung des Workload-Management-Dispatchers und der Festlegung der Dispatcher-Thread-Parallelität auf eine Verringerung des Durchsatzes und der CPU-Auslastung. Sinkt einer dieser beiden Messwerte, erhöhen Sie den Wert des Konfigurationsparameters `wlm_disp_concur` in Inkrementen der Anzahl verfügbarer Prozessoren. Weitere Informationen zum Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers finden Sie in „Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers“

Überwachen und Optimieren der Leistung des Workload-Management-Dispatchers

Die Überwachung und Optimierung der Leistung des Workload-Management-Dispatchers kann mit den im DB2-Datenbankmanager bereitgestellten Tabellenfunktionen und Überwachungselementen erzielt werden. In diesem Abschnitt werden entsprechende Details geliefert.

Einführung

Sie brauchen die geeigneten Tools, um die Leistung des Workload-Management-Dispatchers zu überwachen und zu optimieren und um die besten Ergebnisse zu erzielen. Mithilfe der bereitgestellten Tabellenfunktionen und Überwachungselemente können Sie die Leistung des Dispatchers überwachen. Nach dem hier beschriebenen Analysieren der erfassten Überwachungsdaten können Sie die Dispatcher-Thread-Parallelität anpassen oder die CPU-Nutzungsrechte erneut verteilen, indem Sie die CPU-Anteile der Serviceklassen und die CPU-Grenzwerte zum Optimieren der Dispatcherleistung anpassen.

In den folgenden Abschnitten werden die Typen von Workloads, die sich darin unterscheiden, wie sie am besten überwacht werden, um Ihnen die geeigneten zu analysierenden Daten zu liefern, und die Leistungsmaßnahmen beschrieben, die sich für den bestimmten Typ von Workload am besten eignen.

Typen von Workloads

Es gibt zwei Typen von Workloads für das Messen der Leistung, um Ihre Dispatcherkonfiguration so zu optimieren, dass Sie die bestmögliche Workloadleistung aus Ihrem System abrufen: Stapelworkloads und transaktionsorientierte Workloads. Jeder Workloadtyp verfügt über Merkmalkennzahlen der Leistung, mit denen am besten ermittelt werden kann, wie gut Ihr System unter diesem Workloadtyp ausgeführt wird. Verwenden Sie die Leistungskennzahlen, die den Workloadtyp Ihres Systems am besten charakterisieren.

Stapelworkload

Eine Stapelworkload verfügt über mindestens eine Anwendung, die die Verbindung zur Datenbank herstellt, und jede Anwendung übergibt ununterbrochen nacheinander Aktivitäten bzw. Transaktionen. Die wichtigste Kennzahl der Leistung dieser Workload bezieht sich darauf, wie schnell die gesamte Gruppe von Aktivitäten oder Transaktionen abgeschlossen wird. Die Verarbeitungsgeschwindigkeit des Datenbankmanagers bestimmt hauptsächlich, wie schnell die gesamte Gruppe von Aktivitäten oder Transaktionen abgeschlossen wird.

Transaktionsorientierte Workload

Eine transaktionsorientierte Workload verfügt über einen Benutzer an einem Terminal, der eine Aktivität oder Transaktion an die Datenbank übergibt, dann auf eine Antwort wartet, die Antwort analysiert und entscheidet, ob eine Folgeaktivität oder -transaktion übergeben werden soll. Bei diesem Workloadtyp ist die wichtigste Kennzahl der Leistung die Frage, wie schnell der Benutzer ein einzelnes Ergebnis erhält. Die Geschwindigkeit des Datenbankmanagers beim Verarbeiten einer einzelnen Aktivität oder Transaktion für jeden einzelnen Benutzer auf dem System bestimmt hauptsächlich, wie schnell der durchschnittliche Benutzer ein einzelnes Ergebnis erhalten kann. Wie schnell der Datenbankmanager alle Aktivitäten oder Transaktionen von einem Benutzer über einen angegebenen Zeitraum verarbeiten kann, ist nicht der relevante Messwert, weil er mehr vom Benutzerverhalten als von der Leistung des Datenbankmanagers abhängt.

Leistungskennzahlen

Mit den folgenden Leistungskennzahlen können Sie ermitteln, wie gut Ihr System unter einem bestimmten Workloadtyp ausgeführt wird.

Durchschnittlicher Durchsatz

Der durchschnittliche Durchsatz ist die durchschnittliche Anzahl Servicebeendigungen pro Einheitenzeit. Ist der Service eine Transaktion oder UOW, ist der durchschnittliche UOW-Durchsatz die Anzahl UOW-Beendigungen pro Einheitenzeit. Er wird gewöhnlich als Transaktionen pro Sekunde oder Minute dargestellt. Der durchschnittliche Durchsatz ist eine nützliche Kennzahl der Systemleistung, wenn der Typ der gemessenen Arbeit eine Stapelworkload ist.

Der durchschnittliche Aktivitätsdurchsatz ist die durchschnittliche Anzahl Aktivitätsbeendigungen pro Einheitenzeit. Auf einem System mit UOWs mit überwiegend langer Laufzeit, die viele einzelne Aktivitäten enthalten, ist es einfacher, den Fortschritt der Aktivitäten in der Workload über den Aktivitätsdurchsatz zu messen, anstatt den Fortschritt der Workload über den UOW-Durchsatz zu messen.

Durchschnittliche Antwortzeit

Die durchschnittliche Antwortzeit ist die durchschnittliche Zeit zwischen der Serviceanforderung und dem Abschluss eines bestimmten Service. Wenn der Service eine Transaktion oder UOW ist, ist die durchschnittliche UOW-Antwortzeit die Zeit, die eine UOW für ihren Abschluss seit ihrer Anforderung benötigt. Die durchschnittliche Antwortzeit ist eine nützliche Kennzahl der Systemleistung, wenn der Typ der gemessenen Arbeit eine transaktionsorientierte Workload ist. Die beste Approximation der durchschnittlichen UOW-Antwortzeit ist die Statistik `uow_lifetime_avg`, die über die Tabellenfunktionen `MON_SAMPLE_SERVICE_CLASS_METRICS` und `MON_SAMPLE_WORKLOAD_METRICS`, die Tabellenfunktionen `WLM-`

`_GET_SERVICE_SUBCLASS_STATS` und `WLM_GET_WORKLOAD_STATS` sowie die logischen Datengruppen 'event_scstats' und 'event_wlstats' des Ereignismonitors verfügbar sind, die im WLM-Statistikeremonitor aufgelistet sind. Eine ausgereifere Form der UOW-Lebensdauerinformationen finden Sie im UowLifetime-Histogramm, das auch im Ereignismonitor verfügbar ist.

Die durchschnittliche Aktivitätsantwortzeit ist die durchschnittliche Zeit, die eine einzelne Aktivität seit dem Start der Aktivität benötigt, um ihr Ergebnis zurückzugeben. Die beste Approximation der durchschnittlichen Aktivitätsantwortzeit ist die Statistik `coord_act_lifetime_avg`, die über die Tabellenfunktionen `WLM_GET_SERVICE_SUBCLASS_STATS` und `WLM_GET_WORKLOAD_STATS` sowie die logischen Datengruppen 'event_scstats' und 'event_wlstats' des Ereignismonitors verfügbar sind. Diese Zahl wird für jedes Member gemessen und wird zurückgesetzt, wenn ein Member inaktiviert wird oder wenn die Prozedur `WLM_COLLECT_STATS` aufgerufen wird. Hierbei kann es sich um eine Approximation handeln, weil eine Cursoraktivität vor ihrem Abschluss einige Ergebnisse zurückgeben kann und es die Aufgabe des Benutzer ist, die Ergebnismenge zu lesen und den Cursor zu schließen, bevor die Aktivität als abgeschlossen gilt. Eine ausgereifere Form der Aktivitätslebensdauer finden Sie im `CoordActLifetime`-Histogramm, das auch im Ereignismonitor verfügbar ist.

CPU-Auslastung

Die CPU-Auslastung ist ein weiterer Messwert, der für die Optimierung des Workload-Management-Dispatchers nützlich und vom Typ der Workload unabhängig ist. Die CPU-Auslastung ist der Bruchteil der Zeit, die die CPU-Ressourcen auf dem Host oder der logischen Partition ausgelastet sind. Die CPU-Auslastung ist der Messwert, über den der Workload-Management-Dispatcher die CPU-Ressourcen einer Serviceklasse zuordnet. Außerdem ist die CPU-Auslastung der Messwert, mit dem Sie prüfen können, ob die Konfiguration Ihres Workload-Management-Dispatchers wie gewünscht funktioniert. Sie können die CPU-Auslastung über dieselben Intervalle wie die Monitorelemente `uow_throughput`, `uow_lifetime_avg` und `act_throughput` messen, indem Sie die Tabellenfunktionen `MON_SAMPLE_SERVICE_CLASS_METRICS` und `MON_SAMPLE_WORKLOAD_METRICS`, die Tabellenfunktionen `WLM_GET_SERVICE_SUBCLASS_STATS` und `WLM_GET_WORKLOAD_STATS` sowie die logischen Datengruppen 'event_scstats' und 'event_wlstats' des Ereignismonitors verwenden, die durch den WLM-Statistikeremonitor erfasst und aufgelistet werden.

Anmerkung: Wenn die CPU-Auslastungsmesswerte für Ihre erstellten Serviceklassen nicht wie erwartet ausfallen, prüfen Sie, ob Workloads vorhanden sind, die unter den Standardbenutzer- und Wartungsserviceklassen ausgeführt werden, weil diese Workloads den von Ihnen erstellten Serviceklassen nicht explizit zugewiesen wurden. Wenn Sie vergessen, Workloads einzuschließen, die unter diesen Standard-serviceklassen ausgeführt werden, wobei jeder Serviceklasse bei der ersten Aktivierung von CPU-Anteilen standardmäßig 1.000 feste CPU-Anteile zugewiesen wurden, kann dies zu nicht erwarteten CPU-Auslastungsmesswerten führen.

Die durch die Tabellenfunktionen und Ereignismonitore aufgelistete CPU-Auslastung stellt die CPU-Ressourcen dar, die von der Arbeit verbraucht werden, die nur in den Benutzer- und Wartungsserviceklassen ausgeführt wird. Die Arbeit, die nicht vom Dispatcher abgewickelt wird, wird bei der CPU-Auslastung nicht berücksichtigt.

Folgende Arbeit wird vom Workload-Management-Dispatcher nicht abgewickelt:

- Arbeit, die von Anwendungen oder Middlewareprodukten mit Ausnahme des DB2-Datenbankmanagers ausgeführt wird, die einen Teil ihrer Arbeit außerhalb des DB2-Datenbankmanagers ausführen
- Arbeit, die von Entitäten ausgeführt wird, die in der DB2-Systemserviceklasse ausgeführt werden
- Von anderen DB2-Instanzen ausgeführte Arbeit
- Nicht-DB2-Datenbankmanagerarbeit, die in Prozessen im abgeschirmten Modus wie abgeschirmte gespeicherte Prozeduren ausgeführt wird
- In vertrauenswürdigen Routinen ausgeführte Nicht-DB2-Datenbankmanagerarbeit

Sie müssen die CPU-Auslastung für diese anderen Konsumenten von CPU-Ressourcen über die Überwachung auf der Betriebssystemebene (wie die von Betriebssystem-Workload-Managern bereitgestellte Überwachung) abrufen.

CPU-Geschwindigkeit

Die CPU-Geschwindigkeit ist eine Statistik, die ermittelt, ob für eine Ressource eine Konkurrenzsituation auftritt und, wenn dies zutrifft, wie hoch der Grad dieser Konkurrenzsituation ist. Wenn der gesamte Zugriff auf eine Ressource gegenseitig ausschließend ist und wenn mehrere anfordernder Benutzer dieser Ressource gleichzeitig auf sie zugreifen wollen, muss eine Zugriffswarteschlange vorhanden sein oder anfordernde Benutzer müssen abgewiesen werden. Wenn eine Warteschlange gebildet werden kann, kann die Zeit, die ein Requester für das Abrufen und Beenden einer Ressource benötigt, die Zeit überschreiten, die für die reine Verwendung der Ressource erforderlich ist. Die Geschwindigkeit ist das Verhältnis der Zeit, die für die reine Verwendung der Ressource benötigt wird, zur Gesamtzeit, die für das Warten auf die Ressource und ihre Verwendung erforderlich ist. Sie wird auf einer Skala von 0 bis 100 % gemessen. Wenn für eine Ressource eine hohe Anzahl Konkurrenzsituationen vorhanden sind, sinkt die Geschwindigkeit in Richtung 0. Wenn für eine Ressource keine Konkurrenzsituation vorhanden ist, gibt es keine Wartezeit, d. h., die CPU-Geschwindigkeit erreicht ihren Maximalwert von 100 %.

Wenn der Workload-Management-Dispatcher aktiviert ist, können Sie die CPU-Geschwindigkeit über die Tabellenfunktionen `MON_SAMPLE_SERVICE_CLASS_METRICS`, `MON_SAMPLE_WORKLOAD_METRICS`, `WLM_GET_WORKLOAD_STATS` und `WLM_GET_SERVICE_SUBCLASS_STATS` sowie die logischen Datengruppen 'event_scstats' und 'event_wlstats' messen, die vom WLM-Statistikeignismonitor erfasst und aufgelistet werden. Ein niedriger CPU-Geschwindigkeitswert weist auf eine Konkurrenzsituation bei CPU-Ressourcen des Hosts oder der logischen Partition hin und ist ein Hinweis darauf, dass der Workload-Management-Dispatcher CPU-Ressourcen aus Serviceklassen mit niedriger Priorität in Serviceklassen mit hoher Priorität verschieben kann. Eine hohe CPU-Geschwindigkeit gibt an, dass der Workload-Management-Dispatcher die Workloadleistung nur begrenzt verbessern kann, da jede Anforderung von CPU-Ressourcen bereits ohne Verzögerung verarbeitet wird.

Abbrechen von Aktivitäten

Wenn eine Aktivität zu viele Ressourcen verbraucht oder ihre Ausführung zu lange dauert, können Sie sie abbrechen. Das Abbrechen einer Aktivität ist schonender als das erzwungene Beenden der Anwendung, die die Aktivität übergeben hat. Eine abgebrochene Aktivität gibt SQL4725N an den Benutzer zurück, beendet jedoch nicht die Verbindung und wirkt sich nicht auf andere Benutzeraktivitäten aus. Durch das erzwungene Beenden der Anwendung werden sowohl die Verbindung als auch die Benutzeraktivitäten beendet.

Informationen zu diesem Vorgang

Sie können eine Aktivität nur dann explizit abbrechen, wenn eine Koordinatoraktivität zurzeit an einer Anforderung für die Aktivität arbeitet. Wenn Sie eine Aktivität abbrechen, die sich im Leerlauf befindet (d. h., für die keine Anforderungen verarbeitet werden), erhält die Aktivität den Status CANCEL_PENDING und wird beim Empfang der nächsten Anforderung abgebrochen. Wenn Sie z. B. versuchen, eine Cursoraktivität zwischen Abrufen abzubrechen, wird der Fehler SQL4725N erst beim nächsten Abruf nach der Abbruchanforderung an den Benutzer zurückgegeben.

Alle Benutzeraktivitäten können abgebrochen werden, auch das Dienstprogramm LOAD und gespeicherte Prozeduren.

Vorgehensweise

1. Geben Sie die Aktivität an, die Sie abbrechen wollen. Mit der Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES können Sie die Aktivitäten ermitteln, die in einer Anwendung ausgeführt werden. Sie können auch die Tabellenfunktion MON_GET_ACTIVITY_DETAILS_COMPLETE verwenden, um zusätzliche Details zu einer bestimmten Aktivität zu erhalten, wenn die Informationen in WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES nicht ausreichen, um die von den Aktivitäten ausgeführte Arbeit zu ermitteln.
2. Brechen Sie die Aktivität mithilfe der gespeicherten Prozedur WLM_CANCEL_ACTIVITY ab. Die gespeicherte Prozedur verwendet die folgenden Argumente: *anwendungskennung*, *uow-id* und *aktivitäts-id*. Ein Beispiel für die Verwendung dieser gespeicherten Prozedur finden Sie in „Szenario: Ermitteln von Aktivitäten, deren Ausführung zu lange dauert“ auf Seite 376.

Kapitel 4. Überwachung und Eingriff

Die dritte Phase beim Workload-Management ist die Überwachung, die fortlaufend durchgeführt werden muss.

Der Hauptzweck der Überwachung ist die Überprüfung auf fehlerfreien Zustand und Effizienz Ihres Systems und der einzelnen Workloads, die darauf ausgeführt werden. Mit Tabellenfunktionen können Sie in Echtzeit auf Betriebsdaten wie z. B. eine Liste aktiver Workloadvorkommen und der Aktivitäten in einer Serviceklasse oder der durchschnittlichen Antwortzeiten zugreifen. Mit Ereignismonitoren können Sie detaillierte Aktivitätsinformationen und zusammengefasste Aktivitätsstatistikdaten für die Protokollanalyse erfassen.

Beim Aufbau einer Überwachungsstrategie sollten Sie sich immer zuerst die zusammengefassten Informationen ansehen. Zusammenfassungen bieten ein gutes Bild der Gesamtaktivität des Datenservers und sind auch weniger aufwendig, da Sie nicht Informationen zu jeder einzelnen Aktivität erfassen müssen, an der Sie eventuell interessiert sind. Sobald Sie den Umfang Ihrer Überwachungsbedürfnisse erkannt haben, können Sie genauere Informationen erfassen.

Typische Überwachungstasks, die Sie durchführen können:

- Analysieren der Auslastung auf Ihrem System als Hilfe für die Erstellung Ihrer DB2-Workload-Management-Anfangskonfiguration.
- Verfolgen und Untersuchen des Verhaltens Ihres Systems, indem Sie Typen von Betriebsinformationen beziehen, die Ihnen Folgendes erlauben:
 - Analysieren der Verschlechterung der Systemleistung
 - Diagnostizieren von Aktivitäten, deren Ausführung zu lange dauert
 - Untersuchen von Konkurrenzsituationen bei Agenten
 - Eingrenzen von Abfragen mit schlechter Leistung

Informationen sind für Aktivitäten, Serviceklassen, Workloads, Arbeitsklassen, Schwellenwertwarteschlangen und Schwellenwertverstöße verfügbar.

- Steuern der Ausführungsumgebung durch Abbrechen von Aktivitäten in der Warteschlange, bei denen Sie Probleme erwarten, oder durch Abbrechen von ausgeführten Aktivitäten, bei denen Sie eine negative Auswirkung auf das System diagnostiziert haben.

Echtzeitüberwachung mit Tabellenfunktionen

Echtzeitüberwachungsdaten umfassen Informationen zu der zurzeit auf dem System ausgeführten Arbeit sowie Statistikdaten und Messdaten für die auf dem System bereits ausgeführte Arbeit, die bei der Ermittlung von Verwendungsmustern, der Ressourcenzuordnung und der Erkennung von Problembereichen helfen können. Sie verwenden DB2-Tabellenfunktionen, um diese Betriebsdaten zu erhalten.

Tabellenfunktionen, deren Namen mit *WLM_* beginnen, sind DB2-Workload-Management-Tabellenfunktionen. Diese Tabellenfunktionen ermöglichen den Zugriff auf eine Gruppe von für das Workload-Management relevanten Daten, z. B. Workload-Management-Statistikdaten, als eine virtuelle DB2-Tabelle, auf der Sie eine Anweisung *SELECT* absetzen können. Dies erlaubt es Ihnen, Anwendungen für die Abfrage und Analyse von Daten zu schreiben, als befänden sich diese in einer

physischen Tabelle auf dem Datenserver. Die DB2-Workload-Management-Tabellenfunktionen sind mit dem Schemanamen SYSPROC qualifiziert.

Tabellenfunktionen, deren Namen mit *MON_* beginnen, sind Funktionen für Überwachungsmessdaten. Überwachungsmessdaten stellen Überwachungsdaten zum Zustand und zur Abfrageleistung Ihres DB2-Datenservers bereit, die dann als Eingabe in ein Tool eines anderen Herstellers oder in Verbindung mit zusätzlichem Scripting verwendet werden können, das Sie für die Analyse der zurückgegebenen Messdaten bereitstellen. Hier werden nur die für DB2-Workload-Management relevanten Funktionen für Überwachungsmessdaten behandelt. Die Tabellenfunktionen für Überwachungsmessdaten sind den Tabellenfunktionen für Workload-Management-Statistikdaten ähnlich. Beide geben Elemente zurück, die die auf dem System ausgeführte Arbeit beschreiben. Die wichtigsten Unterschiede zwischen diesen Tabellenfunktionen für Überwachungsmessdaten und den DB2-Workload-Management-Tabellenfunktionen sind:

- Die DB2-Workload-Management-Tabellenfunktionen stellen Daten zur Verfügung, die eher statistischer Natur sind, z. B. berechnete Werte wie Durchschnittswerte, obere Grenzen, Standardabweichungen usw. Dagegen stellen Tabellenfunktionen für Überwachungsmessdaten eine vollständigere Gruppe von Überwachungsrohdaten bereit.
- Die von den DB2-Statistikfunktionen zurückgegebenen Daten werden zurückgesetzt, wenn Daten an einen Statistikereignismonitor gesendet werden. Dieses Zurücksetzen von Daten ist erforderlich, um Werte wie obere Grenzen in einem bestimmten Erfassungsintervall aussagekräftig zu machen. Von den Funktionen für Überwachungsmessdaten zurückgegebene Daten werden ebenfalls von einem Statistikereignismonitor erfasst, sie werden jedoch nie zurückgesetzt. Die von Überwachungsschnittstellen zurückgegebenen Daten werden von dem Zeitpunkt der Aktivierung einer Datenbank bis zum Zeitpunkt ihrer Inaktivierung gesammelt.

Einige Tabellenfunktionen geben Informationen zu der zurzeit auf einem System ausgeführten Arbeit zurück:

Tabelle 49. Tabellenfunktionen, die die zurzeit auf einem System ausgeführte Arbeit zeigen

Objekte, für die Informationen erfasst werden	Funktionen und zurückgegebene Informationen
Workload-vorkommen	<p>Die Tabellenfunktion <code>WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES</code> gibt datenbankmemberübergreifend eine Liste von Workloadvorkommen zurück, die einer Serviceklasse zugeordnet sind. Für jedes Vorkommen gibt es Informationen zum aktuellen Status und zu den Verbindungsattributen, die für die Zuordnung der Workload zur Serviceklasse verwendet wurden, sowie eine Aktivitätsstatistik, die den Umfang und die Erfolgsquote von Aktivitäten angibt. Ein Beispiel für die Verwendung dieser Tabellenfunktion finden Sie in „Beispiel: Untersuchen der Agentenauslastung nach Serviceklasse“ auf Seite 98.</p> <p>Die veraltete Tabellenfunktion <code>WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES_V97</code> ist ebenfalls verfügbar.</p>

Tabelle 49. Tabellenfunktionen, die die zurzeit auf einem System ausgeführte Arbeit zeigen (Forts.)

Objekte, für die Informationen erfasst werden	Funktionen und zurückgegebene Informationen
Aktivitäten von Workloadvorkommen	<p>Die Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES gibt eine Liste aktueller Aktivitäten zurück, die einem Workloadvorkommen zugeordnet sind. Für jede Aktivität gibt es Informationen zum aktuellen Status der Aktivität (z. B. in Ausführung oder in der Warteschlange), zum Typ der Aktivität (z. B. LOAD, READ oder DDL) und zum Zeitpunkt, zu dem die Aktivität gestartet wurde. Beispiele für die Verwendung dieser Tabellenfunktion finden Sie in „Beispiel: Zusammenfassen von Daten mithilfe von DB2-Workload-Management-Tabellenfunktionen“ auf Seite 252 and „Szenario: Ermitteln von Aktivitäten, deren Ausführung zu lange dauert“ auf Seite 376.</p> <p>Die veraltete Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES_V97 ist ebenfalls verfügbar.</p>
Serviceklassenagenten	<p>Die Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS gibt eine Liste von Datenbankagenten zurück, die einer Serviceklasse oder einer Anwendungskennung zugeordnet sind. Die zurückgegebenen Informationen zeigen auch den aktuellen Status des Agenten, die vom Agenten durchgeführte Aktion und den Status dieser Aktion an. Ein Beispiel für die Verwendung dieser Tabellenfunktion finden Sie in „Beispiel: Untersuchen der Agentenauslastung nach Serviceklasse“ auf Seite 98.</p> <p>Die veraltete Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS_V97 ist ebenfalls verfügbar.</p>
Aktivitäten	<p>Die Tabellenfunktion MON_GET_ACTIVITY_DETAILS gibt Messwerte und andere detaillierte Informationen als XML-Dokument zu einer bestimmten Aktivität zurück, die mit ihrer Anwendungskennung, UOW-ID und Aktivitäts-ID angegeben wird. Eine zurückgegebene Information ist der Aktivitätstyp; je nach Typ wird ein Satz zusätzlicher Daten zurückgegeben. Für SQL-Aktivitäten z. B. sind Aufwandsschätzungen und Informationen zum Anweisungstext, den Paketdaten und den zurückgegebenen oder geänderten Zeilen verfügbar. Details zur Isolationsstufe und zu den Prozessorressourcen sind unter anderem ebenfalls verfügbar.</p> <p>Die veraltete Tabellenfunktion WLM_GET_ACTIVITY_DETAILS ist ebenfalls verfügbar. Ein Beispiel für die Verwendung dieser Tabellenfunktion finden Sie in „Beispiel: Verwenden von DB2-Workload-Management-Tabellenfunktionen zum Überwachen des aktuellen Systemverhaltens auf verschiedenen Ebenen“ auf Seite 248.</p>

Einige Tabellenfunktionen geben Überwachungsdaten für alle auf dem System ausgeführten Anforderungen zurück, zusammengefasst nach Serviceunterklassen und Workloadobjekten:

Tabelle 50. Tabellenfunktionen, die nach DB2-Workload-Management-Objekten zusammengefasste Überwachungsdaten zeigen

Objekte, für die Daten zusammengefasst werden	Funktionen und zurückgegebene Informationen
Workloads	<p>Sowohl die Tabellenfunktion MON_GET_WORKLOAD als auch die Tabellenfunktion MON_GET_WORKLOAD_DETAILS geben Messdaten für mindestens eine Workload zurück. Die von dieser Funktion zurückgegebenen Messdaten stellen die Summierung aller Messdaten für alle Workloadvorkommen dar, die dieselbe Workloaddefinition verwenden.</p> <p>Die Tabellenfunktion MON_GET_WORKLOAD gibt die am häufigsten verwendeten Messdaten in einem spaltenbasierten Format zurück und ist eine effiziente Methode zum Abrufen von Basismessdaten.</p> <p>Die Tabellenfunktion MON_GET_WORKLOAD_DETAILS gibt den gesamten Satz verfügbarer Messdaten in einem XML-Dokumentformat zurück. Dies bietet maximale Flexibilität für die Formatierung der Ausgabe. Die XML-basierte Ausgabe kann von einem XML-Parser direkt syntaktisch analysiert werden oder sie kann mithilfe der Funktion XMLTABLE in das relationale Format konvertiert werden.</p>
Serviceunterklassen	<p>Sowohl die Tabellenfunktion MON_GET_SERVICE_SUBCLASS als auch die Tabellenfunktion MON_GET_SERVICE_SUBCLASS_DETAILS geben Messdaten für mindestens eine Serviceunterklasse zurück. Die von den Tabellenfunktionen zurückgegebenen Messdaten stellen die Summierung aller Messdaten für Anforderungen dar, die unter der angegebenen Serviceunterklasse ausgeführt wurden.</p> <p>Die Tabellenfunktion MON_GET_SERVICE_SUBCLASS gibt die am häufigsten verwendeten Messdaten in einem spaltenbasierten Format zurück und ist eine effiziente Methode zum Abrufen von Basismessdaten.</p> <p>Die Tabellenfunktion MON_GET_SERVICE_SUBCLASS_DETAILS gibt den gesamten Satz verfügbarer Messdaten in einem XML-Dokumentformat zurück. Dies bietet maximale Flexibilität für die Formatierung der Ausgabe. Die XML-basierte Ausgabe kann von einem XML-Parser direkt syntaktisch analysiert werden oder sie kann mithilfe der Funktion XMLTABLE in das relationale Format konvertiert werden.</p>
Verbindungen	<p>Die Tabellenfunktion MON_GET_CONNECTION gibt zusammengefasste Daten für Benutzerverbindungen mit dem System zurück.</p>
UOWs (Units of Work)	<p>Die Tabellenfunktion MON_GET_UNIT_OF_WORK gibt zusammengefasste Daten für die aktuelle UOW (Unit of Work - Arbeitseinheit) in einer Benutzerverbindung zurück.</p>

Statistische Informationen

Allgemeine statistische Informationen sind auch für eine Reihe unterschiedlicher Objekte verfügbar. Sie können diese statistischen Informationen für eine Reihe unterschiedlicher Zwecke nutzen, z. B. um zu prüfen, ob Änderungen an der DB2-Workload-Management-Konfiguration die gewünschten Auswirkungen haben. Wenn Sie z. B. eine neue Arbeitsklasse zur Klassifizierung von READ-Aktivitäten erstellen, können Sie überprüfen, ob die READ-Aktivitäten korrekt unter der neuen

Arbeitsklasse klassifiziert werden. Mithilfe von Tabellenfunktionen können Sie auch bestimmte Systemprobleme schnell erkennen. Sie können z. B. mithilfe von Tabellenfunktionen einen akzeptablen Wert für die durchschnittliche Aktivitätslaufzeit ermitteln und erkennen, wenn dieser Wert über dem üblichen Bereich liegt, was auf ein Problem hindeuten kann, das weiter untersucht werden muss.

Die folgende Tabelle listet die Statistikdaten auf, die mithilfe von Tabellenfunktionen bezogen werden können. Alle Tabellenfunktionen für Statistikdaten geben die Statistikdaten zurück, die seit dem letzten Zurücksetzen der Statistikdaten aufgelaufen sind.

Tabelle 51. Tabellenfunktionen, die statistische Information zeigen

Objekte, für die Statistikdaten zurückgegeben werden	Funktionen und zurückgegebene Statistikdaten
Servicesuperklassen	Die Tabellenfunktion WLM_GET_SERVICE_SUPERCLASS_STATS zeigt datenbankmemberübergreifend eine Übersichtsstatistik auf der Ebene der Servicesuperklasse, d. h., die oberen Grenzwerte für gleichzeitig bestehende Verbindungen, die bei der Ermittlung der höchsten Workloadaktivität hilfreich sind.
Serviceunterklassen	Die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS zeigt datenbankmemberübergreifend eine Übersichtsstatistik auf der Ebene der Serviceunterklasse an (alle Aktivitäten, die in Serviceunterklassen ausgeführt werden). Die Statistikdaten umfassen die Anzahl abgeschlossener Aktivitäten und die durchschnittlichen Ausführungszeiten. Diese Informationen sind hilfreich bei der serviceklassen- und datenbankmemberübergreifenden Ermittlung des Gesamtsystemzustands und der Verteilung von Aktivitäten. Beispiele für die Verwendung dieser Tabellenfunktion finden Sie in „Beispiel: Abrufen von Zeitpunktstatistikdaten aus Serviceklassen“ auf Seite 251, „Beispiel: Zusammenfassen von Daten mithilfe von DB2-Workload-Management-Tabellenfunktionen“ auf Seite 252, „Beispiel: Analysieren einer serviceklassenbezogenen Leistungsminderung des Systems“ auf Seite 97 und „Szenario: Untersuchen einer workloadbezogenen Leistungsminderung des Systems“ auf Seite 372. Die veraltete Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS_V97 ist ebenfalls verfügbar.
Workloads	Die Tabellenfunktion WLM_GET_WORKLOAD_STATS zeigt datenbankmemberübergreifend eine Übersichtsstatistik auf der Workloadzebene an. Hierzu gehören obere Grenzen für gleichzeitige Workloadvorkommen und die Anzahl abgeschlossener Aktivitäten. Diese Informationen sind hilfreich beim Überwachen des Gesamtsystemzustands oder beim Abrufen detaillierter Informationen zur Ermittlung von Problembereichen. Ein Beispiel für die Verwendung dieser Tabellenfunktion finden Sie in „Szenario: Untersuchen einer workloadbezogenen Leistungsminderung des Systems“ auf Seite 372. Die veraltete Tabellenfunktion WLM_GET_WORKLOAD_STATS_V97 ist ebenfalls verfügbar.

Tabelle 51. Tabellenfunktionen, die statistische Information zeigen (Forts.)

Objekte, für die Statistikdaten zurückgegeben werden	Funktionen und zurückgegebene Statistikdaten
Arbeitsaktionssets	Die Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS zeigt datenbankmemberübergreifend eine Übersichtsstatistik auf der Ebene des Arbeitsaktionssets, d. h., die Anzahl der Aktivitäten in jeder Arbeitsklasse, auf die die entsprechenden Arbeitsaktionen angewendet wurden. Diese Informationen sind hilfreich beim Erkennen der Effektivität eines Arbeitsaktionssets und beim Ermitteln der Typen von Aktivitäten, die auf dem System ausgeführt werden. Ein Beispiel für die Verwendung dieser Tabellenfunktion finden Sie in „Beispiel: Analysieren von Workloads nach Aktivitätstyp“ auf Seite 66.
Schwellenwertwarteschlangen	Die Tabellenfunktion WLM_GET_QUEUE_STATS zeigt datenbankmemberübergreifend eine Übersichtsstatistik für die Warteschlangen an, die für Schwellenwerte verwendet werden. Zu den Statistikdaten gehören die aktuelle Zahl und die Gesamtzahl der Aktivitäten in der Warteschlange sowie die Gesamtwartezeit in einer Warteschlange. Diese Informationen sind hilfreich bei der Abfrage der aktuellen Aktivitäten in der Warteschlange oder beim Überprüfen, ob Sie einen Schwellenwert korrekt definiert haben. Eine zu hohe Anzahl in der Warteschlange kann darauf hindeuten, dass ein Schwellenwert zu restriktiv ist, eine sehr kleine Anzahl, dass ein Schwellenwert nicht restriktiv genug ist oder nicht benötigt wird.

Statistikdaten sind nur hilfreich, wenn der Zeitraum, in dem sie erfasst werden, aussagekräftig ist. Das Erfassen von Statistikdaten über einen sehr langen Zeitraum sowie über eine beliebige Zeitspanne mit der gespeicherten Prozedur WLM_COLLECT_STATS kann sich als wenig sinnvoll erweisen, wenn es schwierig wird, Änderungen an Trends oder Problembereiche zu erkennen, da zu viele alte Daten vorhanden sind. Deshalb können Sie die Statistikdaten jederzeit zurücksetzen.

Dank der Standardworkload und der Standardbenutzerserviceklassen sind die Überwachungsfunktionen sofort nach der Installation des DB2-Datenservers verfügbar. Diese können Ihnen behilflich sein, Quellen von Aktivitäten zu ermitteln, die Sie zum Erstellen von Workloads und der Serviceklassen verwenden können, denen sie zugeordnet werden sollen.

Beispiel: Verwenden von DB2-Workload-Management-Tabellenfunktionen

Über die Echtzeitüberwachung des DB2-Workload-Managements stehen große Datenmengen zur Verfügung. Das Beispiel in diesem Abschnitt zeigt, wie Sie diese Informationen möglicherweise verwenden können.

In diesem Beispiel sind nur die Standardworkload und die Standardserviceklasse eingerichtet. Anhand dieses Beispiels können Sie verstehen, wie Sie die Tabellenfunktionen verwenden können, um zu erkennen, was auf dem Datenserver ausgeführt wird. Führen Sie die folgenden Schritte aus:

1. Verwenden Sie die Tabellenfunktion WLM_GET_SERVICE_SUPERCLASS_STATS, um alle Servicesuperklassen anzuzeigen. Nach der Installation oder einem Upgrade auf DB2 Version 9.5 oder höher sind drei Standardsuperklassen

definiert: eine für Verwaltungsaktivitäten, eine für Systemaktivitäten und eine für Benutzeraktivitäten. SYSDEFAULTUSERCLASS ist die uns interessierende Serviceklasse.

```
SELECT VARCHAR(SERVICE_SUPERCLASS_NAME,30) AS SUPERCLASS
       FROM TABLE(WLM_GET_SERVICE_SUPERCLASS_STATS('','-1)) AS T
```

```
SUPERCLASS
-----
SYSDEFAULTSYSTEMCLASS
SYSDEFAULTMAINTENANCECLASS
SYSDEFAULTUSERCLASS
```

3 record(s) selected.

- Verwenden Sie die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS, um statistische Daten für alle Serviceunterklassen der Superklasse SYSDEFAULTUSERCLASS anzuzeigen. Für jede Serviceunterklasse können Sie den aktuellen Umfang der verarbeiteten Anforderungen, die Anzahl der abgeschlossenen Aktivitäten und die Gesamtverteilung von Aktivitäten auf die Member anzeigen (wobei eine ungleichmäßige Verteilung möglicherweise auf ein Problem hindeutet). Sie können optional zusätzliche Statistikdaten erhalten, darunter die durchschnittliche Laufzeit für Aktivitäten, die durchschnittliche Wartezeit von Aktivitäten in der Warteschlange usw. Sie können optionale Statistikdaten für eine Serviceunterklasse ermitteln, indem Sie das Schlüsselwort COLLECT AGGREGATE ACTIVITY DATA in der Anweisung ALTER SERVICE CLASS angeben, um die Erfassung von zusammengefassten Aktivitätsstatistikdaten zu ermöglichen.

```
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 record(s) selected.

- Verwenden Sie für eine bestimmte Serviceunterklasse die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES, um die Vorkommen einer Workload aufzulisten, die der Serviceunterklasse zugeordnet sind. Die Tabellenfunktion zeigt alle Verbindungsattribute an, anhand derer Sie die Quelle der Aktivitäten ermitteln können. Diese Informationen können künftig sehr hilfreich bei der Ermittlung von angepassten Workloaddefinitionen sein. Beispielsweise hat ein bestimmtes hier aufgelistetes Workloadvorkommen ein großes Arbeitsvolumen von einer Anwendung, was durch den Zähler für abgeschlossene Aktivitäten angezeigt wird.

```
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 record(s) selected.

- a. Verwenden Sie für diese Anwendung die Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES, um datenbankmemberübergreifend die aktuellen Aktivitäten anzuzeigen, die von der Verbindung dieser Anwendung erstellt wurden. Sie können diese Informationen für mehrere Zwecke verwenden, z. B. um zu ermitteln, welche Aktivitäten Probleme auf dem Datenserver verursachen könnten.

```
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 record(s) selected

- b. Rufen Sie für jede Aktivität genauere Informationen mit der Tabellenfunktion WLM_GET_ACTIVITY_DETAILS ab. Die Daten zeigen möglicherweise an, dass einige SQL-Anweisungen sehr viele Zeilen zurückgeben, dass einige Aktivitäten sehr lange im Leerlauf warten oder dass einige Abfragen mit einem sehr hohen geschätzten Aufwand ausgeführt werden. In diesen und ähnlichen Situationen kann es sinnvoll sein, einige Schwellenwerte zu definieren, um möglicherweise schädigendes Verhalten zu ermitteln und in Zukunft zu verhindern.

```
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 record(s) selected.

Beispiel: Verwenden von DB2-Workload-Management-Tabellenfunktionen zum Überwachen des aktuellen Systemverhaltens auf verschiedenen Ebenen

Das DB2-Workload-Management stellt eine Reihe von Tabellenfunktionen bereit, mit denen Sie Daten zu Ihrer Workload-Management-Konfiguration erhalten können.

Bei der Installation von DB2 Version 9.5 oder höher wird ein Satz von Standardworkloads und -serviceklassen erstellt. Bevor Sie sich entscheiden, wie Sie Ihre eigene DB2-Workload-Management-Lösung implementieren wollen, können Sie mithilfe der Tabellenfunktionen die auf dem System ausgeführte Arbeit im Hinblick auf die Standardworkloadvorkommen, Serviceklassen und Aktivitäten beobachten.

Sie können beginnen, indem Sie die Liste der Workloadvorkommen in einer Serviceklasse abrufen. Verwenden Sie dazu die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES. Im folgenden Beispiel wird eine leere Zeichenfolge für *servicesuperklassenname* und *serviceunterklassenname* und -2 (ein Platzhalterzeichen) für *member* übergeben:

```
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('1', '', -2)) AS SCINFO
ORDER BY SUPERCLASS_NAME, SUBCLASS_NAME, MEMB, APPHNDL, WORKLOAD_NAME, WLO_ID

```

Nehmen wir an, dass das System vier Datenbankmember hat und dass zwei Anwendungen Aktivitäten in der Datenbank ausführen, wenn Sie die Abfrage absetzen. Die Ergebnisse ähneln der folgenden Ausgabe:

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

Die Ergebnisse zeigen, dass beide Workloadvorkommen der Workload SYSDEFAULTUSERWORKLOAD zugeordnet wurden. Außerdem zeigen die Ergebnisse, dass beide Workloadvorkommen der Serviceunterklasse SYSDEFAULTSUBCLASS in der Servicesuperklasse SYSDEFAULTUSERCLASS zugeordnet wurden und dass beide Workloadvorkommen aus demselben Koordinatormember (Member 0) stammen.

Als Nächstes können Sie die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES erneut verwenden, um die Verbindungsattribute der beiden Workloadvorkommen zu ermitteln:

```

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('1', '', 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

Dann können Sie mit der Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES die aktuellen Aktivitäten eines der beiden Workloadvorkommen anzeigen:

```

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

Die Abfrageergebnisse zeigen, dass das Workloadvorkommen 1 zwei Aktivitäten ausführt. Eine Aktivität ist eine gespeicherte Prozedur (angegeben durch den Akti-

vitätstyp CALL) und die andere Aktivität ist eine DML-Aktivität, die einen Lesevorgang ausführt (z. B. eine Anweisung SELECT). Die DML-Aktivität ist im Aufruf der gespeicherten Prozedur verschachtelt. Sie können erkennen, dass die DML-Aktivität verschachtelt ist, da die ID der übergeordneten UOW (Unit of Work - Arbeitseinheit) und die ID der übergeordneten Aktivität der DML-Aktivität der UOW-ID und der Aktivitäts-ID der CALL-Aktivität entsprechen. Sie können auch erkennen, dass die DML-Aktivität in den Datenbankmembern 0, 1, 2 und 3 ausgeführt wird. Die Informationen zur übergeordneten ID sind nur im Koordinator-Member verfügbar.

Sie können weitere Informationen zu einer gerade ausgeführten Aktivität mit der Tabellenfunktion MON_GET_ACTIVITY_DETAILS erhalten. Diese Tabellenfunktion gibt ein XML-Dokument zurück, in dem die Elemente die Aktivität beschreiben. In diesem Beispiel wird die Funktion XMLTABLE verwendet, um eine Ergebnistabelle aus der XML-Ausgabe zurückzugeben.

```

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
65592	1	1	2009-04-07-18.39.42.549197	1	1
65592	0	1	2009-04-07-18.39.42.552763	1	1

PARENT_UOW_ID	PARENT_ACTIVITY_ID	ACTIVITY_TYPE	NESTING_LEVEL	INVOCATION_ID	ROUTINE_ID
-	-	READ_DML	0	0	0
-	-	READ_DML	0	0	0

2 record(s) selected.

Anmerkung: Die Abfrageergebnisse wurden zur besseren Lesbarkeit in zwei Teile geteilt.

Die bisher aufgeführten Tabellenfunktionen stellen eine abstrakte Beschreibung der auf dem System ausgeführten Arbeit bereit. Die von diesen Tabellenfunktionen bereitgestellten Informationen zum Status der Arbeit ist auf einen Aktivitätsstatus wie EXECUTING beschränkt. Wenn Sie präziser untersuchen wollen, was genau zu

einem bestimmten Zeitpunkt in einer Serviceklasse geschieht, können Sie die Tabellenfunktion `WLM_GET_SERVICE_CLASS_AGENTS` ausführen.

Im folgenden Beispiel wird `WLM_GET_SERVICE_CLASS_AGENTS` durch Übergabe von 1 für *anwendungskennung* und -2 (ein Platzhalterzeichen) für *member* aufgerufen:

```
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,1,14) 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

Die Ergebnisse zeigen einen Koordinatoragenten und einen Subagenten im Member 0 sowie einen Subagenten im Member 1, der für eine Aktivität mit der UOW-ID 1 und der Aktivitäts-ID 5 aktiv ist. Die Informationen zum Koordinatoragenten geben an, dass es sich bei der Anforderung um eine Abrufanforderung handelt.

Beispiel: Abrufen von Zeitpunktstatistikdaten aus Serviceklassen

Jede Aktivität wird vor der Ausführung einer Serviceklasse zugeordnet. Sie können das System mit den Tabellenfunktionen für die Serviceklassenstatistik und durch Abfragen aller Serviceklassen in allen Datenbankmitgliedern überwachen, um Zeitpunktstatistikdaten zu erhalten.

Mit der folgenden Anweisung können Sie Statistikdaten für Serviceklassen wie die durchschnittliche Aktivitätslaufzeit erhalten. Wenn eine leere Zeichenfolge für ein Argument für die Tabellenfunktion `WLM_GET_SERVICE_SUBCLASS_STATS` übergeben wird, wird das Ergebnis nicht durch dieses Argument eingeschränkt. Der Wert des letzten Arguments, *member*, ist -2 (ein Platzhalterzeichen), d. h., Daten werden von allen Datenbankmitgliedern zurückgegeben.

Anmerkung: Informationen zur Laufzeit können nur für die Serviceklassen zurückgegeben werden, für die `COLLECT AGGREGATE ACTIVITY DATA` definiert ist.

```
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

Sie können die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS auch verwenden, um die obere Grenze für den gemeinsamen Zugriff von Koordinatoraktivitäten zu erhalten, die in der Serviceklasse in jedem Member ausgeführt werden:

```
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	10
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	2	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	3	0

Durch Überprüfung der durchschnittlichen Laufzeit und der Anzahl abgeschlossener Aktivitäten können Sie die Ausgabe der Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS verwenden, um eine zusammengefasste Ansicht der Workload in jedem Member in der Datenbank zu erhalten. Signifikante Abweichungen bei den von einer Tabellenfunktion zurückgegebenen oberen Grenzen und Durchschnittswerten können auf eine Änderung in der Workload auf dem System hindeuten.

Beispiel: Zusammenfassen von Daten mithilfe von DB2-Workload-Management-Tabellenfunktionen

Sie können verschiedene Zusammenfassungen von Tabellendaten in einer DB2-Workload-Management-Konfiguration durchführen, um das System zu überwachen und mögliche Probleme zu erkennen.

Das folgende Beispiel zeigt die Datenerfassung und -zusammenfassung, die Sie zur Feststellung von Problemen durchführen können.

Erkennen von plötzlichen Anstiegen der Anzahl der in einer Workload ausgeführten Abfragen

Nehmen wir an, Sie haben eine Workload namens WL1. Sie können eine Situation erkennen, in der eine große Anzahl von Abfragen in der Workload ausgeführt wird. Zeigen Sie dazu die Gesamtzahl der ausgeführten unverschachtelten Koordinatoraktivitäten für die Workload über das gesamte System an:

```
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

Beispiel: Ermitteln der von einem WLM-Schwellenwert in die Warteschlange eingereihten Aktivitäten sowie der Reihenfolge dieser Aktivitäten

Mithilfe der Tabellenfunktion `WLM_GET_SERVICE_CLASS_AGENTS` von DB2-Workload-Manager (WLM) können Sie anzeigen, welche Anwendungen oder Aktivitäten von einem WLM-Schwellenwert in die Warteschlange gestellt werden, sowie die Reihenfolge der Anwendungen oder Aktivitäten in der Warteschlange.

Jeder von einem WLM-Schwellenwert in die Warteschlange gestellte Agent enthält den Wert `WLM_QUEUE` in der Spalte `EVENT_OBJECT` in der entsprechenden für den Agenten zurückgegebenen Zeile. Darüber hinaus enthält die Spalte `AGENT_STATE_LAST_UPDATE_TIME` den Zeitpunkt, an dem der Status `WLM_QUEUE` für den Agenten aktiviert wurde. Dies ist der Zeitpunkt, an dem der Agent in die Warteschlange gestellt wurde. Mithilfe dieser Informationen kann eine einfache Sicht erstellt werden, in der alle Anwendungen oder Aktivitäten aufgelistet werden, die von einem WLM-Schwellenwert in die Warteschlange gestellt wurden, und der Zeitpunkt ihres Eingangs in die Warteschlange abgerufen werden.

Führen Sie die folgende Anweisung aus, um eine Sicht zu erstellen, die mithilfe der Tabellenfunktion `WLM_GET_SERVICE_CLASS_AGENTS` die Informationen der WLM-Warteschlange für die hier bereitgestellten Beispiele abrufen:

```
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'
```

Anmerkung: Für Schwellenwerte `TOTALSCPARTITIONCONNECTIONS` sind `uow_id` und `activity_id` nicht zutreffend und erscheinen daher als `NULL`.

Diese Sicht kann zur einfachen Beantwortung von Fragen wie folgender verwendet werden:

- Wie viele Anwendungen oder Aktivitäten wurden zurzeit von einem WLM-Schwellenwert in die Warteschlange gestellt?
- Welche Reihenfolge haben Anwendungen oder Aktivitäten in der WLM-Schwellenwertwarteschlange?

Beispiel 1

Führen Sie die folgende Anweisung aus, um die Anzahl der von den einzelnen Warteschlangenschwellenwerten in die Warteschlange gestellten Anwendungen zu ermitteln:

```
SELECT VARCHAR(THRESHOLD_NAME, 30) AS THRESHOLD, COUNT(*)
    AS QUEUED_ENTRIES FROM WLM_QUEUE_INFO GROUP BY THRESHOLD_NAME
```

Die vorherige Anweisung kann beispielsweise folgende Ausgabe erzeugen:

THRESHOLD	QUEUED_ENTRIES
-----	-----
TH1	3

1 record(s) selected.

Beispiel 2

Führen Sie die folgende Anweisung aus, um alle Aktivitäten aufzulisten, die im Schwellenwert TH1 CONCURRENTDBCOORDACTIVITIES in die Warteschlange gestellt werden, und die Reihenfolge der Aktivitäten nach dem Zeitpunkt ihres Eingangs in die Warteschlange anzuzeigen:

```
SELECT QUEUE_ENTRY_TIME, APPLICATION_HANDLE, UOW_ID, ACTIVITY_ID FROM
       WLM_QUEUE_INFO WHERE THRESHOLD_NAME = 'TH1' ORDER BY QUEUE_ENTRY_TIME ASC
```

Die vorherige Anweisung kann beispielsweise folgende Ausgabe erzeugen:

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 record(s) selected.

Langzeitüberwachung mit WLM-Ereignismonitoren

Der DB2-Workload-Management verwendet Ereignismonitore zur Erfassung von Informationen, die in Zukunft oder für die Protokollanalyse nützlich sein könnten.

Es stehen drei Ereignismonitore zur Verfügung. Jeder Ereignismonitor dient einem anderen Zweck:

Aktivitätseignismonitor

Dieser Monitor erfasst Informationen zu einzelnen Aktivitäten in einer Serviceklasse, Workload oder Arbeitsklasse bzw. Aktivitäten, die einen Schwellenwertverstoß verursacht haben. Wie viele Daten für jede Aktivität erfasst werden, kann konfiguriert werden. Dabei sollten Sie den Plattenspeicherplatz und die Speicherdauer für die Monitordaten berücksichtigen. Eine übliche Verwendung von Aktivitätsdaten ist als Eingabe in Tools wie **db2advsi** oder zur Verwendung von Zugriffsplänen (vom Dienstprogramm EXPLAIN), um die Tabellen-, Spalten- und Indexnutzung für eine Gruppe von Abfragen zu ermitteln.

Sie können Informationen zu einer Aktivität erfassen, indem Sie COLLECT ACTIVITY DATA für die Serviceklasse, Workload oder Arbeitsaktion angeben, zu der eine solche Aktivität gehört, oder für einen Schwellenwert, gegen den eine solche Aktivität verstoßen könnte. Die Informationen werden bei Abschluss der Aktivität erfasst, unabhängig davon, ob die Aktivität erfolgreich abgeschlossen wurde.

Beachten Sie, dass alle in der Warteschlange enthaltenen Aktivitätsdatensätze gelöscht werden, wenn beim Inaktivieren der Datenbank ein Aktivitätseignismonitor aktiv ist. Aktivieren Sie den Aktivitätseignismonitor vor dem Inaktivieren der Datenbank explizit, um sicherzustellen, dass Sie alle Datensätze des Aktivitätseignismonitors abrufen und kein Datensatz gelöscht wird. Wenn ein Aktivitätseignismonitor explizit inaktiviert wird, werden alle in der Warteschlange angesammelten Aktivitätsdatensätze verarbeitet, bevor der Ereignismonitor inaktiviert wird.

Ereignismonitor für Schwellenwertverstöße

Dieser Monitor erfasst Informationen, wenn gegen einen Schwellenwert verstoßen wird. Er gibt an, gegen welchen Schwellenwert verstoßen wurde, welche Aktivität den Verstoß verursachte und welche Aktion in diesem Fall durchgeführt wurde.

Wenn Sie COLLECT ACTIVITY DATA für den Schwellenwert angeben und ein Aktivitätsereignismonitor erstellt und aktiv ist, werden auch Informationen zu Aktivitäten erfasst, die gegen den Schwellenwert verstoßen. Diese Informationen werden erfasst, wenn die Aktivität (erfolgreich oder nicht erfolgreich) endet.

Sie können Details zu einem Schwellenwert durch Abfragen der Sicht SYSCAT.THRESHOLDS erhalten.

Statistikereignismonitor

Dieser Monitor dient als wenig aufwendige Alternative zur Erfassung detaillierter Aktivitätsinformationen, indem Zusammenfassungsdaten erfasst werden (z. B. die Anzahl abgeschlossener Aktivitäten und die durchschnittliche Ausführungszeit). Zusammengefasste Daten umfassen Histogramme für eine Reihe von Aktivitätsmesswerten, einschließlich Laufzeit, Wartezeit, Ausführungszeit und geschätzter Aufwand. Mithilfe von Histogrammen können Sie die Verteilung von Werten verstehen, Ausreißer ermitteln und zusätzliche Statistikdaten wie Durchschnittswerte und Standardabweichungen berechnen. Mithilfe von Histogrammen können Sie beispielsweise nachvollziehen, welche Varianz in der Laufzeit die Benutzer erleben. Wenn eine sehr breite Streuung auftritt, spiegelt die durchschnittliche Laufzeit allein nicht die Erfahrungen der Benutzer wider. Eine Beschreibung zum Senden von Statistikdaten an den Ereignismonitor finden Sie in „Erfassen von Workload-Management-Statistikdaten mithilfe eines Ereignismonitors für Statistikdaten“ auf Seite 286.

Die folgende Abbildung zeigt die unterschiedlichen verfügbaren Überwachungsoptionen für den Zugriff auf Workloadinformationen: Tabellenfunktionen für den Zugriff auf Echtzeitstatistikdaten sowie Aktivitätsdetails und Protokolldaten, die als effiziente Zusammenfassungen oder als Details zu einzelnen Aktivitäten erfasst werden.

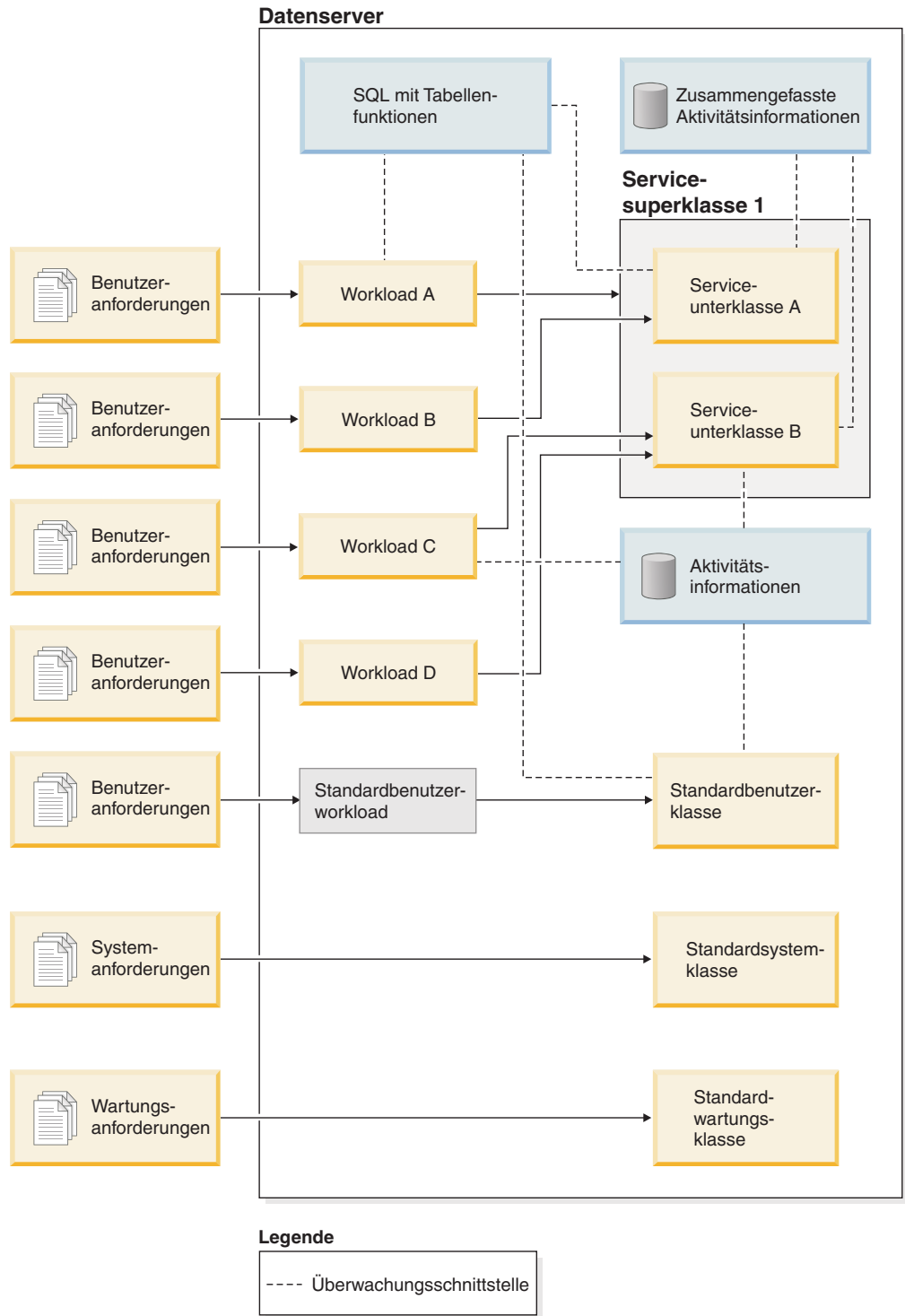


Abbildung 46. Workload-Management mit Überwachung

Im Gegensatz zu Anweisungs-, Verbindungs- und Transaktionsereignismonitoren haben die Aktivitäts- und Statistikereignismonitore und die Ereignismonitore für Schwellenwertverstöße keine Ereignisbedingungen (d. h. Bedingungen, die im Schlüsselwort WHERE der Anweisung CREATE EVENT MONITOR angegeben werden). Diese Ereignismonitore sind stattdessen auf die Attribute von Serviceklassen

sen, Workloads, Arbeitsklassen und Schwellenwerten angewiesen, um zu ermitteln, ob diese Objekte ihre Aktivitätsinformationen oder zusammengefassten Informationen an diese Monitore senden.

Typischerweise schreiben Ereignismonitore Daten entweder in Tabellen oder in Dateien. Sie müssen diese Tabellen oder Dateien regelmäßig bereinigen, da dies nicht automatisch erfolgt.

Mit dem Script `wlmevmon.ddl` im Verzeichnis `sql1ib/misc` können Sie drei Ereignismonitore namens `DB2ACTIVITIES`, `DB2STATISTICS` und `DB2THRESHOLDVIOLATIONS` erstellen und aktivieren. Ändern Sie gegebenenfalls das Script, um den Tabellenbereich oder andere Parameter zu ändern.

Beispiel

Beispiel: Ermitteln von Abfragen mit einem hohen geschätzten Aufwand mithilfe des Statistikereignismonitors: Sie vermuten, dass Ihre Datenbankworkload gelegentlich große aufwendige Abfragen umfasst, möglicherweise aufgrund einer schlechten Optimierung der Anfragen selbst. Sie wollen diese Abfragen identifizieren, sodass Sie verhindern können, dass diese zu viele Ressourcen auf Ihrem System verbrauchen, mit dem Langzeitziel, möglicherweise einige der Abfragen neu zu schreiben, um die Leistung zu verbessern. Der Statistikereignismonitor eine wenig aufwendige Möglichkeit bereit, den geschätzten Aufwand Ihrer Abfragen zu messen, womit Sie dann bestimmen können, wie groß der maximale geschätzte Aufwand für eine Abfrage auf Ihrem Datenserver sein sollte. Eine schlecht optimierte Abfrage lässt sich in der Regel durch einen großen geschätzten Aufwand erkennen, der ein Mehrfaches größer als der geschätzte Aufwand der meisten anderen Abfragen ist.

Zunächst müssen Sie einen Statistikereignismonitor erstellen und aktivieren und dann mit dem Erfassen erweiterter zusammengefasster Aktivitätsdaten für die Serviceklasse beginnen, in der die Abfragen ausgeführt werden:

```
CREATE EVENT MONITOR DB2STATISTICS FOR STATISTICS WRITE TO TABLE
SET EVENT MONITOR DB2STATISTICS STATE 1
```

In diesem Beispiel werden alle Abfragen in der Unterklasse `SYSDEFAULTSUBCLASS` der Serviceklasse `SYSDEFAULTUSERCLASS` ausgeführt, die Sie ändern können, um die erforderlichen Daten zu erfassen:

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA EXTENDED
```

Ein ganzer Arbeitstag kann einen sinnvollen Näherungswert für den Umfang der Abfragen bieten, die Ihr Datenserver typischerweise verarbeitet. Am Ende des Tages können Sie die erfassten Statistikdaten aus dem Speicher in den Statistikereignismonitor kopieren, indem Sie die gespeicherte Prozedur `WLM_COLLECT_STATS` ausführen:

```
CALL WLM_COLLECT_STATS()
```

In den unterschiedlichen Statistiken, die in die Ereignismonitortabellen geschrieben sind, sind die Daten zum geschätzten Aufwand der Abfragen enthalten. Sie können die Serviceklassen-Statistiktafel `SCSTATS_DB2STATISTICS` abfragen, um diese anzuzeigen:

```
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 record(s) selected.

Die Ausgabe zeigt, dass die durchschnittliche Abfrage einen geschätzten Aufwand im Bereich von Hunderttausenden Timeron hat und dass die größten Abfragen einen geschätzten Aufwand von mehr als zehn Millionen Timeron haben. Sie können verifizieren, dass es sich bei Abfragen mit 10 Millionen Timeron oder mehr um Ausreißer handelt, indem Sie sich das Histogramm des geschätzten Aufwands ansehen, das zum gleichen Zeitpunkt generiert wurde, als der in der Ausgabe gezeigte Durchschnitt und die oberen Grenzen in die Ereignismonitortabelle geschrieben wurden. Sie können das Histogramm durch Abfragen der Tabelle HISTOGRAMBIN_DB2STATISTICS wie folgt anzeigen:

```

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

Im Histogramm ist der Wert in der Spalte NUMBER_IN_BIN für Abfragen, deren Maximalwert (TOP) größer als 2616055 ist, gleich null, bis der Maximalwert 14160950 erreicht. Dann ändert sich der Wert in der Spalte NUMBER_IN_BIN in 3. Diese drei Abfragen sind Ausreißer und können mit einem Schwellenwert ESTIMATEDSQLCOST gesteuert werden, der so definiert ist, dass er ausgelöst wird, wenn der geschätzte Aufwand einer Abfrage 10 Millionen Timeron übersteigt. Damit können Sie verhindern, dass solche Aktivitäten ausgeführt werden, und Sie können sie genauer überwachen.

Beispiel: Verwenden des Ereignismonitor für Schwellenwertverstöße: Zur Steuerung von Aktivitäten mit einem bestimmten geschätzten Aufwand können Sie einen Schwellenwert ESTIMATEDSQLCOST in Ihrer Workload definieren, der nur auf diejenige Teilmenge Ihrer Gesamtworkload angewendet wird, die einen bestimmten geschätzten Aufwand überschreitet. Anhand des Histogramms des geschätzten Aufwands haben Sie festgestellt, dass Aktivitäten mit einem geschätzten Aufwand im Bereich von 0 bis unter 3 Millionen Timeron häufig auftreten und dass Aktivitäten mit einem geschätzten Aufwand von über 10 Millionen Timeron sehr selten auftreten (vielleicht nur ein paar Mal pro Tag und möglicherweise immer aufgrund eines Fehlers in der Abfrage, wie z. B. die Verwendung eines kartesischen Join).

Sie können einen Schwellenwertereignismonitor erstellen und aktivieren, um zu überprüfen, ob ein Schwellenwert von 10 Millionen Timeron diese wenigen Aktivitäten pro Tag, die nicht ausgeführt werden sollten, effektiv stoppt:

```
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
```

Am Ende des Tages können Sie sehen, welche Schwellenwertverstöße aufgetreten sind, indem Sie die Tabelle zu Schwellenwertverstößen abfragen:

```
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 record(s) selected.

Beispiel: Verwenden des Aktivitätsereignismonitors

Das vorherige Beispiel zeigte, wie Sie Schwellenwertinformationen in einer Ereignismonitortabelle erfassen können, um sich zu vergewissern, dass die Ausführung von Aktivitäten mit einem hohen geschätzten Aufwand durch einen Schwellenwert verhindert wird. Nachdem Sie diese Schwellenwertverstöße gesehen haben, wollen Sie ermitteln, wie der Text der SQL-Anweisungen lautet, die diese großen Abfragen erzeugen. Sie können also die EXPLAIN-Funktion verwenden, um zu ermitteln, ob ein Index zu den abgefragten Tabellen erforderlich ist.

Für das Erfassen dieser zusätzlichen Informationen müssen Sie einen Aktivitätsereignismonitor erstellen und aktivieren und den Schwellenwert ändern, um die Aktivitätserfassung mit Details zu aktivieren:

```
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
```

Wenn Sie die Tabelle zu Schwellenwertverstößen nach einem weiteren Werktag erneut abfragen, können Sie einen Join mit der Tabelle `ACTIVITYSTMT_DB2ACTIVITIES` durchführen, um den SQL-Anweisungstext jeder Aktivität zu sehen, die gegen den Schwellenwert verstoßen hat:

```
SELECT THRESHOLDID,
       SUBSTR(THRESHOLD_PREDICATE, 1, 20) PREDICATE,
       TIME_OF_VIOLATION,
       SUBSTR(STMTEXT,1,70) STMTEXT
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	STMTEXT
1	EstimatedSQLCost	2008-09-02-23.04.49.000000	select count(*) from syscat.tables,syscat.tables,syscat.tables

1 record(s) selected.

Verfügbare Überwachungsdaten

Überwachungsdaten sind von Workloads, Serviceunterklassen und Servicesuperklassen, Arbeitsklassen und Schwellenwertwarteschlangen verfügbar. Mithilfe dieser Daten können Sie Probleme diagnostizieren und beheben und die Leistung optimieren.

Überwachungsdaten für Workloads

Die folgende Abbildung zeigt die Überwachungsdaten, die für Workloads verfügbar sind. Mit Ereignismonitoren können Sie Workloadstatistikdaten und Informationen zu Aktivitäten erfassen, die in den Workloads ausgeführt werden. Bei Workloads können Sie auch zusammengefasste Aktivitätsstatistikdaten erhalten. Mit Tabellenfunktionen können Sie in Echtzeit auf Workloadstatistikdaten und Informationen zu Workloadvorkommen zugreifen.

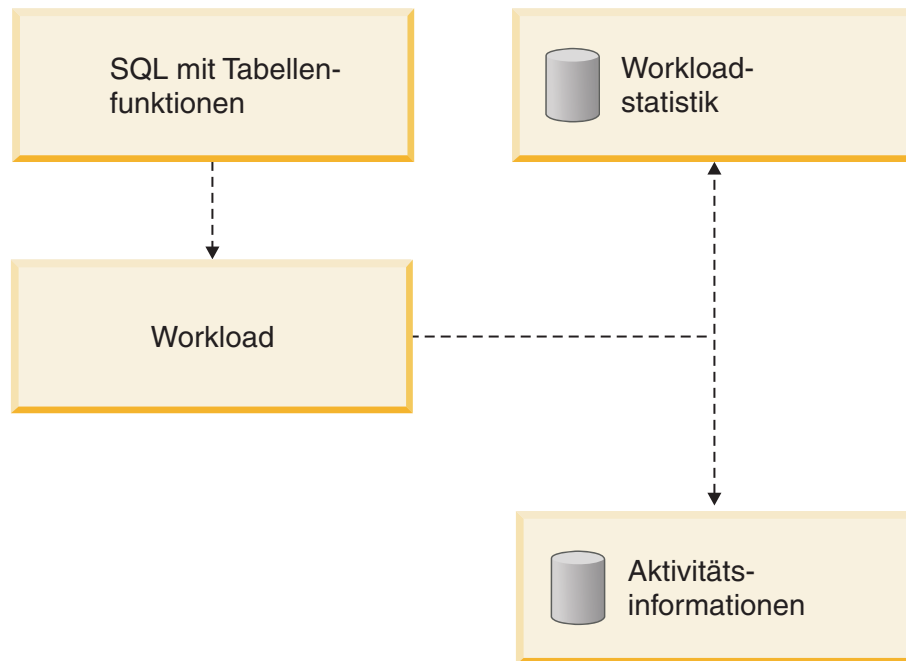


Abbildung 47. Für Workloads verfügbare Überwachungsdaten

Überwachungsdaten für Serviceklassen

Die folgende Abbildung zeigt die Überwachungsdaten, die für Serviceklassen verfügbar sind. Sie können Statistikdaten für Serviceunterklassen und Servicesuperklassen erfassen. Bei Serviceunterklassen können Sie auch zusammengefasste Aktivitäts- und Anforderungsstatistikdaten ermitteln sowie Informationen zu Aktivitäten, die in der Serviceunterklasse ausgeführt werden. Mit Tabellenfunktionen können Sie in Echtzeit auf Statistikdaten zu Servicesuperklassen und Serviceunterklassen sowie auf Informationen zu Agenten in einer bestimmten Serviceklasse zugreifen.

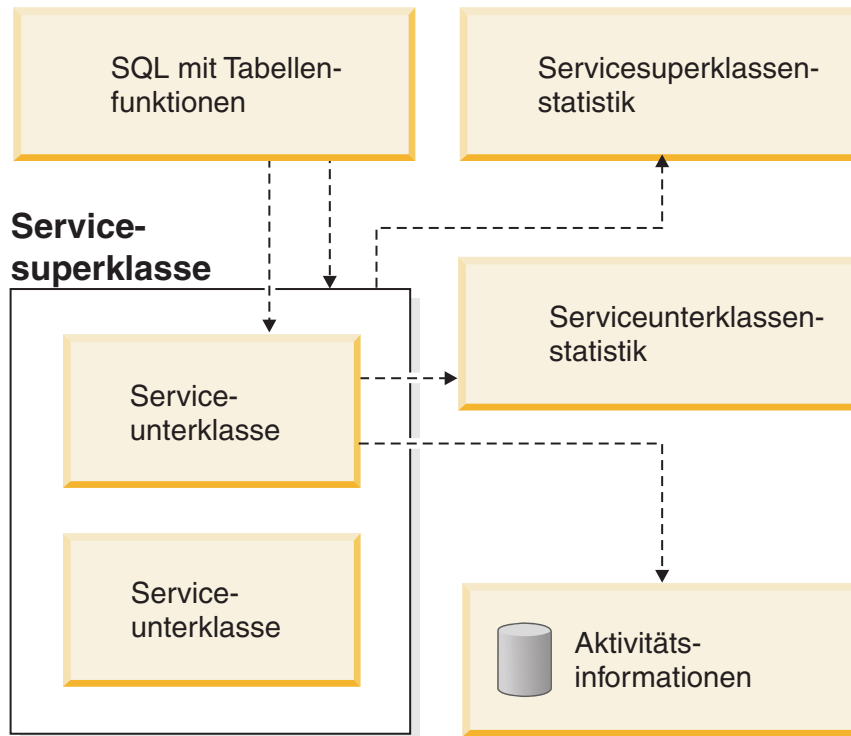


Abbildung 48. Für Serviceklassen verfügbare Überwachungsdaten

Überwachungsdaten für Arbeitsklassen

Die folgende Abbildung zeigt die Überwachungsdaten, die für Arbeitsklassen verfügbar sind. Sie können Statistikdaten zu Arbeitsklassen sowie Informationen zu Aktivitäten erfassen, die einer bestimmten Arbeitsklasse zugeordnet sind. Mit Tabellenfunktionen können Sie in Echtzeit auf Statistikdaten zu Arbeitsklassen zugreifen.

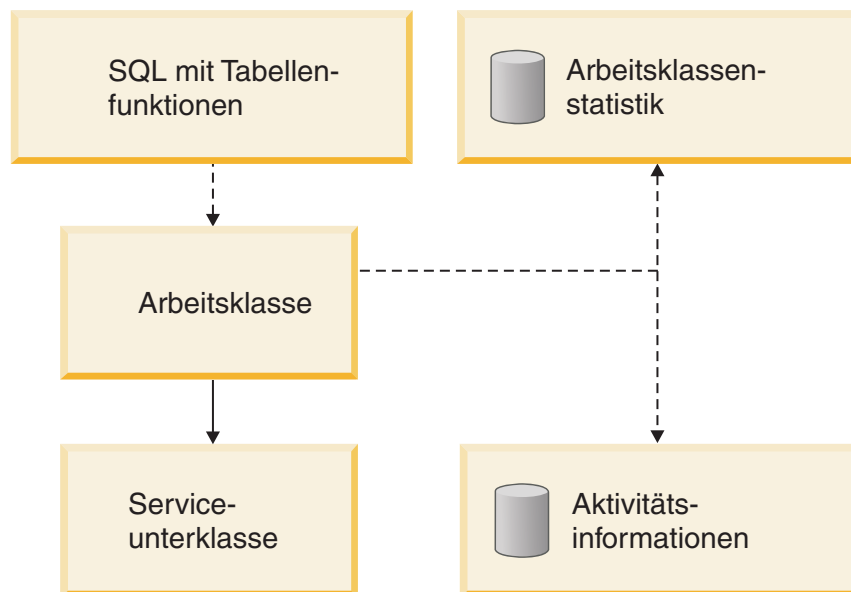


Abbildung 49. Für Arbeitsklassen verfügbare Überwachungsdaten

Überwachungsdaten für Schwellenwerte

Die folgende Abbildung zeigt die Überwachungsdaten, die für Schwellenwerte verfügbar sind. Sie können Informationen zu Schwellenwertverstößen, zu den Aktivitäten, die die Schwellenwertverstöße verursacht haben, sowie Statistikdaten zu Warteschlangen (für Warteschlangenschwellenwerte) erfassen. Mit Tabellenfunktionen können Sie in Echtzeit auf Statistikdaten zu Warteschlangenschwellenwerten zugreifen.

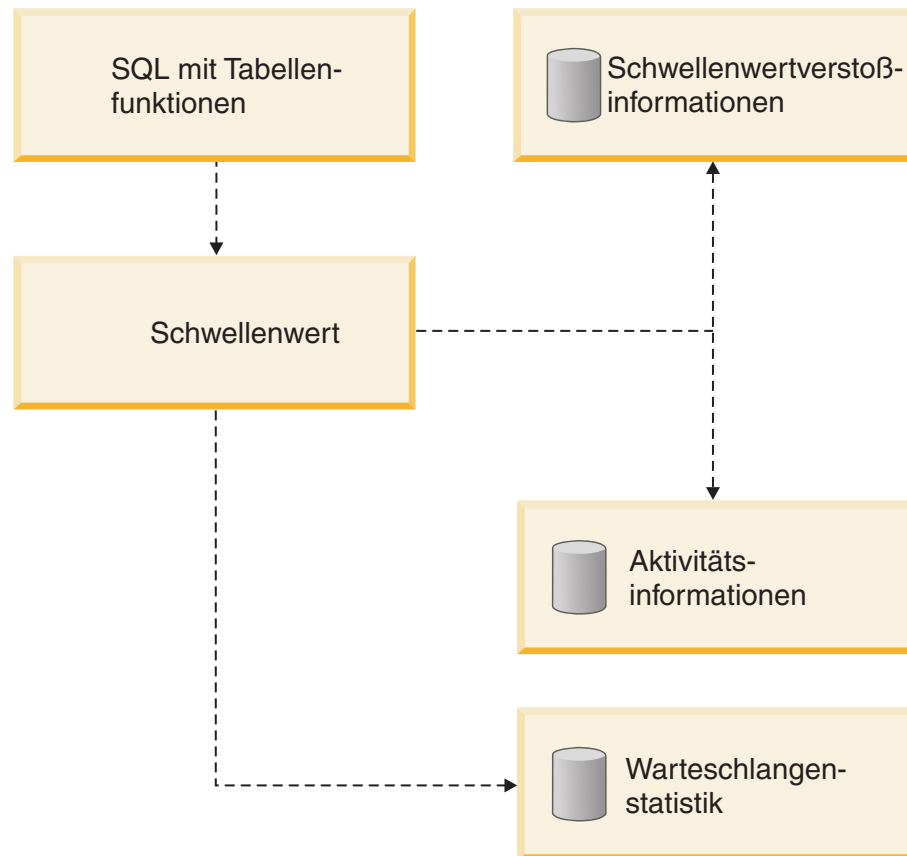


Abbildung 50. Für Schwellenwerte verfügbare Überwachungsdaten

Gespeicherte Prozeduren des DB2-Workload-Managements

Sie können gespeicherte Prozeduren für das Abbrechen einer Aktivität, das Erfassen von Details zu einer Aktivität, das Zurücksetzen der Statistikdaten zu DB2-Workload-Management-Objekten und das Festlegen von Clientinformationen auf dem Datenserver verwenden.

Die folgenden gespeicherten Prozeduren sind für die Verwendung mit dem DB2-Workload-Management verfügbar:

WLM_CANCEL_ACTIVITY(*anwendungskennung, uow-id, aktivitäts-id*)

Mit dieser gespeicherten Prozedur können Sie eine zurzeit ausgeführte oder in der Warteschlange befindliche Aktivität abbrechen. Sie geben die Aktivität durch ihre Anwendungskennung, die UOW-ID und die Aktivitäts-ID an. Sie können jeden Typ von Aktivität abbrechen. Die Anwendung mit der abgebrochenen Aktivität empfängt den Fehler SQL4725N.

WLM_CAPTURE_ACTIVITY_IN_PROGRESS(*anwendungskennung, uow-id, aktivitäts-id*) Mit dieser gespeicherten Prozedur können Sie Informationen zu einer einzelnen Aktivität, die zurzeit ausgeführt wird, an den Aktivitätsereignismonitor senden. Diese gespeicherte Prozedur sendet die Informationen unverzüglich, statt zu warten, bis die Aktivität abgeschlossen ist.

WLM_COLLECT_STATS()

Mit dieser gespeicherten Prozedur erfassen Sie Statistikdaten für DB2-Workload-Management-Objekte und setzen diese zurück. Alle für Serviceklassen, Workloads, Schwellenwertwarteschlangen und Arbeitsaktionssets verfolgten Statistikdaten werden an den aktiven Statistikereignismonitor (sofern vorhanden) gesendet und zurückgesetzt. Wenn kein aktiver Statistikereignismonitor vorhanden ist, werden die Statistikdaten nur zurückgesetzt, jedoch nicht erfasst.

WLM_SET_CLIENT_INFO(*clientbenutzer-id, clientworkstationname, clientanwendungsname, clientabrechnungszeichenfolge, client-workload*)

Mit dieser Prozedur legen Sie die Clientinformationsattribute fest, die auf dem Datenserver verwendet werden, um die Identität der Anwendung oder des Endbenutzers aufzuzeichnen, die/der die Verbindung zurzeit nutzt. In Fällen, in denen zwischen den Anwendungen oder Benutzern und Ihrem Datenserver Middleware vorhanden ist, verwenden Sie die Prozedur **WLM_SET_CLIENT_INFO**, um die unterscheidenden Verbindungsattribute explizit festzulegen.

Statistikdaten für DB2-Workload-Management-Objekte

Für DB2-Workload-Management-Objekte, einschließlich Serviceklassen, Arbeitsklassen, Workloads und Schwellenwertwarteschlangen, werden Statistikdaten erfasst. Diese Statistikdaten befinden sich im Hauptspeicher und können in Echtzeit mit Tabellenfunktionen für DB2-Workload-Management-Statistiken angezeigt werden. Die Statistikdaten können auch erfasst und an einen Statistikereignismonitor gesendet werden, wo sie später für die Protokollanalyse angezeigt werden können.

Über den Statistikereignismonitor können Sie auch Überwachungsmessdaten erhalten. Diese werden jedoch in diesem Abschnitt nicht erörtert. Hier werden nur die für das DB2-Workload-Management spezifischen Statistikdaten beschrieben.

Wenn Statistikdaten an den Ereignismonitor gesendet werden, werden die Werte im Speicher zurückgesetzt, um zu verhindern, dass die Daten in anschließenden Erfassungsintervallen doppelt erfasst werden. Da die Tabellenfunktionen für DB2-Workload-Management-Statistiken die Werte auflisten, die sich zurzeit im Speicher befinden, werden nach einer Erfassung die zurückgesetzten Werte aufgelistet. Die DB2-Workload-Management-Tabellenfunktionen listen nur eine Teilmenge der Statistikdaten auf. Wenn Sie alle Statistikdaten anzeigen wollen, müssen Sie die Statistikdaten erfassen und sie an einen Statistikereignismonitor senden.

Statistikerfassung von zusammengefassten Aktivitätsdaten

Die folgenden Statistikdaten werden für die gegebenen Objekte in jedem Member erfasst, unabhängig vom Wert der Option **COLLECT AGGREGATE ACTIVITY DATA**, die für diese Objekte bei ihrer Erstellung oder Änderung angegeben wird.

Tabelle 52. Für Datenbankobjekte unabhängig von der Einstellung COLLECT AGGREGATE ACTIVITY DATA erfasste Statistikdaten

Datenbankobjekt	Statistikdaten	Beschreibung
Serviceunterklassen	<ol style="list-style-type: none"> 1. Max. gleichzeitig ablaufende Aktivitäten (concurrent_act_top) 2. Abgeschlossene Koordinatoraktivitäten insgesamt (coord_act_completed_total) 3. Abgebrochene Koordinatoraktivitäten insgesamt (coord_act_aborted_total) 4. Zurückgewiesene Koordinatoraktivitäten insgesamt (coord_act_rejected_total) 5. Anzahl aktiver Anforderungen (num_requests_active) 6. Anzahl zugeordneter Aktivitäten und Anzahl Aktivitäten, deren Zuordnung aufgehoben wurde (act_remapped_in und act_remapped_out) 	<ol style="list-style-type: none"> 1. Mithilfe dieser Obergrenze für gleichzeitig ablaufende Aktivitäten können Sie den höchsten gemeinsamen Zugriff von Aktivitäten (einschließlich verschachtelter Aktivitäten) ermitteln, die in einem Member für eine Serviceklasse in dem Zeitintervall erreicht wird, für das die Statistikdaten erfasst werden. 2. Mithilfe dieser Statistik können Sie ermitteln, wieviel Arbeit in einer Serviceklasse ausgeführt wird. 3. Mithilfe dieser Statistik, die den nicht erfolgreichen Abschluss von Aktivitäten misst, können Sie den Zustand des Systems ermitteln. Aktivitäten können aufgrund von Abbruchbefehlen, Fehlern oder reaktiven Schwellenwerten abgebrochen werden. 4. Mithilfe des Zählers für zurückgewiesene nicht verschachtelte Koordinatenaktivitäten, der die Zurückweisung von Aktivitäten misst, können Sie einen Hinweis auf die Zweckmäßigkeit der Zurückweisungsrichtlinie erhalten. Aktivitäten werden als 'Zurückgewiesen' gewertet, wenn sie gegen einen prädiktiven Schwellenwert verstoßen haben, für den die Aktion STOP EXECUTION definiert wurde, oder wenn sie an der Ausführung durch eine Arbeitsaktion gehindert werden. 5. Mithilfe dieser Statistik können Sie die Anzahl der Anforderungen ermitteln, die zurzeit in einer Serviceklasse ausgeführt werden. 6. Mithilfe dieser Statistik können Sie die Anzahl der Aktivitäten, die einer Serviceunterklasse neu zugeordnet wurden bzw. deren Zuordnung zu einer Serviceunterklasse aufgehoben wurde, als Teil der Prioritätssteuerung nach Verweildauer von laufenden Aktivitäten ermitteln.
Servicesuperklassen	Max. gleichzeitig bestehende Verbindungen (concurrent_connection_top)	Mithilfe dieser Obergrenze für gleichzeitig bestehende Koordinatorverbindungen können Sie den Schwellenwert für gemeinsamen Zugriff für eine Verbindung optimieren.

Tabelle 52. Für Datenbankobjekte unabhängig von der Einstellung COLLECT AGGREGATE ACTIVITY DATA erfasste Statistikdaten (Forts.)

Datenbankobjekt	Statistikdaten	Beschreibung
Workloads	<ol style="list-style-type: none"> 1. Max. gleichzeitig ablaufende Workloadvorkommen (concurrent_wlo_top) 2. Max. gleichzeitig ablaufende Aktivitäten von Workloadvorkommen (concurrent_wlo_act_top) 3. Abgeschlossene Koordinatoraktivitäten insgesamt (coord_act_completed_total) 4. Abgebrochene Koordinatoraktivitäten insgesamt (coord_act_aborted_total) 5. Zurückgewiesene Koordinatoraktivitäten insgesamt (coord_act_rejected_total) 6. Abgeschlossene Workloadvorkommen insgesamt (wlo_completed_total) 7. Aktivitäten insgesamt (act_total) 	<ol style="list-style-type: none"> 1. Mithilfe dieser Obergrenze für gleichzeitig ablaufende Workloadvorkommen können Sie die maximale Anzahl gleichzeitig ablaufender Workloadvorkommen bestimmen und den Schwellenwert für gemeinsamen Zugriff eines Workloadvorkommens einrichten oder optimieren, wenn die Anzahl der zurzeit gleichzeitig ablaufenden Workloadvorkommen zu hoch ist (d. h. zu viele Anwendungen, die derselben Workloaddefinition zugeordnet sind, gleichzeitig auf dem System ausgeführt werden). 2. Mithilfe dieses Elements können Sie die höchste Anzahl gleichzeitig ablaufender Aktivitäten ermitteln, die in einem Member für ein Vorkommen dieser Workload im erfassten Zeitintervall erreicht wurde. 3. Mithilfe dieser Statistik, die den erfolgreichen Abschluss von Aktivitäten misst, können Sie einen Hinweis auf den Status des Systems erhalten. 4. Mithilfe dieser Statistik, die den nicht erfolgreichen Abschluss von Aktivitäten misst, können Sie den Zustand des Systems ermitteln. Aktivitäten können aufgrund eines Abbruchbefehls, von Fehlern oder von reaktiven Schwellenwerten abgebrochen werden. 5. Mithilfe dieser Statistik, die die Zurückweisung von Aktivitäten misst, können Sie die Zweckmäßigkeit einer Zurückweisungsrichtlinie ermitteln. Aktivitäten werden als 'Zurückgewiesen' gewertet, wenn sie gegen einen prädiktiven Schwellenwert verstoßen haben, für den die Aktion STOP EXECUTION definiert wurde, oder wenn sie an der Ausführung durch eine Arbeitsaktion gehindert werden. 6. Mithilfe dieser Statistik können Sie feststellen, wie viele Vorkommen einer Workload in einem bestimmten Zeitraum abgeschlossen werden. 7. Mithilfe dieser Statistik können Sie die Effektivität des Arbeitsaktionssets bestimmen und die relativen Prozentsätze der Typen von Aktivitäten auf dem System ermitteln.
Arbeitsklasse (über eine Arbeitsaktion)	Warteschlangenzuweisungen insgesamt (queue_assignments_total)	Mithilfe dieser Statistik können Sie ermitteln, ob eine zu starke Warteschlangennutzung auftritt oder ob die korrekte Anzahl von Aktivitäten in die Warteschlange gestellt wird (d. h. ob der Schwellenwert für gemeinsamen Zugriff zu restriktiv oder nicht restriktiv genug ist).

Tabelle 52. Für Datenbankobjekte unabhängig von der Einstellung COLLECT AGGREGATE ACTIVITY DATA erfasste Statistikdaten (Forts.)

Datenbankobjekt	Statistikdaten	Beschreibung
Schwellenwertwarteschlangen	<ol style="list-style-type: none"> 1. Max. Warteschlangengröße (queue_size_top) 2. Wartezeit insgesamt (queue_time_total) 	<ol style="list-style-type: none"> 1. Mithilfe dieser Statistik können Sie die maximale Warteschlangengröße ermitteln und feststellen, ob die Warteschlangengröße ausreichend ist. 2. Mithilfe dieser Statistik können Sie ermitteln, wieviel Zeit Aktivitäten in der Warteschlange verbringen und ob diese Zeit zu lang ist.

Wenn Sie den Wert der Option COLLECT AGGREGATE ACTIVITY DATA für eine Serviceunterklasse, eine Workload oder eine Arbeitsklasse (über eine Arbeitsaktion) auf BASE einstellen, werden für jedes Member auch einige der folgenden Statistikdaten erfasst oder die entsprechenden Histogramme generiert. Anhand der Durchschnittswerte können Sie schnell erkennen, wo Aktivitäten ihre meiste Zeit (z. B. in der Warteschlange oder bei der Ausführung) verbringen, und die Antwortzeit (Laufzeit) ermitteln. Sie können die Durchschnittswerte auch zur Optimierung der Histogrammschablonen verwenden. Sie können also den tatsächlichen Durchschnitt mit dem vom Histogramm berechneten Durchschnitt vergleichen und bei einer Abweichung der Werte die Änderung der Histogrammschablone für das entsprechende Histogramm unter Verwendung einer Gruppe von Binwerten in Erwägung ziehen, die für die Daten besser geeignet sind.

Tabelle 53. Erfasste Statistikdaten oder Histogramme, wenn COLLECT AGGREGATE ACTIVITY DATA auf BASE gesetzt ist

Statistik oder Histogramm	Beschreibung
Durchschnittliche Anforderungsausführungszeit (request_exec_time_avg)	Mithilfe dieser Statistik können Sie das arithmetische Mittel der Ausführungszeit für Anforderungen ermitteln, die einer Serviceklasse zugeordnet sind.
Durchschnittliche Laufzeit von Koordinatoraktivitäten (coord_act_lifetime_avg)	Mithilfe dieser Statistik können Sie das arithmetische Mittel der Laufzeit für unverschachtelte Koordinatoraktivitäten ermitteln, die einer Serviceklasse, einer Workload oder einer Arbeitsklasse zugeordnet sind.
Durchschnittliche Ausführungszeit von Koordinatoraktivitäten (coord_act_exec_time_avg)	Mithilfe dieser Statistik können Sie das arithmetische Mittel der Ausführungszeit für unverschachtelte Koordinatoraktivitäten ermitteln, die einer Serviceklasse, einer Workload oder einer Arbeitsklasse zugeordnet sind.
Durchschnittliche Wartezeit von Koordinatoraktivitäten (coord_act_queue_time_avg)	Mithilfe dieser Statistik können Sie das arithmetische Mittel der Wartezeit für unverschachtelte Koordinatoraktivitäten ermitteln, die einer Serviceklasse, einer Workload oder einer Arbeitsklasse zugeordnet sind.
Max. geschätzter Aufwand (cost_estimate_top)	Mithilfe dieser Statistik können Sie die Schwellenwerte für den geschätzten Aufwand optimieren.

Tabelle 53. Erfasste Statistikdaten oder Histogramme, wenn COLLECT AGGREGATE ACTIVITY DATA auf BASE gesetzt ist (Forts.)

Statistik oder Histogramm	Beschreibung
Max. tatsächlich zurückgegebene Zeilen (rows_returned_top)	Mithilfe dieser Statistik können Sie die Schwellenwerte für die tatsächlich zurückgegebenen Zeilen optimieren.
Max. zusammengefasster Tabellenbereich für temporäre Tabellen (agg_temp_tablespace_top)	<p>Mithilfe dieser Statistik können Sie die Nutzung des zusammengefassten Tabellenbereichs für temporäre Systemtabellen optimieren.</p> <p>Diese Statistikdaten werden nur überwacht, wenn Sie einen Schwellenwert für die Nutzung des zusammengefassten Tabellenbereichs für temporäre Tabellen definieren. Für jede angegebene Serviceunterklasse wird diese Statistik überwacht, sobald Sie einen Schwellenwert AGGSQLTEMPSPACE in der Serviceunterklasse selbst definieren oder einen ähnlichen Schwellenwert in einer beliebigen Serviceunterklasse in derselben Servicesuperklasse definieren.</p>
Max. Tabellenbereich für temporäre Tabellen (temp_tablespace_top)	<p>Mithilfe dieser Statistik können Sie die Schwellenwerte für die Nutzung des Tabellenbereichs für temporäre Tabellen optimieren.</p> <p>Diese Statistikdaten werden nur überwacht, wenn Sie einen Schwellenwert für die Nutzung des Tabellenbereichs für temporäre Tabellen definieren. Für jede angegebene Serviceunterklasse wird diese Statistik auch überwacht, sobald Sie einen Schwellenwert AGGSQLTEMPSPACE in der Serviceunterklasse selbst definieren oder einen ähnlichen Schwellenwert in einer beliebigen Serviceunterklasse in derselben Servicesuperklasse definieren.</p>
Histogramm zur Laufzeit der Koordinatoraktivität (CoordActLifetime)	<p>Mithilfe dieses Histogramms können Sie eine Ansicht der Gesamtsystemleistung erhalten.</p> <p>Dieses Histogramm erfasst die Zeitdifferenz (in Millisekunden) zwischen dem Eingang einer Aktivität und der Endzeit für unverschachtelte Koordinatoraktivitäten.</p> <p>Wenn die Aktivität eine Routine ist, die nach Beendigung einen Cursor offen lässt, zählt das Laufzeithistogramm die Laufzeit des Cursors nicht zur Laufzeit der Routine, die das übergeordnete Objekt des Cursors ist.</p>

Tabelle 53. Erfasste Statistikdaten oder Histogramme, wenn COLLECT AGGREGATE ACTIVITY DATA auf BASE gesetzt ist (Forts.)

Statistik oder Histogramm	Beschreibung
Histogramm zur Ausführungszeit der Koordinatoraktivität (CoordActExecTime)	<p>Mithilfe dieses Histogramms können Sie die Auswirkung von Systemänderungen auf die Ausführungszeit messen.</p> <p>Dieses Histogramm erfasst die Ausführungszeit (in Millisekunden) für unverschachtelte Koordinatoraktivitäten.</p> <p>Die Ausführungszeit wird wie folgt berechnet:</p> <ul style="list-style-type: none"> • Bei einem Cursor ist die Ausführungszeit die kombinierte Zeit für die Anforderung zum Öffnen des Cursors, für alle Abrufe und für die Anforderung zum Schließen des Cursors. Zeit, während der der Cursor inaktiv ist, wird nicht zur Ausführungszeit gerechnet. • Bei Routinen ist die Ausführungszeit die Zeit vom Start bis zum Ende des Routinenaufrufs. Wenn Cursor nach der Beendigung der Routine offen bleiben, werden die Laufzeiten dieser Cursor nicht zur Ausführungszeit der Routine gerechnet. • Bei allen anderen Aktivitäten ist die Ausführungszeit die Differenz zwischen der Aktivitätslaufzeit und der Zeit, während der die Aktivität in der Warteschlange wartet.
Histogramm zur Wartezeit der Koordinatoraktivität (CoordActQueueTime)	<p>Mithilfe dieses Histogramms können Sie die Auswirkung von Warteschlangenschwellenwerten auf Aktivitäten messen.</p> <p>Dieses Histogramm erfasst die Zeitdauer (in Millisekunden), während der unverschachtelte Koordinatoraktivitäten in der Warteschlange verbleiben.</p>

Wenn Sie den Wert der Option COLLECT AGGREGATE ACTIVITY DATA für eine Serviceunterklasse, eine Workload oder eine Arbeitsklasse auf EXTENDED einstellen, werden für jedes Member für die entsprechende Serviceklasse oder Arbeitsklasse (über eine Arbeitsaktion) die folgenden Systemstatistikdaten erfasst oder Histogramme generiert. Mithilfe der Durchschnittswerte können Sie schnell die durchschnittliche Eingangsrate von Aktivitäten (Eingangsrate ist das Gegenteil der Zwischenankunftszeit (Inter-Arrival Time)) und den Aufwand von Aktivitäten (geschätzter Aufwand) ermitteln. Sie können die Durchschnittswerte auch zur Optimierung der Histogrammschablonen verwenden. Sie können also den tatsächlichen Durchschnitt mit dem vom Histogramm berechneten Durchschnitt vergleichen und bei einer Abweichung der Werte die Änderung der Histogrammschablone für das entsprechende Histogramm unter Verwendung einer Gruppe von Binwerten in Erwägung ziehen, die für die Daten besser geeignet sind. EXTENDED-Statistikdaten

sind hilfreich für eine detailliertere Leistungsmodellierung. Weitere Informationen finden Sie auch in „Modellieren der Workload-Management-Leistung“ auf Seite 301.

Tabelle 54. Erfasste Statistikdaten oder Histogramme, wenn COLLECT AGGREGATE ACTIVITY DATA auf EXTENDED gesetzt ist

Statistik oder Histogramm	Beschreibung
Durchschnittlicher geschätzter Aufwand für Koordinatoraktivitäten (coord_act_est_cost_avg)	Mithilfe dieser Statistikdaten können Sie das arithmetische Mittel des geschätzten Aufwands von DML-Koordinatoraktivitäten auf Verschachtelungsebene 0, die dieser Serviceunterklasse, Workload oder Arbeitsklasse zugeordnet sind, seit der letzten Zurücksetzung der Statistikdaten ermitteln.
Durchschnittliche Zwischenankunftszeit von unverschachtelten Koordinatoraktivitäten (coord_act_interarrival_time_avg)	Mithilfe dieser Statistik können Sie das arithmetische Mittel der Zeit zwischen dem Eingang einer Koordinatoraktivität auf Verschachtelungsebene 0, die dieser Serviceklasse, Workload oder Arbeitsklasse zugeordnet ist, und der nächsten eingehenden Koordinatoraktivität ermitteln. Der Durchschnittswert wird seit der letzten Zurücksetzung der Statistikdaten berechnet.
Histogramm zum geschätzten Aufwand der Koordinatoraktivität (CoordActEstCost)	Mithilfe dieses Histogramms erhalten Sie eine näherungsweise berechnete Servicezeitverteilung. Dieses Histogramm erfasst den geschätzten Aufwand (in Timeron) für unverschachtelte Koordinatoraktivitäten. Diese Daten sind für die Modellierung Ihres Systems oder für die Eingabe in Anwendungen zur Leistungsmodellierung hilfreich.
Histogramm zur Zwischenankunftszeit der Koordinatoraktivität (CoordActInterArrivalTime)	Mithilfe dieses Histogramms können Sie die Verteilung der Zwischenankunftszeiten für unverschachtelte Koordinatoraktivitäten erhalten. Dieses Histogramm erfasst die Zwischenankunftszeit (in Millisekunden) für unverschachtelte Koordinatoraktivitäten. Diese Daten sind für die Modellierung Ihres Systems oder für die Eingabe in Anwendungen zur Leistungsmodellierung hilfreich.

Die folgende Tabelle gibt an, welche Aktivitätsstatistik für die einzelnen DB2-Workload-Management-Objekte erfasst wird, und umfasst alle verfügbaren zusammengefassten Statistikdaten aus Tabellenfunktionen und Ereignismonitoren. Einige Statistikdaten werden für einige Objekte grundsätzlich erfasst. Andere Statistikdaten werden nur erfasst, wenn eine bestimmte Option COLLECT AGGREGATE angegeben wird. Wenn bei zusammengefassten Aktivitätsstatistiken COLLECT AGGREGATE ACTIVITY DATA EXTENDED angegeben wird, werden auch alle zusammengefassten Aktivitätsdaten vom Typ BASE erfasst.

Tabelle 55. Statistikerfassung für zusammengefasste Aktivitätsdaten für DB2-Workload-Management-Objekte

Objekttyp	Standardmäßig immer erfasste Aktivitätsstatistikdaten	Bei Angabe von COLLECT AGGREGATE ACTIVITY DATA BASE erfasste Aktivitätsstatistikdaten	Bei Angabe von COLLECT AGGREGATE ACTIVITY DATA EXTENDED erfasste Aktivitätsstatistikdaten
Serviceunterklasse	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 Histogramm CoordActLifetime Histogramm CoordActExecTime Histogramm CoordActQueueTime cost_estimate_top rows_returned_top temp_tablespace_top	coord_act_est_cost_avg coord_act_interarrival_time_avg Histogramm CoordActEstCost Histogramm CoordActInterArrivalTime
Service-superklasse	concurrent_connection_top	Nicht zutreffend	Nicht zutreffend
Workload	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 Histogramm CoordActLifetime Histogramm CoordActExecTime Histogramm CoordActQueueTime cost_estimate_top rows_returned_top temp_tablespace_top	coord_act_est_cost_avg coord_act_interarrival_time_avg Histogramm CoordActEstCost Histogramm CoordActInterArrivalTime
Arbeitsklasse (über eine Arbeitsaktion)	act_total	agg_temp_tablespace_top coord_act_lifetime_top coord_act_lifetime_avg coord_act_exec_time_avg coord_act_queue_time_avg Histogramm CoordActLifetime Histogramm CoordActExecTime Histogramm CoordActQueueTime cost_estimate_top rows_returned_top temp_tablespace_top	coord_act_est_cost_avg coord_act_interarrival_time_avg Histogramm CoordActEstCost Histogramm CoordActInterArrivalTime
Schwellenwert	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend

Tabelle 55. Statistikerfassung für zusammengefasste Aktivitätsdaten für DB2-Workload-Management-Objekte (Forts.)

Objekttyp	Standardmäßig immer erfasste Aktivitätsstatistikdaten	Bei Angabe von COLLECT AGGREGATE ACTIVITY DATA BASE erfasste Aktivitätsstatistikdaten	Bei Angabe von COLLECT AGGREGATE ACTIVITY DATA EXTENDED erfasste Aktivitätsstatistikdaten
Schwellenwertwarteschlange	queue_assignments_total queue_size_top queue_time_total	Nicht zutreffend	Nicht zutreffend

Statistikerfassung von zusammengefassten Anforderungsdaten

Wenn Sie den Wert der Option COLLECT AGGREGATE REQUEST DATA für eine Serviceunterklasse auf BASE einstellen, werden die folgenden Statistikdaten für die Serviceunterklasse erfasst.

Tabelle 56. Erfasste Statistikdaten oder Histogramme, wenn COLLECT AGGREGATE REQUEST DATA auf BASE gesetzt ist

Statistik oder Histogramm	Beschreibung
Durchschnittliche Anforderungsausführungszeit (request_exec_time_avg)	Mithilfe dieser Statistik können Sie schnell die durchschnittliche Zeitdauer für die Verarbeitung der einzelnen Anforderungen in einem Member ermitteln und die Histogrammschablone für das entsprechende Histogramm der Anforderungsausführungszeit optimieren.

Tabelle 56. Erfasste Statistikdaten oder Histogramme, wenn COLLECT AGGREGATE REQUEST DATA auf BASE gesetzt ist (Forts.)

Statistik oder Histogramm	Beschreibung
<p>Histogramm der Anforderungsausführungszeit (ReqExecTime)</p>	<p>Anhand dieses Histogramms können Sie feststellen, wo Arbeit ausgeführt wird und ob die Verteilung von Arbeit über Memberr hinweg einheitlich ist.</p> <p>Dieses Histogramm zeigt das Arbeitsvolumen, das in einer Serviceklasse ausgeführt wird, und die Verteilung dieser Arbeit über die Datenbankmember hinweg an. Die Anforderungsausführungszeit (in Millisekunden) wird in einem Histogramm für jedes Member und für alle Anforderungen erfasst.</p> <p>Dieses Histogramm enthält Anforderungen für das Koordinatormember und alle Unteraufforderungen für die Koordinator- und Nicht-Koordinatormember (z. B. RPC-Anforderungen oder Anforderungen von SMP-Subagenten). Die enthaltenen Anforderungen sind einer Aktivität zugeordnet oder nicht: PREPARE- und OPEN-Anforderungen sind beispielsweise in diesem Histogramm enthalten, aber während OPEN-Anforderungen immer einer Cursoraktivität zugeordnet sind, gehören PREPARE-Anforderungen zu keiner Aktivität.</p> <p>Die Anforderungsausführungszeit gibt näherungsweise den Aufwand von Agenten an, die in einer Serviceklasse arbeiten. Koordinatoraktivitätszähler zeigen z. B., dass die meisten Benutzeraktivitäten von einem Member ausgehen, der Koordinatoragent jedoch im Rahmen der Verarbeitung der Aktivitäten Unteraufforderungen an ein anderes Member sendet, das die meiste Arbeit ausführt.</p> <p>Das Histogramm der Anforderungsausführungszeit kann bei der Ermittlung der Größe von Anforderungen hilfreich sein, die an ein Member gesendet werden, d. h., ob die Arbeit, die an das Member gesendet wird, im Wesentlichen aus kleinen oder aus großen Anforderungen besteht oder ob es keine bestimmte Verteilung gibt.</p> <p>Histogramme der Anforderungsausführungszeit sollten nicht für die Aktivitätsantwortzeitanalyse verwendet werden, da Aktivitäten aus einer Reihe von Anforderungen und Unteraufforderungen bestehen können, es keine 1:1-Zuordnung zwischen der Anforderung und Aktivitätsausführungszeit gibt und nicht alle Anforderungen Aktivitäten zugeordnet sind.</p>

Die folgende Tabelle gibt an, welche Anforderungsstatistik für die einzelnen DB2-Workload-Management-Objekte erfasst wird, und umfasst alle verfügbaren zusammengefassten Statistikdaten aus Tabellenfunktionen und Ereignismonitoren. Einige Statistikdaten werden für einige Objekte grundsätzlich erfasst. Andere Statistikdaten werden nur erfasst, wenn die Option COLLECT AGGREGATE REQUEST DATA angegeben wird.

Tabelle 57. Statistikerfassung für zusammengefasste Anforderungsdaten für DB2-Workload-Management-Objekte

Objekttyp	Standardmäßig immer erfasste Anforderungsstatistikdaten	Bei Angabe von COLLECT AGGREGATE REQUEST DATA erfasste Anforderungsstatistikdaten
Serviceunterklasse	num_requests_active	request_exec_time_avg request_exec_time_stddev request_exec_time_total Histogramm ReqExecTime
Servicesuperklasse	Nicht zutreffend	Nicht zutreffend
Workload	Nicht zutreffend	Nicht zutreffend
Arbeitsklasse (über eine Arbeitsaktion)	Nicht zutreffend	Nicht zutreffend
Schwellenwert	Nicht zutreffend	Nicht zutreffend
Schwellenwertwarteschlange	Nicht zutreffend	Nicht zutreffend

Statistikerfassung und Überwachung mit neu zugeordneten Aktivitäten

Die Erfassung von Statistikdaten und die Überwachung werden von der dynamischen Neuordnung von Aktivitäten zwischen Serviceunterklassen beeinflusst.

Eine Neuordnung tritt auf, wenn eine Aktivität, die in einer Serviceklasse ausgeführt wird, in eine andere Serviceklasse verschoben wird, damit die Ausführung fortgesetzt werden kann. Diese Neuordnung kann anhand von Schwellenwerten wie CPUTIMEINSC durchgeführt werden und einen integralen Bestandteil Ihrer Workload-Management-Konfiguration darstellen. Dies ist z. B. bei der Prioritätssteuerung nach Verweildauer der Fall.

Von der Neuordnung von Aktivitäten betroffene Statistikdaten

Als eine Ausnahme zur Regel werden die Aktivitätszwischenankunftszeit, der geschätzte Aufwand sowie die Wartezeit der Unterklasse zugeordnet, in der die Ausführung einer Aktivität beginnt, statt der Unterklasse, in der die Ausführung der Aktivität endet. Da sich eine neu zugeordnete Aktivität auf die Statistikerfassung beider Unterklassen auswirkt, kann in einem Histogramm zur Zwischenankunftszeit, zum geschätzten Aufwand oder zur Wartezeit eine andere Anzahl Aktivitäten gezählt werden als in einem Histogramm zur Laufzeit oder zur Ausführungszeit.

Betrachten wir z. B. eine Aktivität, deren Ausführung in der Serviceunterklasse A beginnt und die später der Serviceunterklasse B zugeordnet wird, in der ihre Ausführung endet. Der geschätzte Aufwand dieser Aktivität ist der Serviceunterklasse A zugeordnet, ihre Laufzeit ist jedoch der Serviceunterklasse B zugeordnet. Daher wurde im Histogramm des geschätzten Aufwands für Unterklasse A ein Element mehr gezählt als im Laufzeithistogramm und für Unterklasse B wurde im Laufzeithistogramm ein Element mehr gezählt als im Histogramm des geschätzten Aufwands.

Als zweite Ausnahme zur Regel kann das Monitorelement **concurrent_act_top** in jeder Unterklasse, die eine Aktivität durchläuft, aktualisiert und dieser zugeordnet werden. Das Monitorelement wird nicht nur erhöht, wenn eine Aktivität beginnt, und verringert, wenn eine Aktivität endet, sondern es wird auch erhöht, wenn eine Aktivität der Unterklasse zugeordnet wird, und verringert, wenn die Zuordnung einer Aktivität zur Unterklasse aufgehoben wird (wenn diese einer anderen Unterklasse zugeordnet wird).

Statistikdaten zur Neuordnung von Aktivitäten

Sie können zwei Monitorelemente verwenden, um die Anzahl der Aktivitäten zu zählen, die aufgrund einer Neuordnungsaktion in eine Serviceunterklasse eingehen oder diese verlassen: **act_remapped_in** und **act_remapped_out**. Die Monitorelemente **act_remapped_in** und **act_remapped_out** zählen die Anzahl Aktivitäten für eine bestimmte Unterklasse in einer beliebigen Partition, die seit der letzten Zurücksetzung der Unterklasse zugeordnet wurden oder deren Zuordnung zur Unterklasse aufgehoben wurde. Sie können anhand dieser Monitorelemente überprüfen, ob die Neuordnung von Aktivitäten zwischen Serviceunterklassen wie erwartet erfolgt.

Zur Ermittlung der Quellen- und der Zielserviceunterklassen für eine Neuordnungsaktion können Sie den Datensatz des Ereignismonitors zum Schwellenwertverstoß verwenden, der eine Zielserviceklassen-ID (**destination_service_class_id**) enthält. Über den Datensatz zum Schwellenwertverstoß können Sie auch die Quellserviceklasse ermitteln.

Überwachung mit Neuordnung von Aktivitäten

Die Neuordnung von Aktivitäten zu anderen Unterklassen beeinflusst die Überwachung dieser Aktivitäten. Damit alle Statistikdaten für eine Aktivität erfasst werden, die in einer Serviceklasse beginnt und aufgrund einer Neuordnung in einer anderen endet, aktivieren Sie beim Erstellen oder Ändern der Serviceklassen die Erfassung der zusammengefassten Aktivitätsdaten sowohl für die Serviceunterklasse, in der die Ausführung der Aktivität beginnt, als auch für die Serviceunterklasse, in der die Ausführung der Aktivität endet. Wenn Sie die Erfassung der zusammengefassten Aktivitätsdaten nur für die Serviceunterklasse aktivieren, in der die Aktivität gestartet wurde, trägt die Aktivität nur zur Wartezeitstatistik und bei erweiterten Statistikdaten zur Statistik zum geschätzten Aufwand und zur Zwischenankunftszeit bei. Wenn Sie die Erfassung der zusammengefassten Aktivitätsdaten nur für die Serviceunterklasse aktivieren, in der die Ausführung der Aktivität endet, trägt die Aktivität nur zur Laufzeit- und zur Ausführungszeitstatistik bei, unabhängig davon, ob beim Absetzen der Anweisung CREATE SERVICE CLASS oder ALTER SERVICE CLASS die Option COLLECT AGGREGATE DATA BASE oder COLLECT AGGREGATE DATA EXTENDED angegeben wurde.

Die folgenden Tabellen fassen zusammen, wie die Statistikerfassung durch die Neuordnungs- und Erfassungseinstellungen beeinflusst wird.

Tabelle 58. Auswirkung der Option COLLECT AGGREGATE DATA BASE bei der Erfassung zusammengefasster Statistikdaten für Unterklassen mit Neuordnung

Statistik	Erfassungseinstellung für Anfangsunterklasse und Endunterklasse			
	NONE und NONE	BASE und NONE	NONE und BASE	BASE und BASE
Laufzeit	Nicht erfasst	Nicht erfasst	Erfasst	Erfasst
Wartezeit	Nicht erfasst	Erfasst	Nicht erfasst	Erfasst

Tabelle 58. Auswirkung der Option COLLECT AGGREGATE DATA BASE bei der Erfassung zusammengefasster Statistikdaten für Unterklassen mit Neuuzuordnung (Forts.)

Statistik	Erfassungseinstellung für Anfangsunterklasse und Endunterklasse			
	Nicht erfasst	Nicht erfasst	Erfasst	Erfasst
Ausführungszeit	Nicht erfasst	Nicht erfasst	Erfasst	Erfasst

Tabelle 59. Auswirkung der Option COLLECT AGGREGATE DATA EXTENDED bei der Erfassung zusammengefasster Statistikdaten für Unterklassen mit Neuuzuordnung

Statistik	Erfassungseinstellung für Anfangsunterklasse und Endunterklasse			
	NONE und NONE	EXTENDED und NONE	NONE und EXTENDED	EXTENDED und EXTENDED
Laufzeit	Nicht erfasst	Nicht erfasst	Erfasst	Erfasst
Wartezeit	Nicht erfasst	Erfasst	Nicht erfasst	Erfasst
Ausführungszeit	Nicht erfasst	Nicht erfasst	Erfasst	Erfasst
Zwischenankunftszeit	Nicht erfasst	Erfasst	Nicht erfasst	Erfasst
Geschätzter Aufwand	Nicht erfasst	Erfasst	Nicht erfasst	Erfasst

Tabelle 60. Auswirkung des Mischens der Optionen COLLECT AGGREGATE DATA BASE und COLLECT AGGREGATE DATA EXTENDED bei der Erfassung zusammengefasster Statistikdaten für Unterklassen mit Neuuzuordnung

Statistik	Erfassungseinstellung für Anfangsunterklasse und Endunterklasse	
	BASE und EXTENDED	EXTENDED und BASE
Laufzeit	Erfasst	Erfasst
Wartezeit	Erfasst	Erfasst
Ausführungszeit	Erfasst	Erfasst
Zwischenankunftszeit	Nicht erfasst	Erfasst
Geschätzter Aufwand	Nicht erfasst	Erfasst

Histogramme beim Workload-Management

Ein Histogramm ist eine Erfassung von Bins (Kategorien von Werten), die Container für die Erfassung von getrennten Datenbereichen sind. Histogramme sind nützlich für eine Vielzahl von Aufgaben bei der Workloadanalyse und der Leistungsoptimierung.

DB2-Workload-Management-Histogramme haben eine feste Anzahl von 41 Bins. Der 40. Bin enthält den höchsten definierten Wert für das Histogramm. Der 41. Bin dient zur Aufnahme von Werten, die über dem höchsten definierten Wert liegen. Jeder Bin stellt einen bestimmten Wertebereich dar, wobei für die Binbereiche eine logarithmische Skala verwendet wird, d. h., die Bins stellen in aufsteigender Reihenfolge, also vom 1. bis zum 40. Bin immer größere Bereiche dar. Die folgende Abbildung zeigt ein Histogramm der Aktivitätslaufzeiten in Form eines Balkendiagramms.

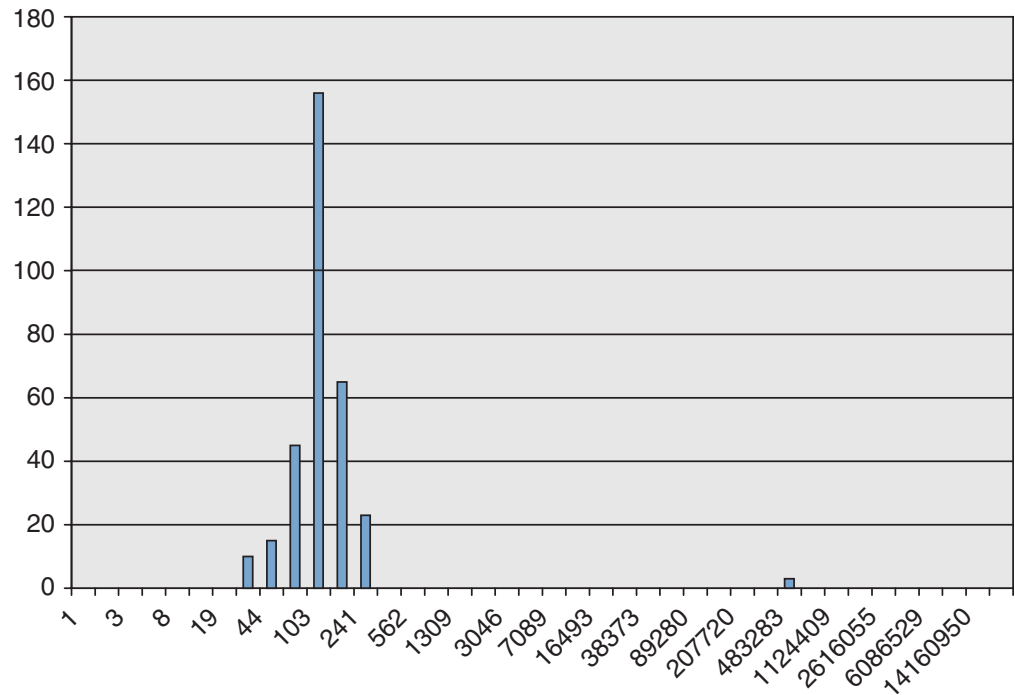


Abbildung 51. Histogramm der Aktivitätslaufzeiten in Form eines Balkendiagramms

Das Histogramm der Aktivitätslaufzeit entspricht den folgenden Daten. Jeder Zähler stellt die Anzahl von Aktivitäten dar, deren Laufzeiten (in Millisekunden) in den Bereich zwischen dem unteren Binwert und dem oberen Binwert fiel. 156 Aktivitäten z. B. hatten eine Laufzeit im Bereich von 68 Millisekunden bis 103 Millisekunden.

Unterer Bin	Oberer Bin	Anzahl
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	Unendlich	0

Sie können Histogramme für eine Reihe unterschiedlicher Zwecke verwenden. Sie können mit ihnen beispielsweise die Verteilung von Werten anzeigen, um Ausreißerwerte zu erkennen oder um Durchschnittswerte und Standardabweichungen zu berechnen. Beispiele für die Verwendung von Histogrammen zum besseren Verständnis und zur besseren Beschreibung Ihrer Workload finden Sie in „Szenario: Optimieren einer DB2-Workload-Management-Konfiguration, wenn keine Kapazitätsplanungsinformationen verfügbar sind“ auf Seite 387 und „Beispiel: Berechnen von Durchschnittswerten und einer Standardabweichung anhand von Histogrammen in einer DB2-Workload-Management-Konfiguration“ auf Seite 284.

In einer Datenbankumgebung mit mehreren Members werden Histogramme in jedem Member erfasst. Bins in Histogrammen haben in allen Datenbankmembers denselben Wertebereich mit bestimmten Anzahlen pro Bin pro Member. Mithilfe der Bins können Sie Informationen auf der Basis einzelner Member analysieren. Sie können auch die Histogramme aller Datenbankmember kombinieren, indem Sie die Anzahlen in den entsprechenden Bins addieren und anhand dieses einzigen Histogramms eine globale Ansicht der Daten erhalten, die Sie dann für Aufgaben wie das Berechnen des globalen Durchschnitts und der Standardabweichung verwenden können.

Histogramme sind für Serviceunterklassen, Workloads und Arbeitsklassen über Arbeitsaktionen verfügbar. Für diese Objekte werden Histogramme erfasst, wenn Sie beim Erstellen oder Ändern der Objekte eine der Klauseln COLLECT AGGREGATE ACTIVITY DATA, COLLECT AGGREGATE REQUEST DATA oder COLLECT AGGREGATE UNIT OF WORK DATA angeben. Für Arbeitsklassen werden Histogramme auch erfasst, wenn Sie eine Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA auf die Arbeitsklasse anwenden. Die folgenden Histogramme sind verfügbar:

- Laufzeit unverschachtelter Koordinatoraktivitäten, wenn Sie AGGREGATE ACTIVITY DATA BASE oder AGGREGATE ACTIVITY DATA EXTENDED für eine Serviceunterklasse, für eine Workload oder für eine auf die Arbeitsklasse angewendete Arbeitsaktion angeben.
- Ausführungszeit unverschachtelter Koordinatoraktivitäten, wenn Sie AGGREGATE ACTIVITY DATA BASE oder AGGREGATE ACTIVITY DATA EXTENDED für eine Serviceunterklasse, für eine Workload oder für eine auf die Arbeitsklasse angewendete Arbeitsaktion angeben.
- Wartezeit unverschachtelter Koordinatoraktivitäten, wenn Sie AGGREGATE ACTIVITY DATA BASE oder AGGREGATE ACTIVITY DATA EXTENDED für eine Serviceunterklasse, für eine Workload oder für eine auf die Arbeitsklasse angewendete Arbeitsaktion angeben.
- Anforderungsausführungszeit, wenn Sie AGGREGATE REQUEST DATA BASE für eine Serviceunterklasse angeben. Dieses Histogramm wird nicht auf Workloads oder Arbeitsklassen angewendet.

- Histogramm der Zwischenankunftszeit unverschachtelter Aktivitäten, wenn Sie AGGREGATE ACTIVITY DATA EXTENDED für eine Serviceunterklasse, für eine Workload oder für eine auf die Arbeitsklasse angewendete Arbeitsaktion angeben.
- Geschätzter Aufwand für unverschachtelte DML-Aktivitäten, wenn Sie AGGREGATE ACTIVITY DATA EXTENDED für eine Serviceunterklasse, für eine Workload oder für eine auf die Arbeitsklasse angewendete Arbeitsaktion angeben.
- UOW-Laufzeit, wenn Sie AGGREGATE UNIT OF WORK DATA BASE für eine Serviceklasse angeben.

Alle aktivitätsbezogenen Histogramme enthalten Informationen zu Aktivitäten, die abgeschlossen, abgebrochen oder zurückgewiesen wurden.

Histogrammschablonen

Sie können optional eine Histogrammschablone angeben, anhand derer ermittelt wird, wie ein bestimmtes Histogramm inklusive des oberen Binwerts aussieht. Eine Histogrammschablone ist ein Objekt, dem keine vordefinierte Maßeinheit zugewiesen ist. Dem Histogramm wird entsprechend dem Kontext, in dem die Histogrammschablone verwendet wird, eine Maßeinheit zugeordnet, wenn eine Serviceklasse, eine Workload oder eine Arbeitsaktion erstellt oder geändert wird. Zeitbasierte Histogramme wie die über die Klausel ACTIVITY LIFETIME HISTOGRAM angegebenen Histogramme beispielsweise verwenden Millisekunden als Einheit, während aufwandsbasierte Histogramme wie die über die Klausel ACTIVITY ESTIMATED COST HISTOGRAM angegebenen Histogramme Timeron als Einheit verwenden.

Sie können eine Histogrammschablone erstellen, indem Sie die Anweisung CREATE HISTOGRAM TEMPLATE verwenden und den maximalen oberen Binwert angeben. Alle anderen Binwerte werden automatisch als exponentiell ansteigende Werte definiert, die sich dem oberen Binwert annähern. Wenn Sie z. B. eine Histogrammschablone mit einem oberen Binwert von 3 000 000 erstellen wollen, setzen Sie eine Anweisung wie die folgende ab:

```
CREATE HISTOGRAM TEMPLATE TEMPLATE1 HIGH BIN VALUE 3000000
```

Diese Anweisung erstellt eine Histogrammschablone mit den folgenden Binwerten:

Unterer Bin	Oberer Bin
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	Unendlich

Sie wenden eine Histogrammschablone mit dem entsprechenden Schlüsselwort HISTOGRAM TEMPLATE an, wenn Sie Serviceunterklassen, Workloads oder Arbeitsaktionen erstellen oder ändern. Wenn Sie keine Histogrammschablone angeben, wird die Standardschablone, SYSDEFAULTHISTOGRAM, verwendet. Wenn Sie die Erfassung mit AGGREGATE ACTIVITY DATA für ein Objekt nicht aktivieren, wird die Histogrammschablone ignoriert.

Wenn Sie z. B. die Histogrammschablone TEMPLATE1 für das vorhandene Histogramm der Aktivitätslaufzeit der Serviceunterklasse MYSUBCLASS unter der Servicesuperklasse MYSUPERCLASS verwenden wollen, setzen Sie die folgende Anweisung ab:

```
ALTER SERVICE CLASS MYSUBCLASS UNDER MYSUPERCLASS
ACTIVITY LIFETIME HISTOGRAM TEMPLATE TEMPLATE1
```

Nachdem Sie ein Commit für die Anweisung ALTER SERVICE CLASS durchgeführt haben, hat das Histogramm der Aktivitätslaufzeit, das für die Serviceunterklasse MYSUBCLASS erfasst wird, obere Binwerte, die von der Histogrammschablone TEMPLATE1 statt von der Histogrammschablone SYSDEFAULTHISTOGRAM bestimmt sind.

Eine Änderung der Angabe der Histogrammschablone oder die Änderung einer Histogrammschablone für eine Serviceklasse oder eine Workload wird erst nach der Zurücksetzung der Statistikdaten wirksam.

Sie können eine Histogrammschablone mit der Anweisung DROP HISTOGRAM TEMPLATE löschen.

Sie können die Histogrammschablonen durch Abfragen der Sicht SYSCAT.HISTOGRAMTEMPLATES und die entsprechenden oberen Binwerte der Histogrammschablone durch Abfragen der Sicht SYSCAT.HISTOGRAMTEMPLATEBINS anzeigen. Der untere Binwert ist immer 0 für den ersten Bin; bei allen anderen Bins ist der untere Binwert immer der obere Binwert des vorherigen Bins.

Beispiel

Das folgende Beispiel erzeugt eine Tabellenfunktion zur Berechnung des Histogramms CoordActLifetime, CoordActExecTime, CoordActQueueTime oder CoordActEstCost für eine Servicesuperklasse als Ganzes durch Addition über alle Unterklassen. Die Addition über Subklassen ist hilfreich, wenn Aktivitäten unter

derselben Servicesuperklasse anderen Serviceunterklassen zugeordnet werden, wie dies in einem Szenario mit Prioritätssteuerung nach Verweildauer auftreten kann, in dem Serviceklassenschichten und spezielle Schwellenwerte zur dynamischen Steuerung von Ressourcen für Aktivitäten verwendet werden. Dieses Beispiel gilt nicht für das Histogramm CoordActInterArrivalTime, da die berechneten gewichteten Mittel nicht die Tatsache berücksichtigen, dass das Histogramm CoordActInterArrivalTime einer Unterklasse die Zeit zwischen der Ankunft einer Abfrage in dieser Unterklasse und der nächsten Abfrage misst, das Histogramm CoordActInterArrivalTime einer Superklasse jedoch die Zeit zwischen der Ankunft einer Abfrage in einer beliebigen Unterklasse und der nächsten Abfrage misst.

```
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
```

Die Ausgabe sieht folgendermaßen aus:

```
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 record(s) selected.

Erstellen einer Histogrammschablone

Mit der Anweisung CREATE HISTOGRAM TEMPLATE können Sie eine Histogrammschablone erstellen. Histogrammschablonen werden von Serviceunterklassen und Arbeitsaktionen verwendet, um die Binwerte für die Statistikdaten festzulegen, die mithilfe von Histogrammen gepflegt werden.

Vorbereitende Schritte

Zum Erstellen einer Histogrammschablone benötigen Sie die Berechtigung WL-MADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in den folgenden Abschnitten:

- „DDL-Anweisungen für Workload-Management“ auf Seite 18
- Anhang A, „Allgemeine Namenskonventionen“, auf Seite 541

Informationen zu diesem Vorgang

Einige Statistikdaten zu DB2-Serviceunterklassen, Arbeitsklassenaktivitäten und Anforderungen werden mithilfe von Histogrammen erfasst. Alle Histogramme haben eine feste Anzahl Bins und jeder Bin stellt einen Bereich dar, in dem Aktivitäten oder Anforderungen gezählt werden. Der für die Bins verwendete Einheitentyp hängt vom Typ des erstellten Histogramms ab. Die Histogrammschablone beschreibt den höchsten Wert des vorletzten Bins im Histogramm, der sich auf die Werte aller Bins im Histogramm auswirkt. Weitere Informationen zu Histogrammen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Histogrammschablone zu erstellen:

1. Setzen Sie die Anweisung CREATE HISTOGRAM TEMPLATE ab und geben Sie den Namen der Histogrammschablone, die Sie erstellen wollen, sowie einen Wert für das Schlüsselwort HIGH BIN VALUE an, um den höchsten Wert für den zweitletzten Bin festzulegen.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird das Histogramm zum Sicht SYSCAT.HIS-

TOGRAMTEMPLATES hinzugefügt und die Bins werden zur Sicht SYSCAT.HISTOGRAMTEMPLATEBINS hinzugefügt.

Ändern einer Histogrammschablone

Mit der Anweisung ALTER HISTOGRAM TEMPLATE können Sie eine vorhandene Histogrammschablone ändern. Histogrammschablonen werden von Serviceunterklassen und Arbeitsaktionen verwendet, um die Binwerte für die Statistikdaten festzulegen, die mithilfe von Histogrammen gepflegt werden.

Vorbereitende Schritte

Zum Ändern einer Histogrammschablone benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Informationen zu diesem Vorgang

Einige Statistikdaten zu DB2-Serviceunterklassen, Arbeitsklassenaktivitäten und Anforderungen werden mithilfe von Histogrammen erfasst. Alle Histogramme haben eine feste Anzahl Bins und jeder Bin stellt einen Bereich dar, in dem Aktivitäten oder Anforderungen gezählt werden. Der für die Bins verwendete Einheitentyp hängt vom Typ des Histogramms ab, das Sie erstellen. Die Histogrammschablone beschreibt den höchsten Wert des vorletzten Bins im Histogramm, der sich auf die Werte aller Bins im Histogramm auswirkt. Weitere Informationen zu Histogrammen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Histogrammschablone zu ändern:

1. Setzen Sie die Anweisung ALTER HISTOGRAM TEMPLATE ab und geben Sie den Namen der Histogrammschablone, die Sie ändern wollen, sowie einen Wert für den Parameter HIGH BIN VALUE an, um den höchsten Wert für den zweitletzten Bin zu ändern.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird der obere Binwert für das Histogramm in der Sicht SYSCAT.HISTOGRAMTEMPLATEBINS aktualisiert. Die Änderung wird erst bei der nächsten Zurücksetzung der Workload-Management-Statistikdaten wirksam. Weitere Informationen finden Sie in „Zurücksetzen von Statistikdaten für DB2-Workload-Management-Objekte“ auf Seite 289.
3. Optional: Führen Sie die gespeicherte Prozedur WLM_COLLECT_STATS aus, um die Statistikdaten zu erfassen und zurückzusetzen, sodass die neue Histogrammschablone sofort verwendet werden kann.

Löschen einer Histogrammschablone

Wenn Sie eine Histogrammschablone nicht mehr benötigen, können Sie sie löschen.

Vorbereitende Schritte

Zum Löschen einer Histogrammschablone benötigen Sie die Berechtigung WLMADM oder DBADM.

Weitere Informationen zu Voraussetzungen finden Sie in „DDL-Anweisungen für Workload-Management“ auf Seite 18.

Die Histogrammschablone SYSDEFAULTHISTOGRAM kann nicht gelöscht werden.

Sie können eine Histogrammschablone nicht löschen, wenn eine Serviceunterklasse, eine Arbeitsaktion oder eine Workload darauf verweist. Sie können die Serviceunterklassen und Arbeitsaktionen, die auf eine Histogrammschablone verweisen, durch Abfragen der Sicht SYSCAT.HISTOGRAMTEMPLATESUSE anzeigen.

Vorgehensweise

Gehen Sie wie folgt vor, um eine Histogrammschablone zu löschen:

1. Verwenden Sie die Anweisung DROP HISTOGRAM TEMPLATE.
2. Führen Sie ein Commit für Ihre Änderungen durch. Beim Durchführen eines Commits für Ihre Änderungen wird das Histogramm aus der Sicht SYSCAT.HISTOGRAMTEMPLATES entfernt und die zugehörigen Bins werden aus der Sicht SYSCAT.HISTOGRAMTEMPLATEBINS entfernt.

Beispiel: Berechnen von Durchschnittswerten und einer Standardabweichung anhand von Histogrammen in einer DB2-Workload-Management-Konfiguration

Eine Verwendungsmöglichkeit von Histogrammen ist die Ermittlung der Standardabweichung für Aktivitätslaufzeiten. Das Beispiel in diesem Abschnitt zeigt, wie Bins für die Berechnung dieser Statistikdaten verwendet werden.

Eine Berechnung der durchschnittlichen Laufzeit für jede Aktivität ist eine nützliche Information. Der Durchschnitt allein beschreibt jedoch nicht präzise, wie diese vom Benutzer erlebt wird. Wenn die Streuung in der Aktivitätslaufzeit groß ist, können die von Ihnen unterstützten Benutzer manchmal sehr schnelle Abfragen erleben (was erstrebenswert ist) und manchmal sehr langsame (was eventuell nicht akzeptabel ist). Wenn Sie ein Ziel für die Aktivitätslaufzeiten definieren, ist nicht nur die durchschnittliche Laufzeit der Aktivitäten wichtig, sondern auch die Standardabweichung der Aktivitätslaufzeit. Sie müssen die Streuung verstehen und kontrollieren, um sicherzustellen, dass Ihre Benutzer tatsächlich den beobachteten Durchschnittswert erleben.

In einer DB2-Workload-Management-Konfiguration werden Statistikdaten für jedes Member erfasst. Das folgende Beispiel zeigt, wie Sie die durchschnittliche Aktivitätslaufzeit für ein einzelnes Member ermitteln können.

Nehmen wir an, Sie haben eine Umgebung mit einem einzigen Member sowie ein Histogramm mit den folgenden Bins. In den echten Histogrammen sind mehr Bins enthalten. In diesem Beispiel ist die Anzahl der Bins jedoch zur Vereinfachung auf 8 beschränkt.

Bin 1 - 0 bis 2 Sekunden
Bin 2 - 2 bis 4 Sekunden
Bin 3 - 4 bis 8 Sekunden
Bin 4 - 8 bis 16 Sekunden
Bin 5 - 16 bis 32 Sekunden
Bin 6 - 32 bis 64 Sekunden
Bin 7 - 64 bis 128 Sekunden
Bin 8 - 128 Sekunden bis Unendlich

Sie können einen Näherungswert des Durchschnitts berechnen, indem Sie annehmen, dass die durchschnittliche Antwortzeit für eine Abfrage, die in einen Bin mit dem Bereich x bis y fällt, $(x + y)/2$ ist. Sie können dann diese Zahl mit der Anzahl der Abfragen im Bin multiplizieren, die Summe über alle Bins bilden und dann die Summe durch die Gesamtzahl dividieren. Im vorstehenden Beispiel nehmen wir an, dass sich eine durchschnittliche Antwortzeit für jeden Bin wie folgt ergibt:

Bin 1 durchschnittliche Laufzeit = $(0+2)/2 = 1$
 Bin 2 durchschnittliche Laufzeit = $(2+4)/2 = 3$
 Bin 3 durchschnittliche Laufzeit = $(4+8)/2 = 6$
 Bin 4 durchschnittliche Laufzeit = $(8+16)/2 = 12$
 Bin 5 durchschnittliche Laufzeit = $(16+32)/2 = 24$
 Bin 6 durchschnittliche Laufzeit = $(32+64)/2 = 48$
 Bin 7 durchschnittliche Laufzeit = $(64+128)/2 = 96$

Nehmen wir an, dass das folgende Histogramm im Abrechnungszeitraum erfasst wurde:

Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6	Bin 7	Bin 8
Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl
20	30	80	10	5	3	2	0

Zur Berechnung der durchschnittlichen Laufzeit muss Bin 8 leer sein. Bin 8 ist nur vorhanden, damit Sie erkennen können, wann Sie die obere Begrenzung Ihres Bereichs ändern müssen. Aus diesem Grund müssen Sie die Obergrenze für den Bereich angeben.

Sie können die durchschnittliche Laufzeit für Member 1 wie folgt näherungsweise berechnen:

$$\begin{aligned} \text{Durchschnittliche Laufzeit} &= (20 \times 1 + 30 \times 3 + 80 \times 6 + 10 \times 12 + 5 \times 24 + 3 \times 48 + 2 \times 96) / 150 \\ &= (20 + 90 + 480 + 120 + 120 + 144 + 192) / 150 \\ &= 1166 / 150 \\ &= 7,77 \text{ Sekunden} \end{aligned}$$

Sie können die Standardabweichung der Laufzeit wie folgt näherungsweise berechnen:

$$\text{Standardabweichung} = [(20 \times (1 - 7,77)^2 + 30 \times (3 - 7,77)^2 + \dots) / 150]^{1/2}$$

In Datenbankumgebungen mit mehreren Membern können Durchschnitte und Standardabweichungen berechnet werden, indem zuerst ein kombiniertes Histogramm über alle Member durch Addition der Anzahl aller Bins über die Datenbankmember berechnet wird.

Nehmen wir z. B. an, dass die Datenbank zwei Member hat, die Binsgrößen im Histogramm wie oben beschrieben sind und das Histogramm die folgenden Daten enthält:

Datenbank-member	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6	Bin 7	Bin 8
	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl
1	20	30	80	10	5	3	2	0
2	1	5	20	20	4	0	0	0

Da die Binsgrößen datenbankmemberübergreifend gleich sind, ist das Gesamthistogramm einfach zu berechnen:

Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6	Bin 7	Bin 8
Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl	Anzahl
21	35	100	30	9	3	2	0

Aus dem kombinierten Histogramm können Sie den Gesamtdurchschnitt der Laufzeit und die Gesamtstandardabweichung auf ähnliche Weise berechnen wie bei einer Umgebung mit einem Member:

$$\begin{aligned} \text{Durchschnittliche Laufzeit} &= (21 \times 1 + 35 \times 3 + 100 \times 6 + 30 \times 12 + 9 \times 24 + 3 \times 48 + 2 \times 96) / 200 \\ &= (21 + 105 + 600 + 360 + 216 + 144 + 192) / 200 \\ &= 1638 / 200 \\ &= 8,19 \text{ Sekunden} \end{aligned}$$

$$\text{Standardabweichung} = [(21 \times (1 - 8,19)^2 + 35 \times (3 - 7,77)^2 + \dots) / 200]^{1/2}$$

Protokollanalysetool

Ihre DB2-Datenserverinstallation umfasst ein Paar Perl-Beispielscripts, die Informationen dazu generieren, auf welche Tabellen, Indizes und Spalten von der Protokollanalyse zugegriffen bzw. nicht zugegriffen wurde.

Diese Scripts stellen eine Protokollanalysefunktionalität unter Verwendung der vom Workload-Management-Aktivitätsereignismonitor erfassten Informationen bereit. Das Protokollanalysetool des Workload-Managements ist in Perl geschrieben; Sie können diese Scripts direkt verwenden oder zur Erstellung zusätzlicher Protokollanalyseberichte für Ihre Bedürfnisse ändern.

Das Protokollanalysetool des Workload-Managements besteht aus zwei Scripts. Diese befinden sich im Pfad `samples/perl` Ihres Installationsverzeichnis:

- `wlmhist.pl` - generiert Protokolldaten
- `wlmhistrep.pl` - erzeugt Berichte aus den Protokolldaten

Eine Datei `DB2WlmHist.pm`, die von den beiden Scripts verwendete allgemeine Perl-Routinen enthält, ist ebenfalls enthalten.

Weitere Informationen zum Einrichten und Ausführen der Scripts finden Sie in der Datei `README_WLMHIST` im selben Verzeichnis.

Erfassen von Workload-Management-Statistikdaten mithilfe eines Ereignismonitors für Statistikdaten

Statistikdaten für DB2-Workload-Management-Objekte können für eine Protokollanalyse an einen Statistikereignismonitor gesendet werden.

Informationen zu diesem Vorgang

Anhand von Statistikdaten können Sie das Verhalten Ihres Systems im Lauf der Zeit erkennen (z. B. die durchschnittliche Laufzeit von Aktivitäten, die Wartezeit von Aktivitäten in der Warteschlange, die Verteilung von großen und kleinen Aktivitäten usw.), Schwellenwerte festlegen (z. B. die Obergrenze für gleichzeitig ablaufende Aktivitäten finden) und Probleme erkennen (z. B. ob die von Benutzern erlebte durchschnittliche Laufzeit höher als normal ist). Eine Beschreibung der für jedes DB2-Workload-Management-Objekt erfassten Statistikdaten finden Sie in „Statistikdaten für DB2-Workload-Management-Objekte“ auf Seite 264.

Sie können automatisch Workload-Management-Statistikdaten an einen Ereignismonitor entweder in einem festen Zeitintervall oder manuell zu einem beliebigen Zeitpunkt senden.

Vorgehensweise

Gehen Sie wie folgt vor, um automatisch Workload-Management-Statistikdaten in einem festen Zeitintervall zu erfassen:

1. Erstellen Sie mit der Anweisung `CREATE EVENT MONITOR` einen `STATISTICS`-Ereignismonitor. Sie könnten z. B. die folgende Anweisung absetzen:
`CREATE EVENT MONITOR STATS1 FOR STATISTICS WRITE TO TABLE`
2. Verwenden Sie die Anweisung `COMMIT`, um die Änderungen festzuschreiben.
3. Aktivieren Sie den Ereignismonitor mithilfe der Anweisung `SET EVENT MONITOR STATE`. Statt der Anweisung `SET EVENT MONITOR STATE` können Sie auch die Standardeinstellung `AUTOSTART` für den `STATISTICS`-Ereignismoni-

tor verwenden, um ihn bei der nächsten Aktivierung der Datenbank aktivieren zu lassen. Wenn Sie mehrere STATISTICS-Ereignismonitore definieren wollen, sollten Sie nicht die Option AUTOSTART verwenden.

4. Verwenden Sie die Anweisung COMMIT, um die Änderungen festzuschreiben.
5. Optional: Aktivieren Sie die Erfassung zusätzlicher Statistikdaten. Standardmäßig wird nur eine minimale Menge von Statistikdaten für jedes DB2-Workload-Management-Objekt erfasst. Eine Beschreibung, welche Statistikdaten standardmäßig für die einzelnen Objekte erfasst werden, finden Sie in „Statistikdaten für DB2-Workload-Management-Objekte“ auf Seite 264. Geben Sie die Erfassung von zusammengefassten Aktivitätsdaten für Serviceunterklassen, Workloads und Arbeitsklassen mit dem Schlüsselwort COLLECT AGGREGATE ACTIVITY DATA in den Anweisungen ALTER SERVICE CLASS und ALTER WORK ACTION SET an. Geben Sie die Erfassung von zusammengefassten Anforderungsdaten für Serviceunterklassen mit dem Schlüsselwort COLLECT AGGREGATE REQUEST DATA in der Anweisung ALTER SERVICE CLASS an. Führen Sie einen COMMIT für die Änderungen durch.
6. Geben Sie ein Erfassungsintervall an, indem Sie den Datenbankkonfigurationsparameter **wlm_collect_int** aktualisieren. Der Parameter **wlm_collect_int** gibt ein Zeitintervall in Minuten an. In jedem Intervall wird die Kopie der Workload-Management-Statistikdaten für alle DB2-Workload-Management-Objekte in den aktiven Statistikereignismonitor geschrieben und die Statistikdaten werden zurückgesetzt. In einer Datenbankumgebung mit mehreren Membern muss der Parameter **wlm_collect_int** im Katalogmember aktualisiert werden. Dieser Parameter kann dynamisch aktualisiert werden. Beispiel:

```
CONNECT TO aliasname_der_datenbank
UPDATE DATABASE CONFIGURATION USING WLM_COLLECT_INT 5 IMMEDIATE
```

Ergebnisse

Nachdem Sie die oben aufgelisteten Schritte ausgeführt haben, werden Workload-Management-Statistikdaten alle **wlm_collect_int** Minuten in den Statistikereignismonitor geschrieben. Jeder in den Statistikereignismonitor geschriebene Datensatz hat einen Wert STATISTICS_TIMESTAMP und einen Wert LAST_WLM_RESET. Das Zeitintervall von LAST_WLM_RESET bis STATISTICS_TIMESTAMP definiert das Erfassungsintervall (d. h. das Zeitintervall, in dem die Statistikdaten im Datensatz erfasst wurden).

Sammlungen erfolgen zu den angegebenen Intervallzeiten und sind in Relation zu Sonntag 00:00:00 gesetzt. Nach der Aktivierung des Katalogmembers erfolgt die nächste Erfassung zu Beginn des nächsten nach der Aktivierung terminierten Intervalls. Das terminierte Intervall ist nicht relativ zur Aktivierungszeit des Katalogmembers. Ist ein Member zum Zeitpunkt der Erfassung nicht aktiv, werden für dieses Member keine Statistikdaten zusammengestellt. Wurde der Intervallwert beispielsweise auf 60 gesetzt und das Katalogmember wurde am Sonntag um 9:24 Uhr aktiviert, erfolgen die Erfassungen stündlich zur vollen Stunde. Die nächste Erfassung erfolgt daher um 10:00 Uhr. Wenn das Member um 10:00 Uhr nicht aktiv ist, werden für dieses Member keine Statistikdaten zusammengestellt.

Wenn der Parameter **wlm_collect_int** auf einen Wert ungleich null gesetzt wird und es keinen aktiven Statistikereignismonitor gibt, werden die Workload-Management-Statistikdaten dennoch alle **wlm_collect_int** Minuten zurückgesetzt, die Statistikdaten werden jedoch nicht erfasst. Die Daten gehen verloren. Deshalb empfiehlt es sich nicht, einen Wert für **wlm_collect_int** ungleich null anzugeben, ohne einen Statistikereignismonitor zu aktivieren.

Wenn der Parameter `wlm_collect_int` auf 0 (Standardeinstellung) gesetzt ist, werden keine Statistikdaten automatisch an den Statistikereignismonitor gesendet. Sie können mit der gespeicherten Prozedur `WLM_COLLECT_STATS` manuell Statistikdaten zur späteren Protokollanalyse an den Statistikereignismonitor senden. Wenn diese Prozedur aufgerufen wird, führt sie dieselben Aktionen aus wie eine automatische intervallgesteuerte Statistikdatenerfassung, d. h., die Statistikdaten werden an den Statistikereignismonitor gesendet und dann im Hauptspeicher zurückgesetzt. Wenn kein Statistikereignismonitor aktiv ist, werden die Werte zurückgesetzt, aber keine Daten erfasst. Wenn Sie nur die Statistikdaten zurücksetzen wollen, können Sie die Prozedur `WLM_COLLECT_STATS` aufrufen, während kein Statistikereignismonitor aktiv ist.

Die manuelle Erfassung von Statistikdaten greift nicht in die automatische Erfassung von Statistikdaten ein. Angenommen, Sie haben `wlm_collect_int` auf 60 gesetzt. Jede Stunde werden nun die Statistikdaten an den Statistikereignismonitor gesendet. Nehmen wir weiter an, dass die Statistikdaten zum letzten Mal um 5:00 Uhr erfasst wurden. Sie können die Prozedur `WLM_COLLECT_STATS` um 5:55 Uhr aufrufen, wodurch die Werte der Statistikdaten an den Ereignismonitor gesendet und die Statistikdaten zurückgesetzt werden. Die nächste automatische Statistikerfassung erfolgt dennoch um 6:00 Uhr, eine Stunde nach der letzten automatisierten Erfassung. Das Erfassungsintervall wird nicht von einer manuellen Erfassung und Zurücksetzung von Statistikdaten beeinflusst, die innerhalb des Intervalls durchgeführt wurde.

Hinweise:

- Die Tabellenfunktionen für DB2-Workload-Management-Statistikdaten listen die aktuellen Werte der Statistikdaten auf. Wenn Sie die automatische Workload-Management-Statistikerfassung aktiviert haben, werden diese Werte regelmäßig in dem mit dem Datenbankkonfigurationsparameter `wlm_collect_int` definierten Intervall zurückgesetzt. Wenn Sie sich die von den Tabellenfunktionen aufgelisteten Statistikdaten ansehen, sollten Sie immer auch die Spalte `LAST_RESET` berücksichtigen. Diese Spalte gibt an, wann die Statistikdaten zuletzt zurückgesetzt wurden. Wenn das Zeitintervall zwischen der letzten Zurücksetzung und der aktuellen Uhrzeit nicht groß genug ist, sind eventuell nicht ausreichend Daten vorhanden, um aussagekräftige Schlüsse ziehen zu können.
- Wenn Sie die automatische Erfassung von Workload-Management-Statistikdaten verwenden, müssen Sie Ihre Ereignismonitordateien oder -tabellen regelmäßig bereinigen. Der Ereignismonitor bereinigt die erfassten Daten nicht automatisch, sodass Ihre Dateien oder Tabellen im Lauf der Zeit durch die automatische Erfassung gefüllt werden.
- Wenn eine Datenbank inaktiviert wird, werden die Statistikdaten zurückgesetzt. Beim Inaktivieren der Datenbank werden die Statistikdaten jedoch nicht an den Statistikereignismonitor gesendet. Wenn Sie die seit der letzten Erfassung aufgelaufenen Statistikdaten bei der Inaktivierung nicht verlieren wollen, sollten Sie manuell die Prozedur `WLM_COLLECT_STATS` vor dem Inaktivieren der Datenbank aufrufen.
- Die Prozedur `WLM_COLLECT_STATS` setzt die Statistikdaten anders zurück als der Befehl **RESET MONITOR**. Der Befehl **RESET MONITOR** setzt die Werte der Elemente von Snapshot Monitor zurück, indem er ihre aktuellen Werte speichert. Nachdem der Befehl **RESET MONITOR** abgesetzt wurde, listet die Momentaufnahmenverarbeitung die Differenz zwischen diesen Werten und den aktuellen Werten auf. Dagegen werden bei der durch die Prozedur `WLM_COLLECT_STATS` veranlassten Zurückset-

zung keine Werte gespeichert, sondern alle Statistikzähler selbst für jedes zutreffende DB2-Workload-Management-Objekt zurückgesetzt.

Außerdem hat beim Befehl **RESET MONITOR** jeder Prozess (Anhang) seine eigene Sicht der Überwachungsdaten. Wenn der Benutzer eine Zurücksetzung ausführt, sind andere Benutzer davon nicht betroffen. Eine Zurücksetzung der Workload-Management-Statistikdaten wirkt sich dagegen auf alle Benutzer aus.

Zurücksetzen von Statistikdaten für DB2-Workload-Management-Objekte

In diesem Abschnitt wird beschrieben, wie Statistikdaten für DB2-Workload-Management-Objekte zurückgesetzt werden können.

Nur DB2-Workload-Management-Statistikdaten können zurückgesetzt werden. Von Überwachungsschnittstellen zurückgegebene Messdaten werden erfasst, aber nicht zurückgesetzt.

Vier Ereignisse setzen die Statistikdaten für jedes DB2-Workload-Management-Objekt zurück. (Eine Beschreibung der für jedes Objekt erfassten Statistikdaten finden Sie in „Statistikdaten für DB2-Workload-Management-Objekte“ auf Seite 264.)

- Die gespeicherte Prozedur `WLM_COLLECT_STATS` wird aufgerufen. Details finden Sie in „Erfassen von Workload-Management-Statistikdaten mithilfe eines Ereignismonitors für Statistikdaten“ auf Seite 286.
- Der Prozess zum automatischen Erfassen und Zurücksetzen von DB2-Workload-Management-Statistikdaten, der vom Datenbankkonfigurationsparameter `wlm_collect_int` gesteuert wird, veranlasst eine Erfassung und Zurücksetzung. Details finden Sie in „Erfassen von Workload-Management-Statistikdaten mithilfe eines Ereignismonitors für Statistikdaten“ auf Seite 286.
- Die Datenbank wird reaktiviert. Bei jeder Aktivierung einer Datenbank in einem Member werden die Statistikdaten für alle DB2-Workload-Management-Objekte in diesem Member zurückgesetzt.
- Das Objekt, für das die Statistikdaten erfasst werden, wird geändert und für die Änderung wird ein `COMMIT` durchgeführt. Wenn z. B. eine Serviceunterklasse geändert wird, werden die Statistikdaten für diese Serviceunterklasse beim Durchführen eines `Commits` für die Anweisung `ALTER` zurückgesetzt.

Sie können den Zeitpunkt der letzten Zurücksetzung der Statistikdaten für ein bestimmtes DB2-Workload-Management-Objekt mit den Statistiktabellenfunktionen und über die Zeitmarke in der Spalte `LAST_RESET` ermitteln. Wenn Sie z. B. den Zeitpunkt der letzten Zurücksetzung für die Serviceunterklasse `SYSDEFAULTSUBCLASS` unter der Servicesuperklasse `SYSDEFAULTUSERCLASS` sehen wollen, könnten Sie eine Abfrage wie die folgende absetzen:

```
SELECT LAST_RESET
FROM TABLE(WLM_GET_SERVICE_SUBCLASS_STATS( 'SYSDEFAULTUSERCLASS',
'SYSDEFAULTSUBCLASS', -2)) AS T
```

Alle Tabellenfunktionen für Statistikdaten geben die Statistikdaten zurück, die seit dem letzten Zurücksetzen der Statistikdaten aufgelaufen sind. Eine Zurücksetzung der Statistikdaten erfolgt, wenn eine Datenbank aktiviert oder reaktiviert wird, wenn Sie ein DB2-Workload-Management-Objekt ändern (nur die Statistikdaten für dieses Objekt werden zurückgesetzt) und wenn Sie die gespeicherte Prozedur `WLM_COLLECT_STATS` aufrufen. Statistikdaten werden auch automatisch entspre-

chend der vom Datenbankkonfigurationsparameter `wlm_collect_int` definierten Zeitspanne zurückgesetzt, wenn Sie diesen Parameter auf einen Wert ungleich null einstellen.

Der von `wlm_collect_int` definierte Zeitraum wird nicht beeinflusst von einer Statistikrücksetzung, die in dem vom Konfigurationsparameter festgelegten Intervall erfolgt. Wenn Sie z. B. die Tabellenfunktion `WLM_COLLECT_STATS` 5 Minuten nach dem Start eines von `wlm_collect_int` definierten 20-minütigen Intervalls ausführen, läuft das Intervall dennoch 15 Minuten später ab. Die erfolgte Statistikerfassung und -zurücksetzung verzögert nicht die nächste Statistikerfassung und -zurücksetzung um 5 Minuten.

Eine Änderung der Angabe der Histogrammschablone oder die Änderung einer Histogrammschablone für eine Serviceklasse oder eine Workload wird erst nach der Zurücksetzung der Statistikdaten wirksam.

Wenn Sie die Tabellenfunktion `WLM_COLLECT_STATS` aufrufen, um Statistikdaten zu erfassen und zurückzusetzen, während eine andere Erfassung und Zurücksetzung abläuft (z. B. wenn der Aufruf der Tabellenfunktion das periodische Erfassungs- und Zurücksetzungsintervall von `wlm_collect_int` überlappt oder wenn ein anderer Benutzer gleichzeitig `WLM_COLLECT_STATS` aufruft), wird die Erfassungs- und Zurücksetzungsanforderung von `WLM_COLLECT_STATS` ignoriert und die Warnung `SQL1632W` wird zurückgegeben.

Überwachungsmessdaten für DB2-Workload-Management

Die Überwachungsmessdaten stellen Daten zum Zustand und zur Abfrageleistung Ihres DB2-Datenservers bereit, die dann als Eingabe für ein Tool eines anderen Herstellers oder in Verbindung mit zusätzlichem Scripting verwendet werden können, das Sie für die Analyse der zurückgegebenen Messdaten bereitstellen.

Messdaten werden für zahlreiche DB2-Datenbankobjekte verwaltet. Diese Messdaten befinden sich im Hauptspeicher und können in Echtzeit mit Tabellenfunktionen für DB2-Überwachungsmessdaten angezeigt werden. Die Messdaten können auch erfasst und an einen Ereignismonitor gesendet werden, wo sie später für die Protokollanalyse angezeigt werden können.

Überwachungsmessdaten für Aktivitäten

Mit den folgenden Elementen können Sie Überwachungsmessdaten für Aktivitäten erhalten:

- Aktivitätsereignismonitor (Tabelle `ACTIVITYMETRICS` oder Spalte `DETAILS_XML` der Tabelle `ACTIVITY`)
- Tabellenfunktion `MON_GET_ACTIVITY_DETAILS`

Überwachungsmessdaten für Aktivitäten werden vom Datenbankkonfigurationsparameter `mon_act_metrics` sowie der Klausel `COLLECT ACTIVITY METRICS` für eine Workload gesteuert. Messdaten werden für eine Aktivität erfasst, wenn für den Datenbankkonfigurationsparameter ein anderer Wert als `NONE` festgelegt ist oder wenn die Aktivität von einer Verbindung übergeben wird, die einer Workload zugeordnet ist, für deren `COLLECT ACTIVITY METRICS`-Einstellung ein anderer Wert als `NONE` festgelegt ist.

Mit Steuerelementen auf Workloadenebene können Sie eine bessere Überwachungsgranularität erzielen, wenn Sie keine Messdaten für alle Aktivitäten erfassen wol-

len. Wenn die Erfassung von Aktivitätsmessdaten auf Datenbankebene aktiviert ist (dies ist die Standardeinstellung), werden Messdaten für alle Aktivitäten erfasst, unabhängig von der Einstellung auf Workloadebene.

Weitere Details finden Sie in der Dokumentation zur Überwachung.

Überwachungsmessdaten auf Systemebene

Mit den folgenden Elementen können Sie Überwachungsmessdaten auf Systemebene erhalten, zusammengefasst nach Serviceklassen und Workloadobjekten:

- Der Statistikereignismonitor (die Monitorelemente `details_xml` und `metrics` in den logischen Datengruppen `EVENT_SCSTATS` und `EVENT_WLSTATS` oder einzelne Monitorelemente in den logischen Datengruppen `EVENT_SCMETRICS` und `EVENT_WLMETRICS`)
- Tabellenfunktionen `MON_GET_SERVICE_SUBCLASS`, `MON_GET_SERVICE_SUBCLASS_DETAILS`, `MON_GET_WORKLOAD` und `MON_GET_WORKLOAD_DETAILS`

Überwachungsmessdaten für Anforderungen an den Datenserver, einschließlich der Anforderungen, die Teil einer Aktivität sind, werden vom Datenbankkonfigurationsparameter `mon_req_metrics` und von der Klausel `COLLECT REQUEST METRICS` für eine Servicesuperklasse gesteuert. Messdaten werden für eine Anforderung erfasst, wenn für den Datenbankkonfigurationsparameter ein anderer Wert als `NONE` festgelegt ist oder wenn die Anforderung von einer Verbindung übergeben wird, die einer Unterklasse unter einer Superklasse zugeordnet ist, für deren `COLLECT REQUEST METRICS`-Einstellung ein anderer Wert als `NONE` festgelegt ist.

Mit Steuerelementen auf Servicesuperklassenebene können Sie eine bessere Überwachungsgranularität erzielen, wenn Sie keine Messdaten für alle Anforderungen erfassen wollen. Wenn die Erfassung von Anforderungsmessdaten auf Datenbankebene aktiviert ist (dies ist die Standardeinstellung), werden Messdaten für alle Anforderungen erfasst, unabhängig von der Einstellung auf Servicesuperklassenebene.

Weitere Details finden Sie in der Dokumentation zur Überwachung.

Integration von Workload-Management-Tabellenfunktionen und Snapshot Monitor

Sie können bei der Fehlerermittlung oder Leistungsoptimierung DB2-Workload-Management-Tabellenfunktionen in Verbindung mit den Tabellenfunktionen von Snapshot Monitor verwenden.

Die DB2-Workload-Management-Tabellenfunktionen und die Tabellenfunktionen von Snapshot Monitor nutzen die folgenden Felder gemeinsam. Sie können Joins auf diesen Feldern ausführen, um Daten abzuleiten, die Sie für Aktivitäten zur Leistungsdiagnose und Leistungsoptimierung benötigen. Beachten Sie, dass die WLM-Tabellenfunktionen im Gegensatz zu den Tabellenfunktionen für Momentaufnahmen ihre Informationen nicht vom Snapshot Monitor beziehen, sodass die in den WLM-Tabellenfunktionen verfügbaren Informationen nicht mit Snapshot Monitor verfügbar sind.

Tabelle 61. Von Tabellenfunktionen des DB2-Workload-Managements und von Snapshot Monitor gemeinsam genutzte Felder

Feld der Workload-Management-Tabellenfunktion	Feld der Tabellenfunktion von Snapshot Monitor
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

Nehmen wir als Grund für die Verwendung eines Joins zwischen unterschiedlichen Tabellenfunktionen an, dass Sie Basisinformationen zu allen Dienstprogrammen in der Servicesuperklasse BATCH erhalten wollen. Sie können die folgende Abfrage absetzen:

```
SELECT SUBSTR(UTILITY_TYPE,1,4) TYPE,
       UTILITY_PRIORITY PRIORITY,
       SUBSTR(UTILITY_DESCRIPTION,1,12) DESCRIPTION,
       SUBSTR(UTILITY_DBNAME,1,8) DBNAME,
       UTILITY_STATE 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
```

Die Ausgabe könnte folgendermaßen aussehen:

TYPE	PRIORITY	DESCRIPTION	DBNAME	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		

Überwachen von Schwellenwertverstößen

Bei einem Verstoß gegen einen DB2-Workload-Manager-Schwellenwert wird ein Schwellenwertverstoßeintrag in den aktiven THRESHOLD VIOLATIONS-Ereignismonitor (sofern vorhanden) geschrieben.

Informationen zu diesem Vorgang

Der Schwellenwertverstoßeintrag umfasst die folgenden Informationen:

- Eine Beschreibung des Schwellenwerts, gegen den verstoßen wurde (ID, Maximalwert usw.).

- Eine ID der Aktivität, die den Schwellenwertverstoß verursachte, einschließlich der ID der Anwendung, die die Aktivität übergab, der eindeutigen Aktivitäts-ID und der UOW-ID.
- Den Zeitpunkt des Schwellenwertverstoßes.
- Die durchgeführte Aktion. Die Aktion gibt an, ob die Aktivität, die den Schwellenwertverstoß verursachte, weiter ausgeführt werden durfte oder gestoppt wurde. Wenn die Aktivität gestoppt wurde, hat die Anwendung, die dies Aktivität übergeben hat, einen Fehler SQL4712N empfangen.

Wenn ein Schwellenwertverstoß für einen Schwellenwert auftritt, für den eine Aktion REMAP ACTIVITY definiert ist, ist ein Schwellenwertverstoßeintrag optional. Ob ein Schwellenwertverstoßeintrag aufgezeichnet wird, wird von der Klausel NO EVENT MONITOR RECORD oder LOG EVENT MONITOR RECORD Ihrer Anweisung CREATE THRESHOLD festgelegt.

Sie können optional detaillierte Aktivitätsinformationen (einschließlich Anweisungstext) in einen aktiven Aktivitätsereignismonitor schreiben lassen, wenn der Schwellenwertverstoß von einer Aktivität verursacht wurde. Die Aktivitätsinformationen werden nach Abschluss der Aktivität und nicht zum Zeitpunkt des Schwellenwertverstoßes geschrieben. Geben Sie an, dass bei einem Schwellenwertverstoß Aktivitätsinformationen erfasst werden sollen, indem Sie das Schlüsselwort COLLECT ACTIVITY DATA in der Anweisung zum Erstellen oder Ändern eines Schwellenwerts oder eines Arbeitsaktionsset verwenden.

Vorgehensweise

Gehen Sie wie folgt vor, um Schwellenwertverstöße zu überwachen:

1. Erstellen Sie mit der Anweisung CREATE EVENT MONITOR einen Ereignismonitor vom Typ THRESHOLD VIOLATIONS. Beispiel:
CREATE EVENT MONITOR VIOLATIONS FOR THRESHOLD VIOLATIONS WRITE TO TABLE
2. Verwenden Sie die Anweisung COMMIT, um die Änderungen festzuschreiben.
3. Aktivieren Sie den Ereignismonitor mithilfe der Anweisung SET EVENT MONITOR STATE. Statt der Anweisung SET EVENT MONITOR STATE können Sie auch die Standardeinstellung AUTOSTART für den THRESHOLD VIOLATIONS-Ereignismonitor verwenden, um ihn bei der nächsten Aktivierung der Datenbank aktivieren zu lassen. Wenn Sie mehrere THRESHOLD VIOLATIONS-Ereignismonitore definieren wollen, sollten Sie nicht die Option AUTOSTART verwenden.
4. Verwenden Sie die Anweisung COMMIT, um die Änderungen festzuschreiben.

Anmerkung: Wenn Sie Schwellenwerte erstellen, sollten Sie einen Ereignismonitor für Schwellenwertverstöße erstellen und aktivieren, sodass Sie alle Schwellenwertverstöße überwachen können. Ein Ereignismonitor für Schwellenwertverstöße hat keine Auswirkung, sofern nicht gegen Schwellenwerte verstoßen wird.

Beispiel

Dieses Beispiel zeigt, wie Sie ermitteln können, welche Neuuzuordnungen einer bestimmten Aktivität als Ergebnis eines Schwellenwertverstoßes aufgetreten sind, der eine Aktion REMAP ACTIVITY enthielt. Verwenden Sie eine Anweisung ähnlich der folgenden, um Aktivitäten zu ermitteln, die neu zugeordnet wurden:

```

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;

```

In diesem Beispiel erfolgten zwei Neuuzuordnungen für die Aktivität, die von der Anwendung mit der ID *N0.swalkty.080613140844 übergeben wurde und die durch die Aktivitäts-ID 1 und die UOW-ID 1 angegeben wird:

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 record(s) selected.

Die Ausgabe ist nach dem Zeitpunkt des Schwellenwertverstoßes geordnet und zeigt, dass die Aktivität nach Beginn ihrer Ausführung zweimal neu zugeordnet wurde. Obwohl dies nicht in der Ausgabe zu sehen ist, ist die anfängliche Serviceunterklasse, der die Aktivität zugeordnet wurde, wahrscheinlich eine Serviceunterklasse mit hoher Priorität. Dies ist typisch für eine dreischichtige Konfiguration, die die schnellere Ausführung kürzer laufender Abfragen erlaubt. Da die Aktivität in der Serviceunterklasse mit hoher Priorität nicht schnell genug abgeschlossen werden konnte, verstieß sie gegen einen Schwellenwert und wurde einer Serviceunterklasse mit mittlerer Priorität neu zugeordnet. Später, nach einem zweiten Schwellenwertverstoß, wurde sie noch einmal neu zugeordnet, und zwar einer Serviceunterklasse mit niedriger Priorität.

Generieren von E-Mail-Benachrichtigungen für Schwellenwertverstöße

Mit der hier beschriebenen Methode können Sie bei einem Verstoß gegen einen DB2-Workload-Manager-Schwellenwert (DB2-WLM-Schwellenwert) E-Mail-Benachrichtigungen generieren.

Vorbereitende Schritte

Für die Implementierung dieses E-Mail-Benachrichtigungskonzepts muss DB2 Version 9.7 oder höher installiert sein. Die hier verwendete SMTP-Unterstützung wird ab DB2 Version 9.7 bereitgestellt.

Informationen zu diesem Vorgang

Nach Abschluss dieser Task werden E-Mail-Benachrichtigungen gesendet, wenn innerhalb von 10 Minuten seit der letzten Ausführung der Prozedur für Schwellenwertbenachrichtigungen Verstöße gegen einen WLM-Schwellenwert auftreten. In diesem Beispiel wird die Prozedur für Schwellenwertbenachrichtigungen mithilfe von DB2 Administrative Task Scheduler so terminiert, dass sie alle 10 Minuten ausgeführt wird.

Vorgehensweise

1. Aktualisieren Sie den Datenbankkonfigurationsparameter **smtp_server** mit dem folgenden Befehl:

```
UPDATE DB CONFIG USING SMTP_SERVER smtp-servername
```

2. Setzen Sie die folgende Anweisung ab, um einen WRITE-TO-TABLE-Ereignismonitor für Schwellenwertverstöße zu erstellen und Verstöße in die Tabelle TEST.THRESHOLDVIOLATIONS_T zu schreiben:

```
CREATE EVENT MONITOR T FOR THRESHOLD VIOLATIONS WRITE TO TABLE  
THRESHOLDVIOLATIONS( TABLE TEST.THRESHOLDVIOLATIONS_T )
```

3. Aktivieren Sie den WRITE-TO-TABLE-Ereignismonitor T für Schwellenwertverstöße, indem Sie die folgende Anweisung absetzen:

```
SET EVENT MONITOR T STATE 1
```

4. Erstellen Sie eine Steuertabelle, um den letzten Schwellenwert zu verfolgen, für den ein Alert generiert wurde, indem Sie die folgende Anweisung absetzen:

```
CREATE TABLE TEST.THRESHOLD_NOTIFY_CONTROL( LAST_NOTIFICATION TIMESTAMP )
```

5. Erstellen Sie eine gespeicherte Prozedur für Schwellenwertbenachrichtigungen, um Benachrichtigungen bei Schwellenwertverstößen zu generieren. Die folgende Beispielprozedur erstellt auf der Grundlage der Tabelle zu Schwellenwertverstößen einen Bericht, der alle Schwellenwertverstöße auflistet, die seit dem letzten Aufrufen der Prozedur aufgetreten sind. Der Bericht wird mithilfe der DB2-SMTP-Prozeduren als E-Mail-Benachrichtigung gesendet.

```
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 = '<e-mail-senderadresse>';
    SET RECIPIENTS = '<e-mail-empfängeradresse>';
    SET SUBJECT = 'Neue WLM-Schwellenwertverstöße' ;

    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 || 'Zeitmarke des Verstoßes           = ' ||
            TIME_OF_VIOLATION || CHAR(X'0A');
        SET MESSAGE = MESSAGE || 'Schwellenwertvergleichselement = ' ||
            THRESHOLD_PREDICATE || CHAR(X'0A');
        SET MESSAGE = MESSAGE || 'Schwellenwert-ID           = ' ||
            THRESHOLDID || CHAR(X'0A');
        SET MESSAGE = MESSAGE || 'Anwendungs-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. Aktivieren Sie DB2 Administrative Task Scheduler, indem Sie den folgenden Befehl ausführen:

```
db2set DB2_ATS_ENABLE=YES
```

7. Terminieren Sie die Prozedur für Schwellenwertbenachrichtigungen so, dass sie alle 10 Minuten ausgeführt wird. Zum Terminieren der Prozedur müssen Sie über das Zugriffsrecht EXECUTE für die Prozedur verfügen. Es folgt ein Beispiel für die Vorgehensweise:

```

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 )@

```

Ergebnisse

Bei einem WLM-Schwellenwertverstoß wird (mit einer Latenzzeit von maximal 10 Minuten) eine E-Mail gesendet. In der E-Mail werden alle WLM-Schwellenwertverstoße seit der letzten Ausführung der Prozedur für Schwellenwertverstoße beschrieben, die so terminiert ist, dass sie alle 10 Minuten ausgeführt wird.

Beispiel

Die folgende Beispielausgabe stellt den Inhalt einer E-Mail-Benachrichtigung dar, die die aufgelaufenen neuen Schwellenwertverstoße seit der letzten Ausführung der Prozedur für Schwellenwertverstoße zeigt:

Subject: Neue WLM-Schwellenwertverstoße

```
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
```

Erfassen von Daten für einzelne Aktivitäten

Sie können mit einem ACTIVITIES-Ereignismonitor Daten für einzelne Aktivitäten erfassen, die auf Ihrem System ausgeführt werden. Die erfassten Daten umfassen Elemente wie Anweisungstext und Kompilierungsumgebung und können zur Untersuchung und Diagnose von Problemen sowie zur Eingabe in andere Tools (z. B. Designadvisor) verwendet werden.

Informationen zu diesem Vorgang

Sie können Informationen zu einzelnen Aktivitäten für Serviceunterklassen, Workloads, Arbeitsklassen (über Arbeitsaktionen) und Schwellenwertverstoße erfassen. Sie aktivieren die Aktivitätserfassung mit dem Schlüsselwort COLLECT ACTIVITY DATA der Anweisungen CREATE und ALTER für diese DB2-Workload-Management-Objekte. Beim Abschluss einer Aktivität werden Informationen zur Aktivität an den aktiven ACTIVITIES-Ereignismonitor gesendet, wenn folgende Bedingungen erfüllt sind:

- Die Aktivität wurde von einer Anwendung übergeben, die einer Workload zugeordnet ist, für die COLLECT ACTIVITY DATA angegeben wurde. Oder:
- Die Aktivität wird in einer Serviceunterklasse ausgeführt, für die COLLECT ACTIVITY DATA angegeben ist. Oder:
- Auf die Aktivität wurde die Arbeitsaktion COLLECT ACTIVITY DATA angewendet. Oder:
- Die Aktivität verstößt gegen einen Schwellenwert, der mit der Aktion COLLECT ACTIVITY DATA definiert wurde.

Mit der Prozedur WLM_SET_CONN_ENV können Sie die Aktivitätserfassung für die Verbindung Ihrer eigenen Anwendung aktivieren, bevor Sie die Abfrage des

Benutzers ausführen. Anschließend können Sie die Abfrage des Benutzers ausführen und dann die Aktivitätserfassung für die Verbindung Ihrer eigenen Anwendung mit `WLM_SET_CONN_ENV` inaktivieren. Wenn Sie einen Aktivitätsereignismonitor erstellt und aktiviert haben, könnte die Anwendung folgendermaßen aussehen:

```
call WLM_SET_CONN_ENV(cast (NULL as bigint), '<collectactdata>WITHOUT DETAILS</collectactdata>')
```

... Abfrage des Benutzers ausführen ...

```
call WLM_SET_CONN_ENV(cast(NULL as bigint), '<collectactdata>NONE</collectactdata>')
```

Das Schlüsselwort `COLLECT ACTIVITY DATA` steuert auch den Umfang der Informationen, die an den `ACTIVITIES`-Ereignismonitor gesendet werden. Wenn das Schlüsselwort die Option `WITH DETAILS` angibt, werden Anweisungsinformationen (wie der Anweisungstext) erfasst. Wenn das Schlüsselwort die Option `WITH DETAILS AND VALUES` angibt, werden auch Datenwerte erfasst.

Auf eine Aktivität werden eventuell mehrere Schlüsselwörter `COLLECT ACTIVITY DATA` angewendet. Beispielsweise wird die Aktivität in einer Serviceunterklasse ausgeführt, für die `COLLECT ACTIVITY DATA` angegeben ist, und verstößt während der Ausführung möglicherweise gegen einen Schwellenwert mit der Aktion `COLLECT ACTIVITY DATA`. In dieser Situation wird die Aktivität nur einmal erfasst. Das Schlüsselwort `COLLECT`, das die größte zu erfassende Datenmenge angibt, wird auf die Aktivität angewendet. Wenn z. B. sowohl `COLLECT ACTIVITY DATA WITHOUT DETAILS` und `COLLECT ACTIVITY DATA WITH DETAILS` auf eine Aktivität angewendet werden, wird die Aktivität mit detaillierten Informationen erfasst.

Werden die Schlüsselwörter `ON ALL DATABASE MEMBERS` mit der Klausel `COLLECT ACTIVITY DATA` verwendet, wird für jedes Member, für das die Aktivität in einer Datenbankumgebung mit mehreren Members ausgeführt wird, ein Aktivitätsdatensatz erfasst. Datensätze des Aktivitätsereignismonitors werden geschrieben, wenn der letzte Agent, der die Aktivität bearbeitet, die Ausführung für das entsprechende Member beendet. Abhängig von der Planung der Ereignisreihenfolge in einem Abschnitt ist es möglich, dass Agenten die Arbeit für eine Aktivität für ein Member mehrmals starten und stoppen, sodass mehrere Aktivitätsdatensätze für dieselbe Abfrage für das Member erfasst werden. Die für die Aktivität für das Member ausgeführte Gesamtarbeit wird durch Zusammenfassen der Messdaten der einzelnen Datensätze berechnet, die für das Member für die Aktivität erfasst werden.

Vorgehensweise

Gehen Sie wie folgt vor, um die Erfassung von Aktivitäten für ein bestimmtes DB2-Workload-Management-Objekt zu aktivieren:

1. Erstellen Sie mit der Anweisung `CREATE EVENT MONITOR` einen `ACTIVITIES`-Ereignismonitor.
2. Verwenden Sie die Anweisung `COMMIT`, um die Änderungen festzuschreiben.
3. Aktivieren Sie den Ereignismonitor mithilfe der Anweisung `SET EVENT MONITOR STATE`. Statt der Anweisung `SET EVENT MONITOR STATE` können Sie auch die Standardeinstellung `AUTOSTART` für den `ACTIVITIES`-Ereignismonitor verwenden, um ihn bei der nächsten Aktivierung der Datenbank aktivieren zu lassen. Wenn Sie mehrere `ACTIVITIES`-Ereignismonitore definieren wollen, sollten Sie nicht die Option `AUTOSTART` verwenden.
4. Verwenden Sie die Anweisung `COMMIT`, um die Änderungen festzuschreiben.

5. Definieren Sie die Objekte, für die Sie Aktivitäten erfassen wollen, mit der Anweisung ALTER SERVICE CLASS, ALTER WORK ACTION SET, ALTER THRESHOLD oder ALTER WORKLOAD, und geben Sie das Schlüsselwort COLLECT ACTIVITY DATA an.
6. Verwenden Sie die Anweisung COMMIT, um die Änderungen festzuschreiben.

Ergebnisse

Anmerkung: Die Erfassung einzelner Aktivitäten ist aufwendiger als die Erfassung von Workload-Management-Statistikdaten. Sie sollten versuchen, eine Aktivitätserfassung einzurichten, die so wenig Aktivitäten wie möglich erfasst. Wenn Sie z. B. Aktivitäten untersuchen müssen, die von einer bestimmten Anwendung übergeben wurden, sollten Sie diese Anwendung eingrenzen, indem Sie eine Workload oder eine Serviceklasse speziell für diese Anwendung erstellen und die Aktivitätserfassung nur für diese Workload oder Serviceklasse aktivieren.

Sie wissen vielleicht nicht immer im voraus, dass Sie eine Aktivität erfassen wollen. Beispielsweise haben Sie eine Abfrage, deren Ausführung sehr lange dauert, und Sie wollen Informationen zu ihr für die spätere Analyse erfassen. In dieser Situation ist es zu spät, das Schlüsselwort COLLECT ACTIVITY DATA für die DB2-Workload-Management-Objekte anzugeben, da die Aktivität bereits in das System eingegangen ist. In diesem Fall können Sie die gespeicherte Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS verwenden. Die gespeicherte Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS sendet Informationen zu einer zurzeit ausgeführten Aktivität an den aktiven ACTIVITIES-Ereignismonitor. Sie geben die zu erfassende Aktivität durch die Anwendungskennung, die UOW-ID und die Aktivitäts-ID an. Informationen zur Aktivität werden sofort beim Aufruf der Prozedur an den ACTIVITIES-Ereignismonitor gesendet; Sie müssen nicht auf den Abschluss der Aktivität warten.

Importieren von Aktivitäten in Designadvisor

Sie können von einem Aktivitätsereignismonitor erfasste Aktivitäten in den Designadvisor importieren, um leichter Entscheidungen über die Datenbankobjekte treffen zu können, auf die diese Aktivitäten zugreifen.

Informationen zu diesem Vorgang

In den Designadvisor importierte Aktivitäten müssen mit den Optionen COLLECT ACTIVITY DATA WITH DETAILS oder COLLECT ACTIVITY DATA WITH DETAILS AND VALUES erfasst worden sein. Die Option COLLECT ACTIVITY DATA WITHOUT DETAILS ist nicht ausreichend. Sie erfasst nicht den Anweisungstext, der für den Designadvisor erforderlich ist.

Führen Sie für den Import von Aktivitätsinformationen aus den Aktivitätsereignismonitortabellen in den Designadvisor den Befehl **db2adv** mit dem Parameter **-wlm** gefolgt von weiteren Parametern aus:

1. Name des Aktivitätsereignismonitors
2. Optional: Name der Workload oder der Serviceklasse
3. Optional: Startzeit und Endzeit

Wenn Sie z. B. Informationen zu allen Aktivitäten importieren wollen, die vom Ereignismonitor DB2ACTIVITIES in der Datenbank SAMPLE erfasst wurden, verwenden Sie den folgenden Befehl:

```
db2adv -d SAMPLE -wlm DB2ACTIVITIES
```

Anmerkung: Sie können Informationen von Aktivitätsereignismonitortabellen nur über die Befehlszeilenschnittstelle des Designadvisors importieren.

Abbrechen von Aktivitäten

Wenn eine Aktivität zu viele Ressourcen verbraucht oder ihre Ausführung zu lange dauert, können Sie sie abbrechen. Das Abbrechen einer Aktivität ist schonender als das erzwungene Beenden der Anwendung, die die Aktivität übergeben hat. Eine abgebrochene Aktivität gibt SQL4725N an den Benutzer zurück, beendet jedoch nicht die Verbindung und wirkt sich nicht auf andere Benutzeraktivitäten aus. Durch das erzwungene Beenden der Anwendung werden sowohl die Verbindung als auch die Benutzeraktivitäten beendet.

Informationen zu diesem Vorgang

Sie können eine Aktivität nur dann explizit abbrechen, wenn eine Koordinatoraktivität zurzeit an einer Anforderung für die Aktivität arbeitet. Wenn Sie eine Aktivität abbrechen, die sich im Leerlauf befindet (d. h., für die keine Anforderungen verarbeitet werden), erhält die Aktivität den Status CANCEL_PENDING und wird beim Empfang der nächsten Anforderung abgebrochen. Wenn Sie z. B. versuchen, eine Cursoraktivität zwischen Abrufen abzubrechen, wird der Fehler SQL4725N erst beim nächsten Abruf nach der Abbruchanforderung an den Benutzer zurückgegeben.

Alle Benutzeraktivitäten können abgebrochen werden, auch das Dienstprogramm LOAD und gespeicherte Prozeduren.

Vorgehensweise

1. Geben Sie die Aktivität an, die Sie abbrechen wollen. Mit der Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES können Sie die Aktivitäten ermitteln, die in einer Anwendung ausgeführt werden. Sie können auch die Tabellenfunktion MON_GET_ACTIVITY_DETAILS_COMPLETE verwenden, um zusätzliche Details zu einer bestimmten Aktivität zu erhalten, wenn die Informationen in WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES nicht ausreichen, um die von den Aktivitäten ausgeführte Arbeit zu ermitteln.
2. Brechen Sie die Aktivität mithilfe der gespeicherten Prozedur WLM_CANCEL_ACTIVITY ab. Die gespeicherte Prozedur verwendet die folgenden Argumente: *anwendungskennung*, *uow-id* und *aktivitäts-id*. Ein Beispiel für die Verwendung dieser gespeicherten Prozedur finden Sie in „Szenario: Ermitteln von Aktivitäten, deren Ausführung zu lange dauert“ auf Seite 376.

Richtlinien für das Erfassen von Informationen zu einer Rogue-Aktivität und Untersuchen dieser Aktivität

Dieser Abschnitt enthält Richtlinien zur Erfassung von Informationen und zur Untersuchung einer Rogue-Aktivität.

Erstellen Sie zuerst eine Gruppe von Kriterien dafür, was Sie als Rogue-Aktivität betrachten würden. Beispiel:

- Eine Aktivität, die in einer Serviceklasse für Aktivitäten mit einem geringen geschätzten Aufwand ausgeführt wird und deren Ausführung länger als 1 Stunde dauert
- Eine Aktivität, die ungewöhnlich viele Zeilen zurückgibt

- Eine Aktivität, die einen sehr großen Tabellenbereich für temporäre Tabellen verbraucht

Erstellen Sie dann Schwellenwerte, die diese Kriterien beschreiben und eine Aktion COLLECT ACTIVITY DATA WITH DETAILS enthalten. Bei einem Schwellenwertverstoß werden Informationen zu der Aktivität, die den Schwellenwertverstoß verursachte, bei Abschluss der Aktivität an den aktiven ACTIVITIES-Ereignismonitor gesendet.

Wenn Sie z. B. Informationen zu Datenbankaktivitäten erfassen wollen, deren Ausführung länger als 3 Stunden dauert, erstellen Sie einen Schwellenwert wie den folgenden:

```
CREATE THRESHOLD LONGRUNNINGACTIVITIES FOR DATABASE ACTIVITIES ENFORCEMENT DATABASE
  WHEN ACTIVITYTOTALTIME > 3 HOURS COLLECT ACTIVITY DATA WITH DETAILS
  CONTINUE
```

Die Überwachung mit dem DB2-Workload-Manager ist unaufwendig, wenn Sie darauf achten, sie nur auf eine kleine Teilmenge Ihrer Abfragen anzuwenden, wie im Beispiel gezeigt, in dem nur Abfragen überwacht werden, die mindestens drei Stunden laufen. Sie können dieses Beispiel weiter eingrenzen, indem Sie einen Schwellenwert nicht auf der globalen Datenbankebene sondern auf der Ebene einer benutzerdefinierten Superklasse erstellen. Wenn diese enger umgrenzte Überwachung für Ihre Zwecke ausreicht, kann sie den Überwachungsaufwand weiter verringern und nur Informationen zur benötigten Ebene bereitstellen:

```
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
```

. Der für den Schwellenwert erstellten Serviceklasse wird eine niedrige Agenten- und Vorabsezugriffspriorität zugewiesen, da sie für lange laufende Abfragen vorgesehen ist (diese SQL-Anweisung funktioniert unter den Betriebssystemen UNIX und Linux. Verwenden Sie unter Windows stattdessen eine Agentenpriorität von -6).

Nachdem Ihr Datenserver einen Teil der Arbeit ausgeführt hat, können Sie die Informationen analysieren, die in den Ereignismonitor für Schwellenwertverstöße und in den Aktivitätsereignismonitor geschrieben wurden. Für DML-Aktivitäten können auch der Anweisungstext sowie Informationen zur Kompilierungsumgebung in den Aktivitätsereignismonitor geschrieben werden, sodass Sie darauf **DB2 explain** ausführen können, um die Leistung der Aktivität weiter zu untersuchen.

Modellieren der Workload-Management-Leistung

Die Auslastung auf Ihrem System kann als Menge von Aktivitäten modelliert werden, die auf dem System mit einer Geschwindigkeit eingehen, die durch eine Eingangsratenverteilung für Aktivitäten (oft gemessen als Kehrwert, die Verteilung der *Zwischenankunftszeit* (Inter-Arrival Time)) und der Zeitdauer gesteuert wird, während der die Aktivitäten zur Ausführung auf dem System verbleiben, die einer Servicezeitverteilung folgt.

Die Zwischenankunftszeit ist die Zeit zwischen dem Eingang einer Aktivität und dem Eingang der nächsten Aktivität. Die Servicezeit ist die Zeit, während der eine

Aktivität zur Ausführung auf dem System verbleibt. Wenn Sie z. B. eine Abfrage zum Zeitpunkt 0 Sekunden übergeben, diese Abfrage 2 Sekunden in einer Warteschlange verbleibt und zum Zeitpunkt 5 Sekunden abgeschlossen wird, beträgt die Servicezeit $5 - 2 = 3$ Sekunden. Bei der Servicezeit wird davon ausgegangen, dass keine andere Arbeit auf dem System ausgeführt wird (d. h., sie ist nicht die beobachtete Ausführungszeit, sondern vielmehr die Zeit, die benötigt würde, um die Aktivität isoliert auszuführen). Die Servicezeitverteilung kann für DML-Aktivitäten näherungsweise berechnet werden anhand des geschätzten Aufwands in Timeron, bei dem sowohl die Prozessor- als auch die E/A-Zeit für eine Aktivität berücksichtigt wird.

Sie können ein Workloadmodell für Ihr System durch Messung der Zwischenankunftszeitverteilung und der Servicezeitverteilung der Aktivitäten auf dem System erstellen. Sie erhalten die Zwischenankunftszeitverteilungen und die (anhand des geschätzten Aufwands) näherungsweise berechneten Servicezeitverteilungen durch erweiterte zusammengefasste Aktivitätsstatistiken für Serviceunterklassen oder Arbeitsklassen (über Arbeitsaktionen) und einen Statistikereignismonitor. Diese Statistikdaten werden standardmäßig nicht erfasst. Weitere Informationen finden Sie in:

- „Statistikdaten für DB2-Workload-Management-Objekte“ auf Seite 264
- A gentle introduction to histograms
- Understanding the six histograms of DB2 workload management
- Visualizing and deriving statistics from DB2 histograms using SQL

Beispiel: Erfassen von Informationen zu einer Aktivität für spätere Analyse

Mit den Workload-Management-Features können Sie Informationen zu einer Aktivität zur späteren Analyse erfassen.

Nehmen wir an, Sie haben eine gespeicherte Prozedur namens `MYSHEMA.MYSLOWSTP`, und diese wird langsamer als gewöhnlich ausgeführt. Sie erhalten allmählich Klagen über diese Situation und beschließen, die Ursache der Leistungsminderung zu untersuchen. Wenn die Untersuchung während der Ausführung der gespeicherten Prozedur nicht praktikabel ist, können Sie Informationen zur Aktivität der gespeicherten Prozedur und aller darin verschachtelter Aktivitäten erfassen.

Nehmen wir an, dass Sie einen aktiven Aktivitätsereignismonitor namens `DB2ACTIVITIES` haben. Sie können dann eine Arbeitsklasse für `CALL`-Anweisungen erstellen, die auf das Schema der gespeicherten Prozedur `MYSHEMA.MYSLOWSTP` angewendet werden. Dann können Sie eine Arbeitsaktion erstellen, um die `CALL`-Aktivität und alle verschachtelten Aktivitäten einer Serviceklasse zuzuordnen, für die die Erfassung von Aktivitäten aktiviert ist. Die `CALL`-Aktivität und alle darin verschachtelten Aktivitäten werden an den Ereignismonitor gesendet. Im Folgenden sehen Sie Beispiele für den DDL-Code, der zum Erstellen der DB2-Workload-Management-Objekte erforderlich ist:

```
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);
```

Nachdem die gespeicherte Prozedur MYSHEMA.MYSLOWSTP ausgeführt wird, können Sie die folgende Abfrage absetzen, um die Anwendungskennung, die UOW-ID und die Aktivitäts-ID für die Aktivität zu ermitteln:

```
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'));
```

Wenn die erfasste Aktivität die Anwendungskennung 1, die UOW-ID 2 und die Aktivitäts-ID 3 hat, werden die folgenden Ergebnisse generiert:

AGENT_ID	UOW_ID	ACTIVITY_ID
1	2	3

Mit diesen Informationen können Sie die folgende Abfrage auf den Tabellen ACTIVITY_DB2ACTIVITIES und ACTIVITYSTMT_DB2ACTIVITIES absetzen, um zu ermitteln, wo die Aktivität ihre Zeit verbracht hat:

```
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;
```

Die Ergebnisse ähneln der folgenden Ausgabe:

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

Die Ergebnisse deuten darauf hin, dass die gespeicherte Prozedur ihre meiste Zeit bei der Abfrage der Tabelle MYHUGETABLE verbringt. Als Nächstes müssen Sie untersuchen, welche Änderungen an der Tabelle MYHUGETABLE für die Verlangsamung der darauf ausgeführten Abfragen verantwortlich sein könnten.

Wenn viele gespeicherte Prozeduren gleichzeitig ausgeführt werden, ergibt sich bei der Durchführung der Analyse ein größerer Systemaufwand. Zur Behebung dieses Problems können Sie eine Workload und eine Serviceklasse für die Ausführung einer gespeicherten Prozedur erstellen, die von einer bestimmten Berechtigungs-ID und/oder einer bestimmten Anwendung abgesetzt wird. Dann können Sie mit der oben beschriebenen Methode das Verhalten der gespeicherten Prozedur analysieren.

Kapitel 5. Integration in Betriebssystem-Workload-Manager

Verwenden Sie, sofern verfügbar, das DB2-Workload-Management in Verbindung mit einem Betriebssystem-Workload-Manager, der weitere Funktionalität zur Verfügung stellt.

Der Integrationspunkt zwischen dem DB2-Workload-Management und Betriebssystem-Workload-Managern ist die DB2-Serviceklasse. Sie erstellen eine Zuordnung zwischen einer DB2-Serviceklasse und einer Betriebssystem-Workload-Manager-Klasse, wenn Sie eine DB2-Serviceklasse über die Option `OUTBOUND CORRELATOR` der Anweisung `CREATE SERVICE CLASS` oder `ALTER SERVICE CLASS` definieren.

Wenn der abgehende Korrelator festgelegt ist, werden alle Threads in der DB2-Serviceklasse mithilfe des abgehenden Korrelators zu Beginn der nächsten Aktivität dem Betriebssystem-Workload-Manager zugeordnet.

Integration von AIX Workload Manager in DB2-Workload-Management

Unter dem Betriebssystem AIX haben Sie aufgrund der optionalen Integration zwischen DB2-Serviceklassen und AIX-WLM-Klassen die Möglichkeit, den Anteil der Prozessorressourcen zu steuern, der jeder Serviceklasse zugeordnet wird.

Die Implementierung von Steuerelementen für AIX-WLM-Klassen ist eventuell nicht erforderlich, um Ihre Leistungsziele zu erreichen. Aber selbst wenn Sie AIX WLM nicht ausführen müssen, sind die von AIX WLM über AIX-Klassen bereitgestellten Betriebssystemstatistikdaten oft nützlich für die Überwachung und die Optimierung.

AIX WLM weist Klassen relative oder absolute Anteile von Prozessorressourcen zu. Diese Klassen profitieren von Steuerelementen, die Sie dynamisch ändern können und die sofort wirksam werden. Wenn relative AIX-CPU-Anteile nicht den erforderlichen Steuerungsgrad bereitstellen, haben Sie auch die Möglichkeit, einen festen maximalen Prozentsatz von CPU-Ressourcen zuzuordnen. Dadurch opfern Sie etwas Flexibilität gegenüber der relativen CPU-Zuordnung, die vor allem bei geringer Systemauslastung vorteilhaft ist, erhalten jedoch eine ausgezeichnete und garantierte Steuerung mit einem festen Maximalwert der Ressourcenzuordnung für die CPU-Zeit.

Empfohlene Zuordnungen zwischen DB2-Serviceklassen und AIX-Klassen

Verwenden Sie eine 1:1-Zuordnung von DB2-Serviceklassen zu Serviceklassen von AIX Workload Manager, um die Vorteile der AIX-WLM-Prozessorsteuerung nutzen zu können. Durch eine 1:1-Zuordnung zwischen DB2-Serviceklassen und Serviceklassen von AIX Workload Manager können Sie die AIX-Prozessorressourcen für jede DB2-Serviceklasse individuell anpassen, um Ihre Geschäftsprioritätsziele zu erreichen.

Die folgende Abbildung zeigt die Integration von DB2-Workload-Management und AIX Workload Manager. Beachten Sie die 1:1-Zuordnung zwischen jeder DB2-Serviceklasse und jeder Serviceklasse von AIX Workload Manager auf der Ebene der Servicesuperklasse und der Serviceunterklasse.

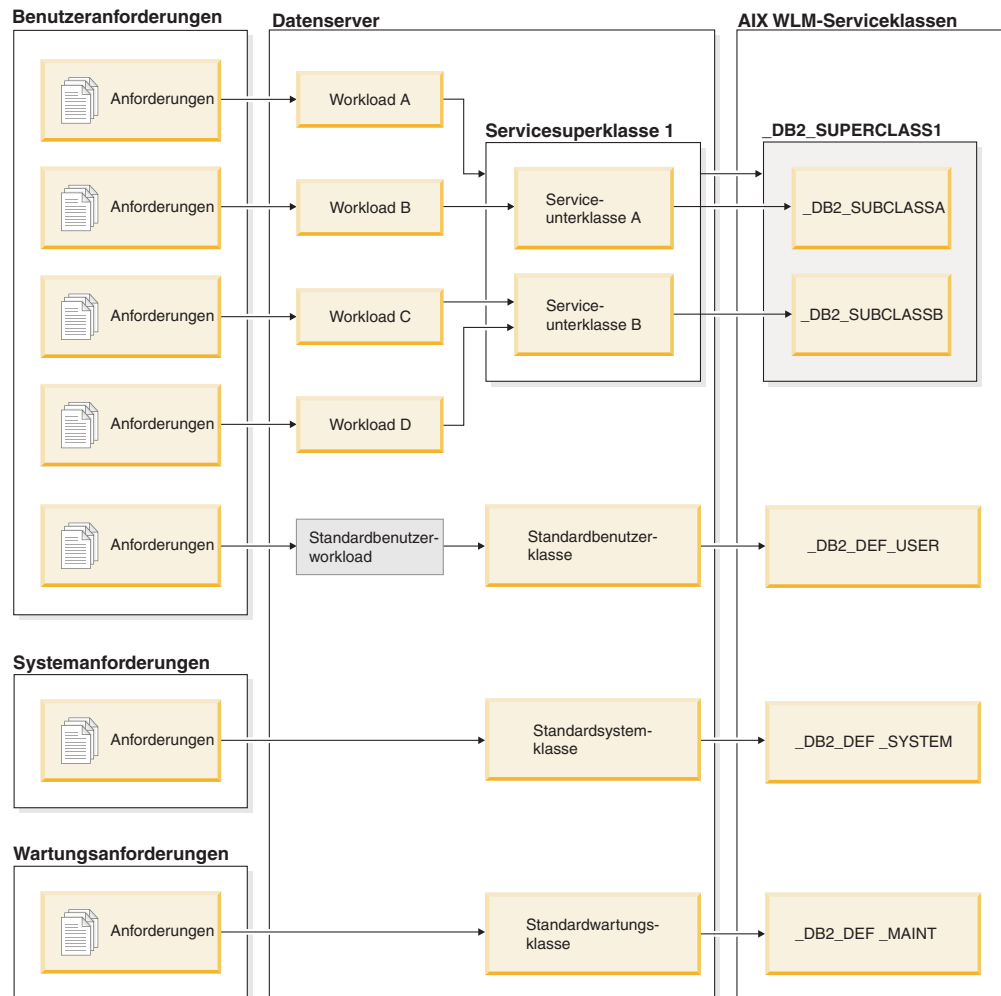


Abbildung 52. Integration von DB2-Workload-Managements und AIX Workload Manager

Wenn eine DB2-Umgebung aus einer einzigen Datenbank in einer einzigen DB2-Instanz besteht, wie im Beispiel in der obigen Abbildung, ist eine direkte Zuordnung zwischen DB2-Serviceklassen und Serviceklassen von AIX Workload Manager möglich. Jede DB2-Servicesuperklasse kann eine entsprechende Servicesuperklasse in AIX Workload Manager haben und jede DB2-Serviceunterklasse kann einer entsprechenden AIX-Serviceunterklasse zugeordnet werden.

In Situationen, in denen die DB2-Umgebung aus mehreren Datenbanken und DB2-Instanzen besteht, können mehrere Ebenen Kandidaten für die Ressourcensteuerung sein. Da AIX Workload Manager eine zweischichtige Hierarchie mit Superklasse und Unterklasse unterstützt, können jeweils nur zwei Ebenen einer DB2-Umgebung den Klassen von AIX Workload Manager zugeordnet werden. Die folgende Abbildung zeigt eine Möglichkeit, eine 1:1-Zuordnung bei mehreren Datenbanken jeweils mit mehreren Superklassen zu erreichen. Hier hat jede Datenbank ihre eigene Superklasse von AIX Workload Manager und jede DB2-Servicesuperklasse wird einer Unterklasse von AIX Workload Manager zugeordnet.

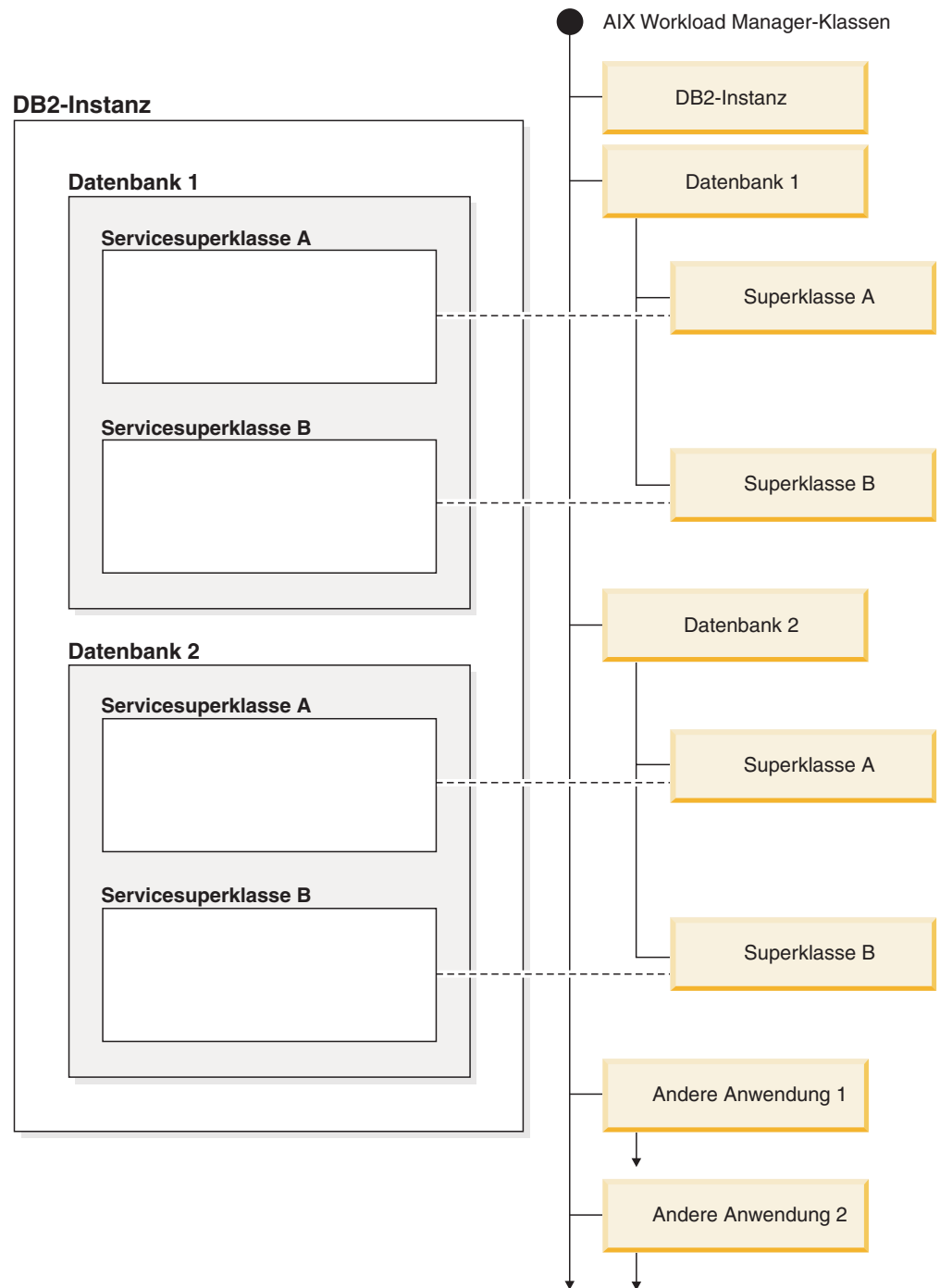


Abbildung 53. DB2-Serviceklassen, die AIX-Klassen zugeordnet sind (nur mit DB2-Servicesuperklassen)

Eine alternative Konfiguration besteht darin, jede DB2-Servicesuperklasse ihrer eigenen Superklasse von AIX Workload Manager zuzuordnen, was in diesem Beispiel zu vier Superklassen führt. In dieser Situation wird die Datenbankebene der Ressourcensteuerung explizit in den Serviceklassendefinitionen von AIX Workload Manager dargestellt.

Die folgende Abbildung zeigt eine Möglichkeit, die 1:1-Zuordnung in Situationen zu erreichen, in denen Sie mehrere Datenbanken jeweils mit Servicesuperklassen und Serviceunterklassen haben. Hier entspricht jede Datenbank einer AIX-Superklasse und jede DB2-Serviceunterklasse wird einer Unterklasse von AIX Workload Manager zugeordnet. Die DB2-Servicesuperklasse wird in den Serviceklassendefinitionen von AIX Workload Manager nicht explizit gezeigt.

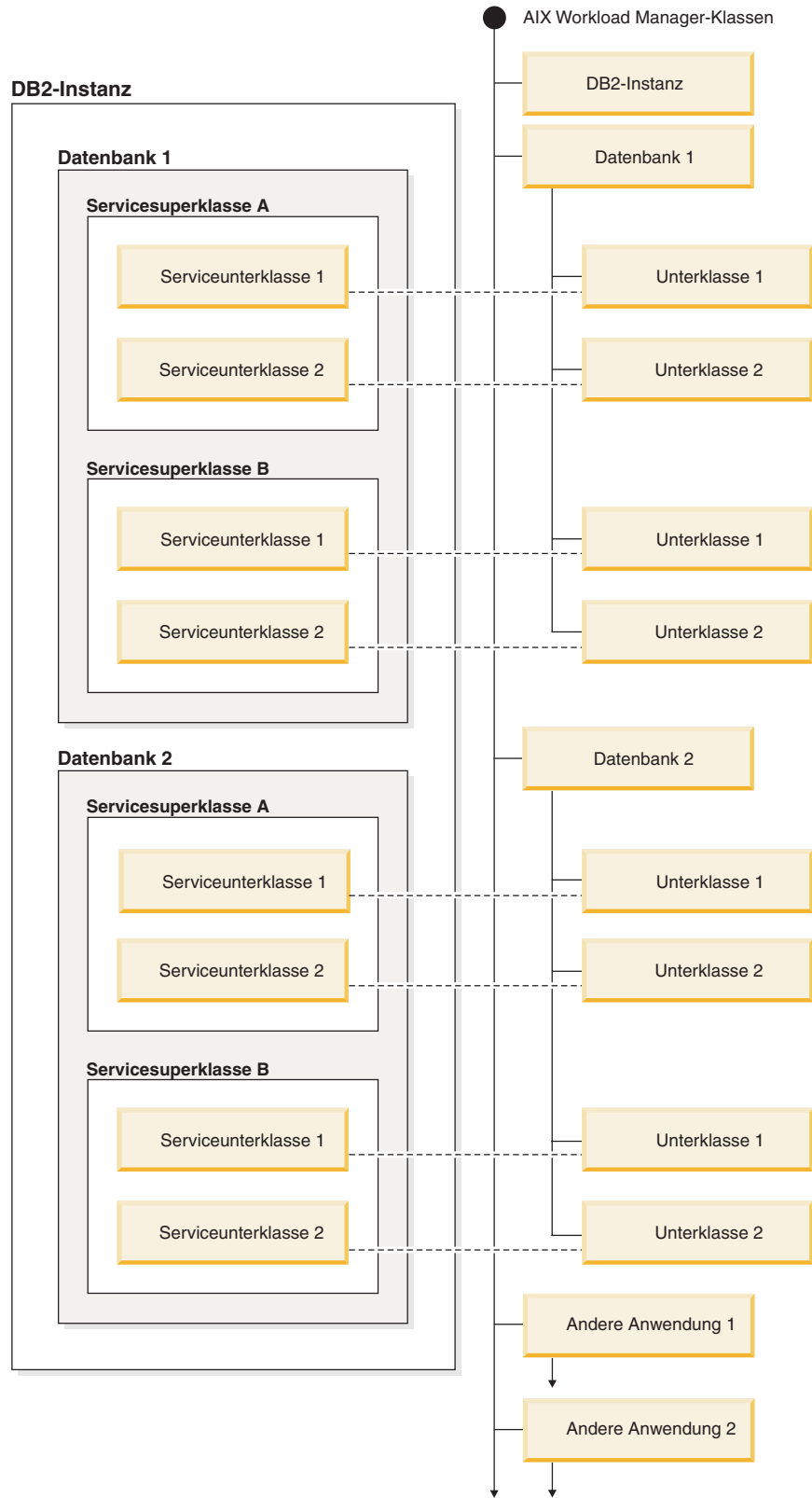


Abbildung 54. DB2-Serviceklassen, die Klassen von AIX Workload Manager zugeordnet sind (mit DB2-Serviceunterklassen)

Definieren von Zuordnungen zwischen DB2-Serviceklassen und AIX-Klassen

Die Zuordnung zwischen DB2-Serviceklassen und Klassen von AIX Workload Manager wird für die DB2-Serviceklasse mit dem Schlüsselwort `OUTBOUND CORRELATOR` der Anweisung `CREATE SERVICE CLASS` oder `ALTER SERVICE CLASS` angegeben.

Zum Einrichten der Klassen von AIX Workload Manager mit dem DB2-Datenserver sind die folgenden Schritte auszuführen:

1. Erstellen Sie die DB2-Servicesuperklassen und -Serviceunterklassen und geben Sie die `OUTBOUND CORRELATOR`-Kennungen (Tags) an.
2. Erstellen Sie die entsprechenden AIX-Klassen.
3. Erstellen Sie die zugehörigen Regeldateien von AIX Workload Manager, die die Zuordnungen des DB2-Workload-Managements zu AIX Workload Manager enthalten. Verwenden Sie hierzu die `OUTBOUND CORRELATOR`-Kennungen in den Kennungsspalten.
4. Starten Sie AIX Workload Manager.
5. Setzen Sie gegebenenfalls diese Konfiguration von AIX Workload Manager auf aktiv.

Wenn ein Thread einer DB2-Serviceklasse beitrifft, ruft der DB2-Datenserver die entsprechende API von AIX Workload Manager auf, um den Thread der entsprechenden AIX-Serviceklasse zuzuordnen. Der DB2-Datenserver sendet die AIX-Zielserviceklasse des Threads an AIX Workload Manager, indem er die im Parameter `OUTBOUND CORRELATOR` festgelegte Anwendungskennung übergibt.

Sie müssen sicherstellen, dass AIX Workload Manager korrekt installiert, konfiguriert und aktiv ist. Wenn der DB2-Datenserver nicht mit AIX Workload Manager kommunizieren kann, wird eine Nachricht in den `db2diag.log`-Dateien und im DB2-Administratorprotokoll protokolliert. Die Datenbankaktivität wird fortgesetzt.

Der DB2-Datenserver kann nicht erkennen, ob der Wert von `OUTBOUND CORRELATOR`, den er an AIX Workload Manager übergibt, von AIX Workload Manager erkannt wird. Sie müssen sicherstellen, dass der Wert, der für die DB2-Serviceklasse angegeben ist, den Anwendungskennungen entspricht, die DB2-Threads den AIX-Serviceklassen zuordnen. Wenn der Wert von `OUTBOUND CORRELATOR` von AIX Workload Manager nicht erkannt wird, wird die Datenbankaktivität fortgesetzt.

Andere wichtige Punkte:

- DB2-Serviceklassen können nicht die Vererbungsfunktion von AIX Workload Manager verwenden. Die Vererbung ist die Standardeinstellung für eine AIX-Serviceklasse. Die Vererbung muss explizit inaktiviert werden, indem das Vererbungsattribut auf `N0` gesetzt wird. Die Vererbung von AIX Workload Manager erzwingt die Zuordnung aller untergeordneten Threads und Prozesse zur selben Klasse, der auch der übergeordnete Thread oder Prozess zugeordnet ist. Wenn die Vererbung aktiviert ist, kann das DB2-Workload-Management die AIX Workload Manager-Klasse eines Threads nicht mithilfe von Tagging ändern. Durch diese Einschränkung wird jegliche Integration des DB2-Workload-Managements in AIX Workload Manager unbrauchbar. Der DB2-Datenserver kann nicht erkennen, ob die Vererbung von AIX Workload Manager aktiviert ist, und setzt keine Fehlernachricht ab, wenn die Vererbung aktiviert ist.

- DB2-Serviceklassen sind nicht kompatibel mit dem AIX Workload Manager-Feature für die manuelle Zuordnung. Mit der manuellen Zuordnung können Benutzer einen Prozess manuell einer bestimmten Klasse von AIX Workload Manager zuordnen. Durch die manuelle Zuordnung des DB2-Prozesses werden alle Threads im Prozess einer Zielklasse von AIX Workload Manager zugeordnet. Dabei wird die DB2-Serviceklassenzuordnungslogik unterdrückt und die Ergebnisse sind nicht vorhersagbar.

Weitere Informationen zu AIX Workload Manager finden Sie im AIX Information Center unter <http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp>.

Festlegen von Prozessorsteuerelementen für AIX-Klassen

Mithilfe von AIX Workload Manager kann der Anteil der Prozessorressourcen gesteuert werden, der den einzelnen Serviceklassen zugeordnet wird. Es gibt u. a. die Möglichkeit, für jede Serviceklasse den minimalen, maximalen oder relativen Prozessorressourcenanteil festzulegen.

Bei der Integration von AIX Workload Manager in das DB2-Workload-Management wird nur die Prozessorressourcenzuordnung unterstützt. Sie sollten keine Speicher- und E/A-Einstellungen für die AIX-Klassen festlegen. DB2-Hauptspeicher auf Datenbankebene wird von allen Agenten von unterschiedlichen DB2-Serviceklassen gemeinsam genutzt. Sie können also die Hauptspeicherzuordnung nicht auf unterschiedliche Serviceklassen aufteilen. Die E/A-Steuerung der AIX-Version unterstützt nicht das Threading-Modell der DB2-Engine. Zur Steuerung der Ein-/Ausgabe können Sie das Attribut für Vorabsezugriffspriorität einer DB2-Serviceklasse verwenden, um unterschiedliche E/A-Prioritäten für unterschiedliche DB2-Serviceklassen anzugeben.

Wenn Sie den einer Serviceklasse zugeordneten Prozessorressourcenanteil über AIX steuern, ändern Sie nicht gleichzeitig die Agentenprioritätseinstellung für diese DB2-Serviceklasse. Verwenden Sie nur einen dieser Mechanismen zur Steuerung des Zugriffs auf Prozessorressourcen. Sie können für eine Serviceklasse nicht sowohl den Wert von AGENT PRIORITY als auch den Wert von OUTBOUND CORRELATOR festlegen. Weitere Informationen finden Sie in „Agentenpriorität von Serviceklassen“ auf Seite 80.

Einstellungen von AIX Workload Manager sollten auf allen physischen Computern, die zu einer Instanz gehören, konsistent sein. Wenn z. B. die Ressourceneinstellung für eine AIX-Serviceklasse auf einem Computer hoch eingestellt ist, sollte dieselbe Einstellung für diese AIX-Serviceklasse auf allen anderen Computern verwendet werden. Wenn die Einstellungen zur Ressourcenauslastung über mehrere Computer inkonsistent sind, zeigen Anforderungen, die in derselben AIX-Serviceklasse ausgeführt werden, in unterschiedlichen Datenbankmitgliedern unterschiedliche Leistungswerte. Diese Situation kann zu einem schlechten Gesamtdurchsatz für Verbindungen in einer AIX-Serviceklasse führen.

Integration von Linux-Workload-Management in DB2-Workload-Management

Unter dem Betriebssystem Linux haben Sie aufgrund der optionalen Integration zwischen DB2-Serviceklassen und Linux-Klassen (Steuergruppen) die Möglichkeit, den Anteil der Prozessorressourcen zu steuern, der jeder Serviceklasse zugeordnet wird. Bei Aktivierung werden alle Threads, die in einer DB2-Serviceklasse ausge-

führt werden, einer Linux-Klasse zugeordnet, in der sie der von Ihnen definierten Prozessorressourcensteuerung unterliegen.

Zur Nutzung der Linux-Workload-Management-Unterstützung benötigen Sie eine Linux-Kernelversion 2.6.26 oder höher auf einem 64-Bit-System sowie das Bibliothekspaket `libcgroup`.

Das Linux-Workload-Management unterstützt eine Hierarchie von aus Superklassen und Unterklassen bestehenden Klassen mit Prozessoranteilen für Unterklassen, die proportional zu den Anteilen der übergeordneten Klasse unterteilt sind. Diese Anteile ermöglichen eine Methode zur Kontrolle über Prozessorressourcen, sodass alle Threads auf dem System immer ausgeführt werden, der Anteil der Prozessorzeit, die jeder Thread erhält, aber von der Anzahl der Anteile abhängt, die der Linux-Klasse zugeordnet sind.

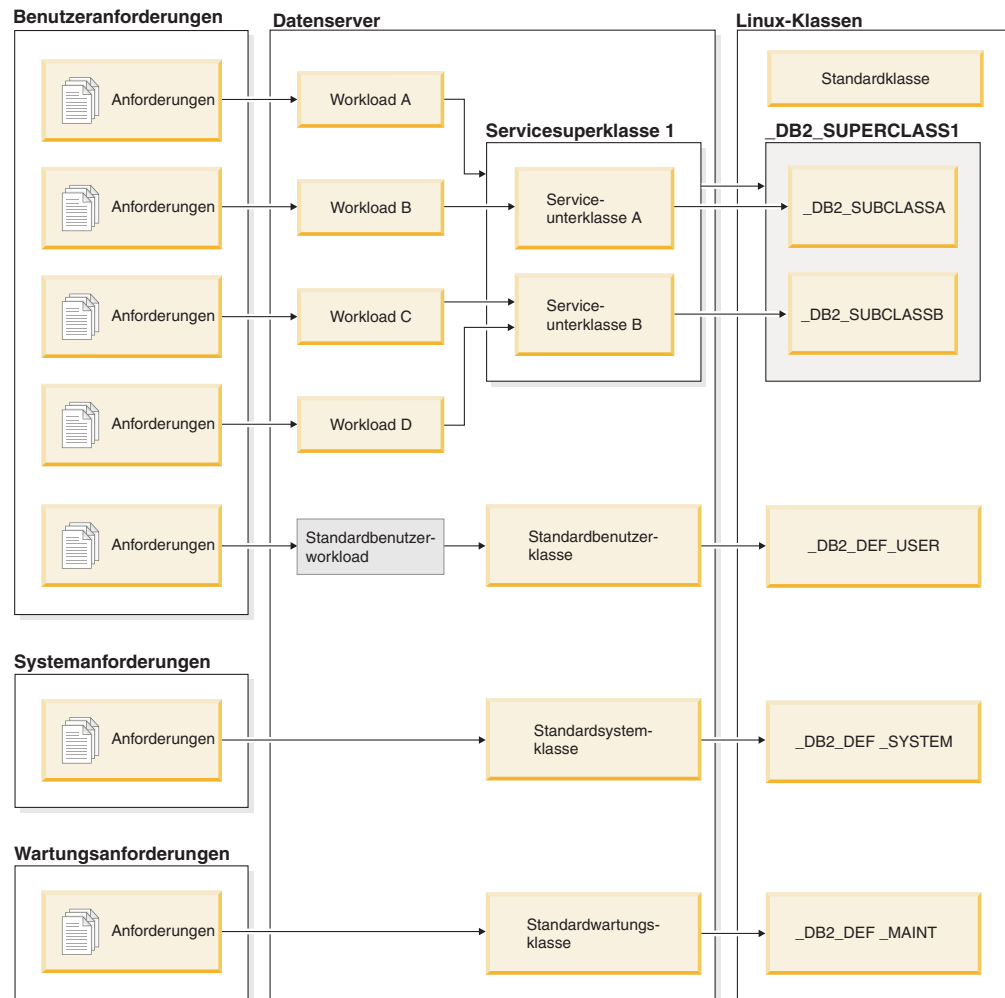
Unter dem Betriebssystem Linux werden Prozessorressourcen in Anteilen relativ zur Linux-Workload-Management-Standardklasse zugeordnet, deren Prozessoranteil standardmäßig einen Wert von 1024 hat. Wenn Sie keine weiteren Linux-Klassen definieren, werden alle Threads in dieser Standardklasse ausgeführt. Wenn Sie eine Klasse definieren, die einen Anteilswert von 1024 hat, dann erhält diese Klasse denselben Anteil an Prozessorressourcen wie die Linux-Standardklasse mit dem Standardprozessoranteil. Analog erhält eine Klasse mit einem Anteilswert von 2048 eine doppelt so hohe Zielprozessorauslastungsquote wie die Standardklasse. In komplexeren Systemen sollten Sie eine Erhöhung des Prozessoranteils der Linux-Standardklasse in Erwägung ziehen, die die Granularität für Anteile systemübergreifend verbessert, sodass Sie Prozessorressourcen genauer zuordnen können.

Empfohlene Zuordnungen zwischen DB2-Serviceklassen und Linux-Klassen

Sie sollten eine 1:1-Zuordnung zwischen DB2-Serviceklassen und Linux-Klassen verwenden. Dies erlaubt es Ihnen, die Linux-Prozessoranteile, die Aktivitäten in jeder DB2-Serviceklasse zugewiesen werden, individuell entsprechend der Geschäftspriorität anzupassen. Es ist wichtig, dass Sie jede DB2-Serviceklasse einer Linux-WLM-Klasse zuordnen, indem Sie entweder einen abgehenden Korrelator für jede Servicesuperklasse und Unterklasse setzen, oder durch Übernahme von der übergeordneten Serviceklasse für Unterklassen. Hierzu gehören die Standardserviceklassen `SYSDEFAULTSYSTEMCLASS`, `SYSDEFAULTMAINTENANCECLASS` und `SYSDEFAULTUSERCLASS`.

Die folgende Abbildung zeigt, wie zwei DB2-Serviceunterklassen unter derselben benutzerdefinierten Servicesuperklasse 1:1 Linux-Unterklassen unter einer gemeinsamen Superklasse zugeordnet werden können. In diesem Beispiel unterliegt Arbeit, die durch zwei Workloads für jede DB2-Serviceunterklasse erkannt und zugeordnet wird, der Prozessorressourcensteuerung der entsprechenden Linux-Unterklassen (`_DB2_SUBCLASSA`, `_DB2_SUBCLASSB`). Ebenfalls zu sehen sind drei Linux-Klassen, die den DB2-Workload-Management-Standardserviceklassen entsprechen (`_DB2_DEF_USER`, `_DB2_DEF_SYSTEM`, `_DB2_DEF_MAINT`). Wenn Sie das DB2-Workload-Management in das Linux-Workload-Management integrieren, sollten Sie diese zusätzlichen Linux-Klassen immer analog zu den DB2-Standardserviceklassen erstellen. Zur Vermeidung von Engpässen sollte die Linux-Klasse, die der DB2-Standardssystemklasse entspricht, mehr Prozessoranteile erhalten als alle anderen Linux-Klassen, denen DB2-Aktivitäten zugeordnet werden. Die Linux-Klasse, die der Standardverwaltungsklasse entspricht, sollte dagegen weniger Prozessoranteile erhalten.

Abbildung 55. Integration von DB2-Workload-Management und Linux-Workload-Management



Definieren von Zuordnungen zwischen DB2-Serviceklassen und Linux-Workload-Management-Klassen

Zur Integration des DB2-Workload-Managements in das Linux-Workload-Management, das als Betriebssystemservice ausgeführt wird, müssen die folgenden Schritte ausgeführt werden:

1. Definieren Sie die Linux-Klassen, Klassenberechtigungen und Prozessoranteile, indem Sie die Steuergruppenkonfigurationsdatei `/etc/cgconfig.conf` bearbeiten. Welche Linux-Klassen Sie erstellen, hängt von den Bedingungen Ihrer Geschäftsprioritäten für die von Ihrem Datenserver ausgeführte Arbeit ab. Wenn Sie z. B. Prozessorressourcen basierend auf der Quelle bestimmter Arbeiten anwenden wollen, erstellen Sie eine Linux-Klasse, die der DB2-Serviceklasse entspricht, der Arbeit von der Workload zugeordnet wird, die diese Arbeit erkennt. Definieren Sie einen Eintrag für jede Linux-Klasse entsprechend der zu erstellenden DB2-Serviceklasse, die Sie für die Zuordnung verwenden wollen. Die folgenden Abschnitte müssen in der Konfigurationsdatei `/etc/cgconfig.conf` enthalten sein:
 - `group`: Der Name der Linux-Klasse. Wenn Sie z. B. die Gruppe `_klasse1` angeben, erstellen Sie eine Superklasse mit dem Namen `"_klasse1"`. Wenn Sie

die Gruppe `_klasse1/_unterklasse1` angeben, erstellen Sie die Unterklasse "`_unterklasse1`" unter der Superklasse "`_klasse1`".

- `perm`: Der Berechtigungsabschnitt, der festlegt, wer die Zuordnung von Threads zu einer Linux-Klasse steuern und wer die Prozessoranteile von Klassen in der Konfigurationsdatei `/etc/cgconfig.conf` ändern kann.
 - `task`: Die Benutzer-ID (**uid**) und die Gruppen-ID (**gid**), deren Threads in der Linux-Workload-Management-Klasse ausgeführt werden können. Damit Linux-Workload-Management mit dem DB2-Workload-Management arbeiten kann, sollten Sie **uid** auf die Benutzer-ID des DB2-Instanzzeichners setzen.
 - `admin`: Die Benutzer-ID (**uid**) und die Gruppen-ID (**gid**), die die Prozessoranteile für eine Linux-Workload-Management-Klasse ändern können.
- `cpu`: Der Abschnitt mit der Definition der Prozessoranteile
 - `cpu.shares`: Der Anteil, der dieser Linux-Klasse relativ zur Standardklasse zugeordnet wurde

Die Konfigurationsdatei `/etc/cgconfig.conf` muss diese Abschnitte im folgenden Format enthalten:

```
# Superclass name
group _name
{
    perm
    {
        task
        {
            uid = db2inst1;
            gid = db2iadml;
        }
        admin
        {
            uid = db2inst1;
            gid = db2iadml;
        }
    }

    cpu
    {
        cpu.shares = 1024;
    }
}
```

2. Starten Sie den Linux-Workload-Management-Servicedämonprozess mit dem Befehl **service cgconfig start** und starten Sie dann Ihren DB2-Datenserver mit dem Befehl **db2start**.
3. Zur Zuordnung einer DB2-Serviceklasse zu einer der Linux-Klassen schließen Sie beim Erstellen oder Ändern der Serviceklasse den Namen der Linux-Klasse in die Klausel `OUTBOUND CORRELATOR` ein. Dies ordnet Threads von der DB2-Serviceklasse der externen Linux-Klasse zu.
4. Wenn Sie herausfinden wollen, welche Threads einer bestimmten Linux-Klasse zugeordnet sind, können Sie den Befehl `cat` in der Datei `/cgroup/klassenname/tasks` verwenden. Dabei steht *klassenname* für den Namen der fraglichen Linux-Klasse. Alle Threads, die nicht einer benutzerdefinierten Linux-Klasse zugeordnet sind, werden der Linux-Standardklasse unter `MOUNTPOINT/sysdefault` zugeordnet, wobei `MOUNTPOINT` in der Konfigurationsdatei `cgconfig.conf` definiert ist.
5. Wenn Sie Linux-Klassen hinzufügen oder entfernen wollen, müssen Sie den Linux-Workload-Management-Service mit dem Befehl **service cgconfig stop** stoppen, Ihre Änderungen vornehmen und dann den Service erneut starten. Das Stoppen des Services hat Auswirkungen auf das gesamte System, da alle

Tasks in die Standardklasse verschoben werden. Wenn Sie das Script `/etc/init.d/cgred` zum Starten des Servicedämonprozesses verwendet haben, setzen Sie `/etc/init.d/cgred stop` ab, um den Prozess zu stoppen.

Damit die Integration in das DB2-Workload-Management funktioniert, müssen Sie sicherstellen, dass der Linux-Workload-Management-Service korrekt installiert, konfiguriert und aktiv ist. Wenn der DB2-Datenserver nicht mit dem Linux-Workload-Management-Service kommunizieren kann, wird eine Nachricht in den `db2diag.log`-Dateien und im DB2-Administratorprotokoll protokolliert. Datenbankaktivitäten werden weiterhin ausgeführt.

Der DB2-Datenserver kann nicht erkennen, ob der abgehende Korrelator, den er an externe Workload-Manager übergibt, vom Linux-Workload-Management erkannt wird. Sie müssen sicherstellen, dass der Wert `OUTBOUND CORRELATOR`, der für eine DB2-Serviceklasse angegeben ist, dem Linux-Klassennamen entspricht, sodass DB2-Threads der Linux-Klasse zugeordnet werden. Wenn ein abgehender Korrelator nicht erkannt wird, werden Datenbankaktivitäten weiterhin ausgeführt.

Beispiel

Das folgende Beispiel veranschaulicht, wie Sie Möglichkeiten zur Linux-Workload-Management-Prozessorsteuerung durch die Integration in das DB2-Workload-Management nutzen können. In diesem Beispiel erstellen wir zwei benutzerdefinierte DB2-Serviceklassen, eine für Stapelanwendungen (`BATCHAPPS`) und eine für Onlineanwendungen (`ONLINEAPPS`). Der Einfachheit halber zeigt dieses Beispiel nicht die Standardserviceklassen, die in einer Implementierung enthalten sein sollten, die die empfohlene 1:1-Zuordnung zwischen DB2-Serviceklassen und Linux-Klassen erstellt. Da die Antwortzeit kritisch für die Onlineanwendungen ist, soll die Serviceklasse `ONLINEAPPS` den dreifachen Prozessoranteil bezogen auf Arbeit erhalten, die in der Linux-Standardklasse ausgeführt wird ($3 \times 1024 = 3072$ Anteile). Stapelanwendungen haben eine niedrigere Geschäftspriorität und der Klasse `BATCHAPPS` sollte die Hälfte der Prozessorressourcen im Vergleich zu Arbeit in der Linux-Standardklasse zugewiesen werden ($1024 / 2 = 512$ Anteile). Die gesamte andere Arbeit auf dem System wird in der Linux-Standardklasse ausgeführt. Beachten Sie, dass dieses Beispiel keine Linux-Klassen entsprechend der drei Standardserviceklassen des DB2-Workload-Managements erstellt.

Erstellen Sie für diese Konfiguration zuerst die beiden entsprechenden Linux-Klassen `_BATCHAPPS` und `_ONLINEAPPS` und legen Sie ihren relativen Prozessoranteil fest, indem Sie die Aufgabendatei `/etc/cgconfig.conf` editieren. Nach dem Editieren enthält die Aufgabendatei die folgenden beiden Einträge, einen für jede Linux-Klasse:

```
# Superclass ONLINEAPPS
group _ONLINEAPPS
{
    perm
    {
        task
        {
            uid = db2inst1;
            gid = db2iadm1;
        }
        admin
        {
            uid = db2inst1;
            gid = db2iadm1;
        }
    }
}
```

```

    cpu
    {
        # 3 x 1024 = 3072 Anteile
        cpu.shares = 3072;
    }
}

# Superclass BATCHAPPS
group _BATCHAPPS
{
    perm
    {
        task
        {
            uid = db2inst1;
            gid = db2iadml;
        }
        admin
        {
            uid = db2inst1;
            gid = db2iadml;
        }
    }

    cpu
    {
        # 1024 / 2 = 512 Anteile
        cpu.shares = 512;
    }
}

```

Die absolute Prozessorzeit (in Prozent), die jeder Linux-Klasse als Prozessoranteil zugeordnet wird, hat die folgenden Werte:

Tabelle 62. Prozessoranteile und absolute Prozessorzeit für Linux-Klassen

Linux-Klasse	Anteile	Absolute Prozessorzeit in Prozent
Standardklasse	1024 (Standardwert)	1024 / 4608 = 22 %
_ONLINEAPPS	1024 x 3 = 3072	3072 / 4608 = 67 %
_BATCHAPPS	1024 x ½ = 512	512 / 4608 = 11 %
	Insgesamt = 1024 + 3072 + 512 = 4608 Anteile	

Sobald die Linux-WLM-Klassen erstellt sind, können Sie den Linux-Workload-Management-Service starten:

```
service cgconfig start
```

Als Nächstes erstellen Sie die zugehörigen DB2-Serviceklassen mit den folgenden Anweisungen:

```
DB2 CREATE SERVICE CLASS BATCHAPPS OUTBOUND CORRELATOR '_BATCHAPPS'
DB2 CREATE SERVICE CLASS ONLINEAPPS OUTBOUND CORRELATOR '_ONLINEAPPS'
```

Setzen Sie den Befehl `cat ab`, um herauszufinden, welche Threads in einer Linux-Klasse ausgeführt werden. Für die geschäftskritische Linux-Klasse `_ONLINEAPPS` würden der Befehl und die Ausgabe wie folgt aussehen. Sie können sehen, dass sechs Threads in dieser Linux-Klasse ausgeführt werden:

```
cat /cgroup/_ONLINEAPPS/tasks
```

```
1056  
1087  
1107  
985  
1036  
1205
```

Kapitel 6. Lernprogramm für DB2-Workload-Management

Die Übungen in diesem Lernprogramm sollen Ihnen eine praktische Einführung in das DB2-Workload-Management zu bieten. Jede Übung befasst sich mit einem oder mehreren der im DB2-Workload-Management verfügbaren Workload-Management-Features.

Diese Übungen enthalten eine Anleitung für die Verwendung von DB2-Workload-Management-Features, die Sie für Ihre eigenen Zwecke anpassen können. Sie sollten jedoch beachten, dass die Erstkonfiguration, die Sie für Ihren Datenserver ausgewählt haben, anders sein kann und auf Ihren spezifischen Workload-Management-Zielen basieren sollte.

Vorbereitung

Dieses Lernprogramm ist für die Ausführung in der Datenbank SAMPLE konzipiert. Für seine Ausführung ist, sofern nichts anderes angegeben ist, die Berechtigung DBADM oder WLMADM erforderlich (oder die Berechtigung SQLADM, wenn nur die Klausel COLLECT ACTIVITY DATA angegeben ist). Bevor Sie fortfahren, sollten Sie auch die Instanz starten und die Datenbank SAMPLE aktivieren:

```
db2start  
db2 activate db sample
```

Einige der Befehls- und Abfrageanweisungen in diesen Übungen sind ziemlich lang. Sie können die meisten dieser Anweisungen in der Textdatei `wlm-tutorial-steps.txt` finden, aus der Sie die Anweisungen beim Durcharbeiten der Übungen kopieren können. Die Scripts, die die erforderlichen Workloads für die verschiedenen Übungen darstellen, sind ebenfalls enthalten.

Die Datei `wlm-tutorial-steps.txt` und die Workloadscripts finden Sie hier.

Übung 1: Einführung in die grundlegende Überwachung mit DB2-Workload-Management-Standardobjekten

Diese Übung veranschaulicht die grundlegenden Typen von Überwachungsdaten, die aus den Standardworkload- und Standardserviceklassenobjekten abgerufen werden können.

Geschätzte Zeit: 20-25 Minuten

Standardmäßig werden für jede Datenbank immer die Benutzerworkload (SYSDEFAULTUSERWORKLOAD) und eine Standardbenutzerserviceklasse (SYSDEFAULTUSERCLASS) erstellt. Mithilfe dieser Standardobjekte können die neuen DB2-Workload-Management-Überwachungsfeatures genutzt werden, ohne dass benutzerdefinierte Workloads oder Serviceklassen erstellt werden müssen. Wenn keine benutzerdefinierten Workloads und Serviceklassen erstellt werden, werden alle Benutzeraktivitäten diesen Standardobjekten zugeordnet.

In dieser Übung werden zwei unterschiedliche Überwachungsfeatures veranschaulicht:

1. Die Möglichkeit zum Erfassen von zusammengefassten Statistikdaten für alle Aktivitäten, die in einer Serviceklasse ausgeführt werden. Zusammengefasste

Aktivitätsstatistikdaten stellen eine wenig aufwendige Methode zur Darstellung der Arbeit in einer Serviceklasse als Ganzes dar. Sie zeigen Informationen wie die Anzahl der in der Serviceklasse ausgeführten Aktivitäten und die durchschnittlichen Laufzeiten dieser Aktivitäten an.

2. Die Möglichkeit zum Erfassen von Informationen zu einzelnen Aktivitäten. Aktivitätsinformationen können hilfreich sein, wenn die Leistung oder das Verhalten einer bestimmten Aktivität untersucht werden soll. Aktivitätsinformationen umfassen z. B. den Anweisungstext, die Kompilierungsumgebung usw. Aktivitätsinformationen sind aufwendiger zu erfassen als zusammengefasste Aktivitätsstatistikdaten und richten sich in der Regel auf eine bestimmte Teilmenge von Aktivitäten.

Schritt 1: Erstellen und Aktivieren von Ereignismonitoren

Stellen Sie eine Verbindung zur Datenbank her und erstellen und aktivieren Sie Ereignismonitore für Aktivitäten und Statistikdaten.

```
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
```

Schritt 2: Erfassen einzelner Aktivitäten

Aktivieren Sie die Erfassung von Daten zu einzelnen Aktivitäten mit der Klausel `COLLECT ACTIVITY DATA` in der Anweisung `CREATE WORKLOAD` oder `ALTER WORKLOAD`. Wenn die Klausel `COLLECT ACTIVITY DATA` für eine Workload angegeben ist, werden bei Abschluss der Aktivität Informationen zu jeder Aktivität, die von einem Vorkommen dieser Workload übergeben wurde, an den aktiven `ACTIVITIES`-Ereignismonitor gesendet. Über die Klausel `COLLECT ACTIVITY DATA` können Sie angeben, wie viele Daten erfasst werden sollen. Verwenden Sie dazu eine der folgenden Optionen:

- `WITHOUT DETAILS`: Aktivitätsinformationen ohne Anweisung und Kompilierungsumgebung erfassen.
- `WITH DETAILS`: Aktivitätsinformationen mit Anweisung und Kompilierungsumgebung erfassen.
- `WITH DETAILS AND VALUES`: Aktivitätsinformationen mit Anweisung und Kompilierungsumgebung erfassen und Datenwerte eingeben.

Bei dieser Übung geben Sie die Klausel `WITH DETAILS` an, sodass der Anweisungstext erfasst wird.

```
ALTER WORKLOAD SYSDEFAULTUSERWORKLOAD  
COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS
```

In diesem Beispiel werden Aktivitätsdaten für die Standardbenutzerworkload erfasst. Da zurzeit keine anderen benutzerdefinierten Workloads aktiv sind, werden also Informationen zu allen Benutzeraktivitäten erfasst. In einer Produktionsumgebung wäre dies zu aufwendig. Ein besserer Ansatz wäre, die relevanten Aktivitäten mit einer bestimmten benutzerdefinierten Workload oder Serviceklasse einzugrenzen und die Klausel `COLLECT ACTIVITY DATA` nur auf diese Workload oder Serviceklasse anzuwenden.

Weitere Informationen: Die Klausel `COLLECT ACTIVITY DATA` kann auch für eine Serviceklasse, Arbeitsklasse (über eine Arbeitsaktion) oder einen Schwellenwert an-

gegeben werden. Wenn die Klausel für eine Serviceklasse angegeben wird, werden Informationen zu allen Aktivitäten erfasst, die in dieser Serviceklasse ausgeführt werden. Wenn sie für eine Arbeitsklasse (über eine Arbeitsaktion) angegeben wird, werden Informationen zu allen Aktivitäten erfasst, auf die die Arbeitsaktion angewendet wird. Wenn die Klausel für einen Schwellenwert angegeben wird, werden Aktivitätsinformationen bei einem Verstoß gegen diesen Schwellenwert erfasst.

Schritt 3: Erfassen von zusammengefassten Aktivitätsstatistikdaten

Aktivieren Sie mit der Klausel `COLLECT AGGREGATE ACTIVITY DATA` die Erfassung der zusammengefassten Aktivitätsstatistikdaten für die Standardunterklasse unter der Standardbenutzerserviceklasse. Wenn diese Klausel angegeben wird, werden zusammengefasste Statistikdaten für die entsprechende Serviceklasse im Speicher gepflegt (z. B. Statistikdaten wie die durchschnittliche Aktivitätslaufzeit). Die Statistikdaten können mit der Statistiktabellenfunktion der Serviceunterklasse angezeigt werden oder sie können erfasst und zur späteren Analyse an den Statistikereignismonitor gesendet werden.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA BASE
```

Weitere Informationen: Es gibt eine Reihe von Statistikdaten, die standardmäßig für alle DB2-Workload-Management-Objekte erfasst werden. Die Klausel `COLLECT AGGREGATE ACTIVITY DATA` aktiviert die Erfassung einer Reihe von zusätzlichen optionalen Statistikdaten wie z. B. das Histogramm der Aktivitätslaufzeit.

In diesem Beispiel werden alle Benutzeraktivitäten in der Serviceunterklasse `SYSDEFAULTSUBCLASS` unter der Servicesuperklasse `SYSDEFAULTUSERCLASS` ausgeführt, da keine benutzerdefinierten Serviceklassen erstellt wurden. Es werden also Informationen für alle Benutzeraktivitäten erfasst.

Schritt 4: Ausführen einiger Aktivitäten

Führen Sie einige Aktivitäten aus, die dazu führen, dass die Statistikdaten aktualisiert und die Aktivitäten erfasst werden.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
```

Die Scripts, die Anwendungen darstellen, (z. B. `work1.db2` und `work2.db2`) trennen Ihre Verbindung zur Datenbank, sodass Sie nach der Ausführung die Verbindung wiederherstellen müssen.

Schritt 5: Anzeigen der Statistikdaten

Sie können die Serviceklassenstatistikdaten mit der Tabellenfunktion `WLM_GET_SERVICE_SUBCLASS_STATS` anzeigen. Beispiel:

```
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
```

Die Ausgabe dieser Abfrage könnte folgendermaßen aussehen:

```

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 +1.40288000000000E+002

1 record(s) selected.

```

Die Spalte COORD_ACT_COMPLETED_TOTAL gibt an, wie viele Aktivitäten in dieser Serviceklasse erfolgreich abgeschlossen wurden. Die Spalte LAST_RESET gibt an, wann die Statistik für diese Serviceklasse zuletzt zurückgesetzt wurde.

Weitere Informationen: Wenn Sie die zusammengefassten Aktivitätsstatistikdaten für eine Serviceklasse mit der Klausel COLLECT AGGREGATE ACTIVITY DATA nicht aktivieren, haben einige der von der Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS aufgelisteten Statistikdaten den Wert NULL.

Schritt 6: Senden von Statistikdaten an den Ereignismonitor

Mit der gespeicherten Prozedur WLM_COLLECT_STATS können Sie die Statistikdaten für alle DB2-Workload-Management-Objekte an den aktiven Statistikereignismonitor senden. Wenn Statistikdaten erfasst und an den Statistikereignismonitor gesendet werden, werden die Werte zurückgesetzt.

```
CALL SYSPROC.WLM_COLLECT_STATS()
```

Weitere Informationen: Auch wenn es keinen aktiven Statistikereignismonitor gibt, können Sie dennoch die Prozedur WLM_COLLECT_STATS verwenden, um die Statistikdaten zurückzusetzen. Die aktuellen Werte gehen dabei jedoch verloren. Es ist möglich, die Erfassung von Workload-Management-Statistikdaten mit dem Datenbankkonfigurationsparameter WLM_COLLECT_INT zu automatisieren. Wenn Sie diesen Parameter auf einen Wert ungleich null setzen, werden die Workload-Management-Statistikdaten automatisch alle `wlm_collect_int` Minuten erfasst (als riefen Sie die Prozedur WLM_COLLECT_STATS manuell alle `wlm_collect_int` Minuten auf).

Schritt 7: Erneutes Anzeigen der Statistikdaten

Rufen Sie die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS erneut auf. Beachten Sie, dass die Zeitmarke in der Spalte LAST_RESET aktualisiert wurde und dass die Statistikdaten zurückgesetzt wurden.

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

```

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 record(s) selected.

Schritt 8: Anzeigen der durch den Statistikereignismonitor erfassten Serviceklassenstatistikdaten

Die Prozedur WLM_COLLECT_STATS sendete die Serviceklassenstatistikdaten an den Statistikereignismonitor. Sie können sich die vom Ereignismonitor erfassten Statistikdaten mit einer Anweisung wie der folgenden ansehen:

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

```

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 record(s) selected.

Jedes Mal, wenn Statistikdaten an den Ereignismonitor gesendet werden, wird ein Statistikdatensatz für jedes DB2-Workload-Management-Objekt erstellt. Beachten Sie die beiden Zeitmarken LAST_WLM_RESET und STATISTICS_TIMESTAMP. Das Zeitintervall von LAST_WLM_RESET bis STATISTICS_TIMESTAMP gibt den Zeitraum an, in dem die Statistikdaten im Datensatz erfasst wurden. STATISTICS_TIMESTAMP gibt an, wann die Statistikdaten erfasst wurden. Beachten Sie, dass die durchschnittliche Laufzeit für Aktivitäten für die Standardsystem- und Standardwartungsserviceklassen auf dem Koordinator -1 beträgt. Die Statistik der durchschnittlichen Aktivitätslaufzeit wird nur dann für eine Serviceklasse gepflegt, wenn zusammengefasste Aktivitätsstatistikdaten mit der Klausel COLLECT AGGREGATE ACTIVITY DATA aktiviert wurden.

Schritt 9: Anzeigen von Aktivitätsinformationen

Informationen zu den einzelnen Aktivitäten in der Standardbenutzerworkload wurden durch die Angabe der Klausel COLLECT ACTIVITY DATA für die Standardworkload in Schritt 2 auch vom Aktivitätsereignismonitor erfasst. Sie können sich diese Aktivitätsinformationen mit einer Abfrage wie der folgenden ansehen:

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

APPL_NAME	CLIENT_APP_NAME	APPL_ID
ACTIVITY_ID	UOW_ID	STMT_TEXT
db2bp	CLP w1mmonbasic.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
...		

Beachten Sie, dass Sie einige Warnungen bezüglich des Abschneidens (SQL0445) erhalten können.

Wenn der Befehlszeilenprozessor ein Script ausführt, setzt er das Sonderregister CURRENT CLIENT_APPLNAME auf "CLP **scriptname**". So können Sie aus der Abfrage oben feststellen, welches Script die jeweilige Aktivität übergab.

Schritt 10: Zurücksetzen für die nächste Übung

Aktualisieren Sie die Workload SYSDEFAULTUSERWORKLOAD und die Serviceunterklasse SYSDEFAULTSUBCLASS, sodass keine Aktivitätsdaten oder zusammengefassten Aktivitätsstatistikdaten erfasst werden, inaktivieren Sie Ereignismonitore, löschen Sie den Inhalt der Aktivitäts- und Statistiktabellen und rufen Sie WLM_COLLECT_STATS() auf, um die Statistikdaten zurückzusetzen.

```
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 ACTIVITYSTMT_DB2ACTIVITIES
DELETE FROM SCSTATS_DB2STATISTICS
DELETE FROM WLSTATS_DB2STATISTICS
```

```
CALL WLM_COLLECT_STATS()
```

Übung 2: Eingrenzen der Aktivitäten mit Serviceklassen und Workloads

Diese Übung veranschaulicht, wie Serviceklassen erstellt und Aktivitäten mithilfe einer Workload an eine Serviceklasse gesendet werden können. Sie zeigt auch, wie einige der WLM-Überwachungsfeatures verwendet werden können, um die Workload zu ermitteln, der Aktivitäten zugeordnet werden, und um Informationen zu den in einer Serviceklasse und unter einer Workload ausgeführten Aktivitäten zu erhalten.

Geschätzte Zeit: 20-25 Minuten

Serviceklassen stellen die primäre Position für die Ressourcensteuerung für Datenbankaktivitäten dar. Auch zu Überwachungszwecken sind sie nützlich. Sie können z. B. Statistikdaten für Aktivitäten in einer bestimmten Serviceklasse erfassen, um festzustellen, ob die Leistungsziele für diese Serviceklasse erfüllt werden. Standardmäßig werden drei Standardserviceklassen (SYSDEFAULTSYSTEMCLASS, SYSDEFAULTMAINTENANCECLASS und SYSDEFAULTUSERCLASS) für jede Datenbank erstellt. Wenn keine benutzerdefinierten Serviceklassen erstellt werden, werden Benutzeraktivitäten unter der Standardbenutzerserviceklasse (SYSDEFAULTUSERCLASS) ausgeführt.

Eine Workload ist eine Entität, die mindestens eine UOW (Unit of Work - Arbeitseinheit) basierend auf Kriterien wie der Systembenutzer-ID, Sitzungsbenutzer-ID usw. gruppiert. Workloads sind eine Möglichkeit, Arbeit zur späteren Verwaltung einer Serviceklasse zuzuordnen. Für jede Datenbank wird eine Standardbenutzerworkload (SYSDEFAULTUSERWORKLOAD) und eine Standardverwaltungsworkload (SYSDEFAULTADMWORKLOAD) erstellt. Wenn keine benutzerdefinierten Workloads erstellt werden, werden alle Benutzeraktivitäten der Standardbenutzerworkload zugeordnet.

In dieser Übung werden vier separate Features veranschaulicht:

- Erstellen einer Serviceklasse
- Erstellen einer Workload
- Untersuchen der grundlegenden Workloadstatistikdaten
- Erfassen von Aktivitätsinformationen zu Aktivitäten, die in einer einzelnen Workload ausgeführt werden

Schritt 1: Untersuchen, wo Aktivitäten ausgeführt werden, wenn keine benutzerdefinierten Serviceklassen und Workloads vorhanden sind

Zuerst untersuchen Sie, wo Aktivitäten ausgeführt werden, wenn keine benutzerdefinierte Serviceklasse oder Workload vorhanden ist. Alle DB2-Aktivitäten werden einer Workload zugeordnet und in einer Serviceklasse ausgeführt. Wenn keine benutzerdefinierten Serviceklassen erstellt sind, werden Aktivitäten in der Standardunterklasse (SYSDEFAULTUSERCLASS) unter der Standardbenutzerserviceklasse (SYSDEFAULTUSERCLASS) ausgeführt. Wenn keine benutzerdefinierten Workloads erstellt sind, werden Aktivitäten unter der Standardbenutzerworkload (SYSDEFAULTUSERWORKLOAD) ausgeführt.

Führen Sie die Scripts `work1.db2` und `work2.db2` aus und untersuchen Sie dann die Statistikdaten für `SYSDEFAULTSUBCLASS` von `SYSDEFAULTUSERCLASS` mithilfe von `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

```

Die Ausgabe könnte folgendermaßen aussehen:

SUPERCLASS D_TOTAL	SUBCLASS	COORD_ACT_COMPLETE
-----	-----	-----
SYSDEFAULTSYSTEMCLASS 0	SYSDEFAULTSUBCLASS	
SYSDEFAULTMAINTENANCECLASS 0	SYSDEFAULTSUBCLASS	
SYSDEFAULTUSERCLASS 75	SYSDEFAULTSUBCLASS	

3 record(s) selected.

Beachten Sie, dass alle Aktivitäten in der Servicesuperklasse SYSDEFAULTUSERCLASS ausgeführt werden.

Weitere Informationen: Es gibt noch zwei weitere Serviceklassen: SYSDEFAULTSYSTEMCLASS und SYSDEFAULTMAINTENANCECLASS. Diese Serviceklassen werden für interne Verwaltungstasks sowie für Tasks auf Systemebene verwendet. In diesen Serviceklassen werden keine Benutzeraktivitäten ausgeführt. Sie werden möglicherweise auch in diesen Serviceklassen Aktivitätszähler ungleich null feststellen, wenn der DB2-Datenserver interne Aktivitäten abgesetzt hat.

Verwenden Sie die Tabellenfunktion WLM_GET_WORKLOAD_STATS zum Anzeigen von Workloadstatistikdaten, um zu ermitteln, welche Workload den Anwendungen zugeordnet ist.

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

WL_DEF_NAME	WLO_COMPLETED_TOTAL	CONCURRENT_WLO_ACT_TOP
-----	-----	-----
SYSDEFAULTUSERWORKLOAD	3	5
SYSDEFAULTADMWORKLOAD	0	0

2 record(s) selected.

Beachten Sie, dass sowohl für beide Scripts (work1.db2 und work2.db2) ein Workloadvorkommen vorhanden ist als auch für die Verbindung, die zur Ausführung des vorherigen Befehls verwendet wurde.

Schritt 2: Erstellen einer Serviceklasse und einer Workload

Erstellen Sie eine Serviceklasse und anschließend eine Workload, sodass alle Aktivitäten aus dem Script work1.db2 der neu erstellten Serviceklasse zugeordnet werden. Wenn der Befehlszeilenprozessor ein Script ausführt, wird der Wert des Sonderregisters CURRENT CLIENT_APPLNAME auf "CLP **scriptname**" gesetzt.

```
CREATE SERVICE CLASS work1_sc

CREATE WORKLOAD work1_w1 CURRENT CLIENT_APPLNAME('CLP work1.db2')
SERVICE CLASS work1_sc
```

Weitere Informationen: Es gibt eine Reihe von Attributen, die beim Erstellen einer Workload oder einer Serviceklasse angegeben werden können. Sie können z. B. beim Erstellen einer Workload die Verbindung basierend auf dem Anwendungsnamen, dem Sitzungsbenutzer usw. angeben. Weitere Informationen finden Sie in der Dokumentation zu CREATE WORKLOAD und CREATE SERVICE CLASS.

Schritt 3: Erteilen des Zugriffsrechts USAGE für eine Workload

Erteilen Sie das Zugriffsrecht GRANT für die Workload. (Hierfür wird die Berechtigung ACCESSCTRL oder SECADM vorausgesetzt.)

```
GRANT USAGE ON WORKLOAD work1_w1 TO PUBLIC
```

Weitere Informationen: Eine Verbindung kann nur dann einer Workload zugeordnet werden, wenn der Sitzungsbenutzer das Zugriffsrecht USAGE für die Workload hat. Dies ist erforderlich, um zu verhindern, dass Benutzer die Verbindungsattribute ihrer Anwendung ändern, um ihre Arbeit in einer Serviceklasse höherer Priorität ausführen zu lassen. Einige Verbindungsattribute können programmgesteuert geändert werden (z. B. über die API sqleseti). In dieser Übung erteilen wir das Zugriffsrecht USAGE einfach dem Benutzer PUBLIC. In einem echten System würden Sie wohl etwas kritischer vorgehen. Da die Datenbank SAMPLE als DBADM ausgeführt wird, könnte dieser Schritt ganz übersprungen werden.

Schritt 4: Zurücksetzen der Statistikdaten

Setzen Sie die Statistikdaten mit der Funktion WLM_COLLECT_STATS zurück, um die erfassten Statistikdaten zu löschen.

```
CALL SYSPROC.WLM_COLLECT_STATS()
```

Schritt 5: Ausführen einiger Aktivitäten

Führen Sie die beiden Scripts work1.db2 und work2.db2 aus.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
```

Schritt 6: Anzeigen der Workload- und Serviceklassenstatistikdaten

Verwenden Sie die Tabellenfunktion WLM_GET_WORKLOAD_STATS zum Anzeigen von Workloadstatistikdaten, um zu ermitteln, welche Workload den Anwendungen zugeordnet ist.

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

WL_DEF_NAME	WLO_COMPLETED_TOTAL	CONCURRENT_WLO_ACT_TOP
WORK1_WL	1	5
SYSDEFAULTUSERWORKLOAD	1	5
SYSDEFAULTADMWORKLOAD	0	0

Beachten Sie, dass unter WORK1_WL ein Workloadvorkommen ausgeführt wurde, nämlich das Script work1.db2. Ein Workloadvorkommen, nämlich work2.db2, wurde unter SYSDEFAULTUSERWORKLOAD ausgeführt.

Eventuell wird ein zweites Workloadvorkommen für SYSDEFAULTUSERWORKLOAD angezeigt. Dies ist dann die Verbindung, die zum Aufruf der Prozedur WLM_COLLECT_STATS verwendet wurde. WLM_COLLECT_STATS ist eine asynchrone Prozedur, die vor der eigentlichen Erfassung der Statistikdaten abgeschlossen sein kann und daher eventuell nicht enthalten ist.

Sie können auch die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS verwenden, um zu zeigen, unter welcher Serviceklasse die Aktivitäten durch die Erstellung der neuen Workload ausgeführt werden.

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

SUPERCLASS	SUBCLASS	COORDACTCOMP
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	37
WORK1_SC	SYSDEFAULTSUBCLASS	37

Beachten Sie die Aktivitäten, die unter WORK1_SC aufgrund der Workloadzuordnung WORK1_WL ausgeführt wurden.

Schritt 7: Erstellen einer weiteren Serviceklasse und Workload

Erstellen Sie eine zweite Serviceklasse und dann eine Workload, sodass alle Aktivitäten der Anwendung work2.db2 der neu erstellten Serviceklasse zugeordnet werden. Richten Sie außerdem die Workload so ein, dass sie einige Aktivitätsdaten erfasst. In diesem Beispiel werden nur Aktivitätsdaten ohne zusätzliche Details oder Werte erfasst.

```
CREATE SERVICE CLASS work2_sc

CREATE WORKLOAD work2_w1
    CURRENT_CLIENT_APPLNAME('CLP work2.db2')
    SERVICE CLASS work2_sc
    COLLECT ACTIVITY DATA
```

Weitere Informationen: Wenn die Klausel COLLECT ACTIVITY DATA für eine Workload angegeben ist, werden Informationen zu jeder Aktivität, die von einem Vorkommen dieser Workload übergeben wurde, bei Abschluss der Aktivität an den aktiven ACTIVITIES-Ereignismonitor gesendet. Über die Klausel COLLECT ACTIVITY DATA können Sie angeben, wie viele Daten erfasst werden sollen. Verwenden Sie dazu eine der folgenden Optionen:

- WITHOUT DETAILS: Aktivitätsinformationen ohne Anweisung und Kompilierungsumgebung erfassen.

- WITH DETAILS: Aktivitätsinformationen mit Anweisung und Kompilierungsumgebung erfassen.
- WITH DETAILS AND VALUES: Aktivitätsinformationen mit Anweisung und Kompilierungsumgebung erfassen und Datenwerte eingeben.

Schritt 8: Aktivieren des Aktivitätsereignismonitors

Aktivieren Sie die Ereignismonitore für Aktivitäten.

Der Aktivitätsereignismonitor wurde in Übung 1 erstellt.

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

Schritt 9: Zurücksetzen der Statistikdaten und Ausführen einiger Aktivitäten

Verwenden Sie die gespeicherte Prozedur WLM_COLLECT_STATS, um die Statistikdaten erneut zurückzusetzen, und führen Sie die Scripts work1.db2 und work2.db2 erneut aus.

```
CALL SYSPROC.WLM_COLLECT_STATS()
```

```
db2 -o -tvf work1.db2
```

```
db2 -o -tvf work2.db2
```

Schritt 10: Anzeigen der Workload- und Serviceklassenstatistikdaten

Verwenden Sie die Tabellenfunktion WLM_GET_WORKLOAD_STATS erneut, um zu ermitteln, welcher Workload die Anwendungen zugeordnet sind.

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

WL_DEF_NAME	WLO_COMPLETED_TOTAL	CONCURRENT_WLO_ACT_TOP
WORK1_WL	1	5
WORK2_WL	1	5
SYSDEFAULTUSERWORKLOAD	0	0
SYSDEFAULTADMWORKLOAD	0	0

Beachten Sie, dass dieses Mal für beide Workloaddefinitionen ein Workloadvorkommen ausgeführt wird, ein Mal für jedes Script.

Eventuell sehen Sie ein Workloadvorkommen, das für SYSDEFAULTUSERWORKLOAD ausgeführt wurde. Dies hängt davon ab, ob das Workloadvorkommen, über das der Aufruf an die Prozedur WLM_COLLECT_STATS übergeben wurde, vor Beginn der Statistikdatenerfassung geschlossen wurde.

Verwenden Sie WLM_GET_SERVICE_SUBCLASS_STATS erneut, um zu zeigen, unter welcher Serviceklasse die Aktivitäten durch die Erstellung der neuen Workload ausgeführt werden.

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

SUPERCLASS	SUBCLASS	COORDACTCOMP
SYSDEFAULTSYSTEMCLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTMAINTENANCECLASS	SYSDEFAULTSUBCLASS	0
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	1
WORK1_SC	SYSDEFAULTSUBCLASS	37
WORK2_SC	SYSDEFAULTSUBCLASS	37

Beachten Sie, dass dieses Mal aufgrund der WORK2_WL-Zuordnung einige Aktivitäten unter der Servicesuperklasse WORK2_SC ausgeführt wurden. Die eine Aktivität unter SYSDEFAULTUSERCLASS ist die Abfrage, die zuvor mit WLM_GET_WORKLOAD_STATS ausgeführt wurde.

Schritt 11: Anzeigen der erfassten Aktivitätsdaten

Fragen Sie die Aktivitätentabelle nach Informationen zu den ausgeführten Aktivitäten ab. Nur die Aktivitäten aus dem Script work2.db2 wurden erfasst, da nur für die Workloaddefinition WORK2_WL das Attribut COLLECT ACTIVITY DATA angegeben war.

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

WL_DEF_NAME	APPL_NAME	ACT_TYPE
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	WRITE_DML
WORK2_WL	db2bp	WRITE_DML
WORK2_WL	db2bp	WRITE_DML
WORK2_WL	db2bp	WRITE_DML
WORK2_WL	db2bp	WRITE_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	READ_DML
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	READ_DML
WORK2_WL	db2bp	READ_DML
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	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
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL
WORK2_WL	db2bp	DDL
:		
:		

Schritt 12: Zuordnen von Ressourcen zu Serviceklassen

Nun, da Sie die von diesen beiden Scripts abgesetzten Aktivitäten in getrennte Serviceklassen eingegrenzt haben, können Sie den Serviceklassen Ressourcen zuordnen oder die in diesen Serviceklassen ausgeführten Aktivitäten überwachen. Einige Beispiele: Wenn die vom Script work2.db2 ausgeführte Arbeit wichtiger ist als die vom Script work1.db2 ausgeführte Arbeit, könnten Sie die Priorität von Agenten in der Serviceklasse WORK2_SC mit einer Anweisung wie der folgenden erhöhen.

In UNIX-Betriebsumgebungen (ein negativer Wert bedeutet eine höhere Priorität):

```
ALTER SERVICE CLASS WORK2_SC AGENT PRIORITY -6
```

In Windows-Betriebsumgebungen (ein positiver Wert bedeutet eine höhere Priorität):

```
ALTER SERVICE CLASS WORK2_SC AGENT PRIORITY 6
```

Wenn Sie Details zu den einzelnen Aktivitäten in der Serviceklasse WORK2_SC erfassen wollen, könnten Sie die Aktivitätserfassung für diese Serviceklasse mit dem folgenden Befehl aktivieren:

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER WORK2_SC
  COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS
```

Schritt 13: Zurücksetzen für die nächste Übung

Aktualisieren Sie die Workload WORK2_WL, sodass keine Aktivitätsdaten erfasst werden, inaktivieren Sie den Ereignismonitor, bereinigen Sie die Ereignismonitortabelle und rufen Sie WLM_COLLECT_STATS() auf, um die Statistikdaten zurückzusetzen.

```
ALTER WORKLOAD work2_w1
  COLLECT ACTIVITY DATA NONE

SET EVENT MONITOR DB2ACTIVITIES STATE 0

DELETE from ACTIVITY_DB2ACTIVITIES

CALL WLM_COLLECT_STATS()
```

Übung 3: Verwenden von Schwellenwerten zum Steuern von Rogue-Aktivitäten und Verwenden des Schwellenwertverstoßmonitors

Diese Übung veranschaulicht, wie Sie anhand von Schwellenwerten unsachgemäße Ressourcenauslastung oder eine beginnende Systemüberlastung erkennen können, indem Sie für den Verbrauch bestimmter Ressourcen Grenzwerte definieren.

Geschätzte Zeit: 15-20 Minuten

Wird gegen einen Schwellenwert verstoßen, kann eine definierte Aktion ausgelöst werden. Die folgenden Aktionen werden unterstützt:

- STOP EXECUTION: Die Aktivität, die den Schwellenwertverstoß verursachte, wird nicht weiter verarbeitet.
- CONTINUE: Die Verarbeitung wird fortgesetzt.
- Erfassen von Informationen zur Aktivität, die den Schwellenwertverstoß verursacht. Diese Aktion kann in Verbindung mit der Aktion CONTINUE oder STOP EXECUTION angegeben werden.

Unabhängig davon, ob eine Aktivität, die gegen einen Schwellenwert verstößt, gestoppt wird oder weiter ausgeführt werden darf, wird bei jedem Verstoß gegen einen Schwellenwert ein Datensatz des Verstoßes in einen aktiven Ereignismonitor THRESHOLD VIOLATIONS geschrieben (vorausgesetzt, es wurde vorab ein Ereignismonitor definiert). Der Datensatz enthält Informationen wie den Schwellenwert, gegen den verstoßen wurde, den Zeitpunkt des Verstoßes und die Schwellenwertaktion.

Diese Übung veranschaulicht, wie mithilfe von Schwellenwerten Rogue-Aktivitäten erkannt werden können und verhindert werden kann, dass derartige Aktivitäten auf Ihrem System ausgeführt werden und Systemressourcen verbrauchen. Eine Rogue-Aktivität ist jede Aktivität, die eine unerwartet große Ressourcenmenge nutzt, z. B. eine Abfrage, die ungewöhnlich lange ausgeführt wird oder die eine unerwartet große Ergebnismenge zurückgibt.

Schritt 1: Erstellen eines Ereignismonitors für Schwellenwertverstöße

Erstellen und aktivieren Sie einen WRITE-TO-TABLE-Ereignismonitor, der zur Erfassung der Schwellenwertverstoßinformationen verwendet wird, und aktivieren Sie den in Übung 1 erstellten Aktivitätsereignismonitor.

```
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
```

Schritt 2: Erstellen einer Workload

Erstellen Sie eine Workload, sodass alle Aktivitäten aus dem Script workth.db2 der Serviceklasse work1_sc zugeordnet werden.

Die Serviceklasse work1_sc ist bereits vorhanden. Sie wurde in Übung 2 erstellt.

```
CREATE WORKLOAD workth_wl
    CURRENT CLIENT_APPLNAME('CLP workth.db2')
    SERVICE CLASS work1_sc
```

Schritt 3: Erstellen von Schwellenwerten

Erstellen Sie zwei Schwellenwerte, von denen einer (th_estcost) ein Schwellenwert vom Typ ESTIMATEDSQLCOST und der andere (th_sqlrows) ein Schwellenwert vom Typ SQLROWSRETURNED ist. Wenden Sie diese auf die Serviceklasse an, deren Aktivitäten Sie steuern wollen (in diesem Fall die Serviceklasse work1_sc).

Der Schwellenwert `th_estcost` gibt eine Obergrenze (10.000 Timeron) für den vom Optimierungsprogramm geschätzten Aufwand (in Timeron) für eine in der Serviceklasse `work1_sc` ausgeführte Aktivität an. Wenn eine Abfrage mit einem geschätzten Aufwand von mehr als 10.000 Timeron versucht, in der Serviceklasse `work1_sc` ausgeführt zu werden, wird gegen diesen Schwellenwert verstoßen und die Abfrage darf nicht ausgeführt werden.

Der Schwellenwert `th_sqlrows` gibt an, dass jede in der Serviceklasse `work1_sc` ausgeführte Aktivität höchstens 30 Zeilen vom Datenserver zurückgeben kann. Wenn eine Abfrage versucht, mehr als 30 Zeilen zurückzugeben, wird gegen diesen Schwellenwert verstoßen, und es werden nur 30 Zeilen an den Client zurückgegeben. Die Abfrage wird gestoppt. Darüber hinaus werden Daten zu der Aktivität erfasst, die den Schwellenwertverstoß verursachte.

In jedem Fall wird beim Verstoß einer Aktivität gegen den Schwellenwert ein Schwellenwertverstoßdatensatz in den in Schritt 1 definierten THRESHOLD VIOLATIONS-Ereignismonitor geschrieben und die Ausführung der Aktivität wird gestoppt (aufgrund der Aktion STOP EXECUTION). Die Anwendung, die die Aktivität übergeben hat, erhält einen Fehler 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
```

Weitere Informationen: Ein Schwellenwert kann entweder prädiktiv oder reaktiv sein:

- **Prädiktiver Schwellenwert:** Die Begrenzungen eines prädiktiven Schwellenwerts werden vor der Ausführung der Aktivität geprüft. Der Datenserver bezieht Nutzungsschätzungen vom Abfragecompiler, um zu prüfen, ob ein prädiktiver Schwellenwert verletzt würde. In diesem Beispiel ist der Schwellenwert `th_estcost` ein prädiktiver Schwellenwert.
- **Reaktiver Schwellenwert:** Die Begrenzungen eines reaktiven Schwellenwerts werden geprüft, während eine Aktivität ausgeführt wird. Es werden näherungsweise berechnete Nutzungsschätzungen der Laufzeit der gesteuerten Ressource verwendet, um die Begrenzungen von reaktiven Schwellenwerten auszuwerten. Die Laufzeitnutzungsschätzungen werden nicht kontinuierlich bezogen, sondern an ausgewählten vordefinierten Prüfpunkten in der Laufzeit der verfolgten Arbeit. In diesem Beispiel ist der Schwellenwert `th_sqlrows` ein reaktiver Schwellenwert.

Schritt 4: Ausführen einiger Aktivitäten

Führen Sie einige Aktivitäten aus, von denen einige gegen die im vorherigen Schritt definierte Schwellenwertobergrenze verstoßen.

```
db2 -o -tvf workth.db2
```

Beachten Sie, dass die Anweisungen, die gegen die oben definierten Schwellenwerte verstoßen, mit einem Fehler SQL4712N/SQLSTATE 5U026 fehlschlagen.

Schritt 5: Anzeigen des Ereignismonitors für Schwellenwertverstöße

Vom THRESHOLD VIOLATIONS-Ereignismonitor werden Informationen zu jedem Schwellenwertverstoß erfasst. Sie können die Schwellenwertverstoßinformationen abfragen, indem Sie reguläre SQL-Anweisungen auf der Schwellenwertverstoßmonitortabelle absetzen (siehe folgendes Beispiel).

```
CONNECT TO SAMPLE
SELECT APPL_ID,
       UOW_ID,
       ACTIVITY_ID,
       COORD_PARTITION_NUM AS COORDPART,
       THRESHOLD_PREDICATE,
       THRESHOLD_ACTION,
       TIME_OF_VIOLATION
FROM THRESHOLDVIOLATIONS_THREVI0
ORDER BY THRESHOLD_ACTION, THRESHOLD_PREDICATE, TIME_OF_VIOLATION
```

Die Ausgabe könnte folgendermaßen aussehen:

```
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 record(s) selected.
```

Schritt 6: Anzeigen von Informationen zur Aktivität, die den Schwellenwertverstoß verursachte

Es werden Aktivitätsinformationen für jede Aktivität erfasst, die gegen einen Schwellenwert verstößt, für den eine Klausel COLLECT definiert ist. Zeigen Sie die detaillierten Informationen zu den Aktivitäten an, die gegen einen Schwellenwert verstoßen haben:

```
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_THREVI0 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
```

Die Ausgabe könnte folgendermaßen aussehen:

```
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                               3
0 2007-08-31-09.01.16.000000 SELECT * FROM SALES
```

Beachten Sie, dass die Aktivität, die gegen den Schwellenwert `th_estcost` (EstimatedSqlCost) verstieß, nicht angezeigt wird. Der Grund dafür ist, dass für den Schwellenwert nicht die Klausel `COLLECT ACTIVITY DATA` definiert war, sodass für diese Aktivität keine Aktivitätsdaten erfasst wurden.

Schritt 7: Zurücksetzen für die nächste Übung

Inaktivieren Sie die aktivierten Ereignismonitore. Inaktivieren und löschen Sie auch die erstellten Schwellenwerte `th_estcost` und `th_sqlrows`.

```
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
```

Bereinigen Sie auch die Aktivitätsereignismonitortabellen und die Schwellenwertverstoßtabelle.

```
DELETE from ACTIVITY_DB2ACTIVITIES
DELETE from ACTIVITYSTMT_DB2ACTIVITIES
DELETE from THRESHOLDVIOLATIONS_THREVIEW
```

```
CALL WLM_COLLECT_STATS()
```

Übung 4: Unterscheiden von Aktivitäten nach Aktivitätstyp

Diese Übung veranschaulicht, wie ein Arbeitsaktionsset für die folgenden Zwecke verwendet werden kann: Erfassen von Informationen zu allen Aktivitäten eines bestimmten Typs; Anwenden eines Schwellenwerts auf alle Aktivitäten eines bestimmten Typs; Eingrenzen von Aktivitäten eines bestimmten Typs durch Zuordnung zu einer bestimmten Serviceunterklasse.

Geschätzte Zeit: 25-30 Minuten

Arbeitsaktionssets werden verwendet, um eine Aktion auf eine Aktivität anzuwenden, und zwar basierend auf der Art der Aktivität statt auf dem Benutzer, der sie übergeben hat (wie dies bei Workloads der Fall ist).

Aktionen können angewendet werden auf:

- Alle Datenbankaktivitäten eines bestimmten Typs (mit einem Arbeitsaktionsset für Datenbanken)
- Nur auf Aktivitäten eines bestimmten Typs in einer bestimmten Serviceklasse (mit einem Arbeitsaktionsset für Serviceklassen)

In dieser Übung werden beide Methoden gezeigt.

Weitere Informationen: Es gibt andere Aktionen, die angewendet werden können, wie z. B. das Erfassen von Statistikdaten für Aktivitäten eines bestimmten Typs, die nicht im Rahmen dieser Übung behandelt werden.

Schritt 1: Erstellen eines Arbeitsklassensets

Erstellen Sie zuerst ein Arbeitsklassenset mit Arbeitsklassen für die spezifischen Typen von Aktivitäten, an denen Sie interessiert sind. Dieses Arbeitsklassenset wird in Verbindung mit Arbeitsaktionssets verwendet, um Aktionen an den ausgewählten Aktivitätstypen durchzuführen. Im Folgenden Beispiel wird ein Arbeitsklassenset mit Arbeitsklassen aller möglichen Typen erstellt. Wenn Sie jedoch nur an einem Aktivitätstyp interessiert sind, könnte Ihr Arbeitsklassenset so erstellt werden, dass es nur eine Arbeitsklasse enthält.

```
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)
```

Schritt 2: Aktivieren des Aktivitätsereignismonitors

Aktivieren Sie den in Übung 1 erstellten Ereignismonitor für Aktivitäten.

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

Schritt 3: Erstellen eines Arbeitsaktionssets für Datenbanken

Wenn Sie eine bestimmte Aktion für alle Aktivitäten eines bestimmten Typs ausführen wollen (z. B. Anwenden eines Schwellenwerts oder Erfassen von Aktivitätsinformationen), verwenden Sie ein Arbeitsaktionsset für Datenbanken.

Erstellen Sie ein Arbeitsaktionsset auf Datenbankebene, das Arbeitsaktionen für die spezifische Arbeitsklasse enthält, die den Typ der einzugrenzenden Aktivitäten darstellt. In diesem Beispiel wollen wir Aktivitätsdaten für alle DDL-, READ- und LOAD-Aktivitäten erfassen, die auf dem System ausgeführt werden, und wir wollen auch die Ausführung aller großen READ-Aktivitäten stoppen. In dieser Übung gilt jede Anweisung SELECT mit einem geschätzten Aufwand von mehr als 10.000 Timern als große READ-Aktivität.

```
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 )
```

Schritt 4: Ausführen von Aktivitäten und Anzeigen von Statistikdaten für Arbeitsaktionssets

Führen Sie die Scripts work1.db2 und work3.db2 aus.

```
db2 -o -tvf work1.db2
db2 -o -tvf work3.db2
```

Sie können die Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS für den Zugriff auf die Statistikdaten des Arbeitsaktionssets im Speicher verwenden, um zu ermitteln, wie oft bestimmte Aktivitätstypen ausgeführt wurden. Bei der Ausführung der folgenden Abfrage werden nur die Arbeitsklassen load_wc, read_wc und ddl_wc angezeigt, da sie die einzigen Arbeitsklassen mit einer anwendbaren Arbeitsaktion sind. Alle anderen Aktivitäten werden unter "*" gezählt:

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

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 record(s) selected.

Schritt 5: Anzeigen der erfassten Aktivitätsdaten

Informationen zu den einzelnen DDL-, READ- und LOAD-Aktivitäten wurden vom Aktivitätsereignismonitor erfasst. Dies geschieht aufgrund der Angabe der Arbeitsaktion COLLECT ACTIVITY DATA, die in Schritt 3 auf die Arbeitsklassen ddl_wc, read_wc und load_wc angewendet wurde. Im Folgenden sehen Sie zwei Beispiele, wie Sie sich diese Aktivitätsinformationen ansehen können:

Wenn Sie einige grundlegende Informationen zu den Aktivitäten erhalten wollen, können Sie einfach die Tabelle des Aktivitätsereignismonitors mit einer Anweisung wie der folgenden Anweisung abfragen:

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

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 record(s) selected.

Wenn Sie zusätzliche Informationen zu jeder Aktivität, wie den Aktivitätstext und die Serviceklasse, unter der sie ausgeführt wurde, erhalten wollen, können Sie eine Abfrage ähnlich der folgenden ausführen:

```

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

```

Die Ausgabe könnte folgendermaßen aussehen:

```

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

:
:

```

Beachten Sie, dass eine der Aktivitäten einen SQLCODE von -4712 hat. Dies weist darauf hin, dass die Ausführung der Aktivität aufgrund eines Schwellenwertverstoßes gestoppt wurde. Der für die Arbeitsaktion stop_large_read_wa definierte Schwellenwert verhindert die Ausführung jeder Anweisung SELECT mit einem geschätzten Aufwand von mehr als 10.000 Timeron.

Weitere Informationen: LOAD-Aktivitäten (ohne LOAD von einem Cursor) haben keinen Eintrag in der Ereignismonitortabelle der Aktivitätsanweisung (activitystmt_db2activities), was erklärt, warum es keinen Eintrag für die einzelne Ladeaktivität, die vom Script work1.db2 ausgeführt wird, in der Ausgabe der letzten Abfrage oben gibt. Dies liegt daran, dass Ladeaktivitäten keine SQL-Anweisungen sind. Für das Laden aus Cursoraktivitäten gibt es einen Eintrag für die Cursoranweisung in der Ereignismonitortabelle der Aktivitätsanweisung, da der Cursor selbst eine separate Aktivität ist. Es gibt einen Eintrag für alle Ladeaktivitäten in der Aktivitätseignismonitortabelle (activity_db2activities).

Schritt 6: Inaktivieren von Arbeitsaktionen

Bevor Sie mit dem Arbeitsaktionsset für Serviceklassen fortfahren, löschen Sie das Arbeitsaktionsset für Datenbanken.

```
DROP WORK ACTION SET db_was
```

Weitere Informationen: Vor dem Löschen eines Schwellenwerts für gemeinsamen Zugriff muss dieser Schwellenwert zuerst inaktiviert werden. In diesem Fall gibt es keine Arbeitsaktionen, die einen Schwellenwert für gemeinsamen Zugriff darstellen. Wenn dies jedoch der Fall wäre, könnte er nur durch Inaktivieren der Arbeitsaktion inaktiviert werden. Ein Arbeitsaktionsschwellenwert kann nicht über SQL-Anweisungen für Schwellenwerte (THRESHOLD) manipuliert werden, sondern nur über SQL-Anweisungen für Arbeitsaktionssets (WORK ACTION SET). Nur Arbeitsaktionen, die Schwellenwerte für gemeinsamen Zugriff darstellen, müssen vor dem Löschen des nachfolgenden Arbeitsaktionssets inaktiviert werden. Da es bei dieser Übung keine Arbeitsaktionen gibt, die einen Schwellenwert für gemeinsamen Zugriff darstellen, müssen vor dem Löschen des Arbeitsaktionssets keine Arbeitsaktionen inaktiviert werden.

Wenn Sie eine bestimmte Aktion, wie z. B. einen Schwellenwert, auf alle Aktivitäten eines bestimmten Typs in einer Servicesuperklasse anwenden wollen, sollten Sie die Verwendung eines Arbeitsaktionssets für Serviceklassen in Erwägung ziehen. Sie können eine Zuordnungsarbeitsaktion erstellen, um bestimmte Typen von Aktivitäten einer bestimmten Serviceunterklasse zuzuordnen, und dann einen Schwellenwert auf diese Serviceunterklasse anwenden. Die folgenden Schritte zeigen, wie Arbeitsaktionssets für Serviceklassen verwendet werden können.

Schritt 7: Erstellen einer Serviceklasse und einer Workload

Erstellen Sie eine Serviceunterklasse unter der Servicesuperklasse `work1_sc`, die in Übung 2 in Schritt 2 erstellt wurde.

Die Servicesuperklasse `work1_sc` ist die Serviceklasse, der über die Workloads Aktivitäten zugeordnet werden. Die Serviceunterklasse `work1_sc_read` ist die Serviceklasse, der über die Arbeitsaktion Leseaktivitäten zugeordnet werden.

```
CREATE SERVICE CLASS work1_sc_read UNDER work1_sc
```

Erstellen Sie eine Workload, sodass alle vom Script `work3.db2` übergebenen Aktivitäten der Servicesuperklasse `work1_sc` zugeordnet werden. Aktivitäten von `work1.db2` werden bereits wegen einer der früheren Übungen der Serviceklasse `work1_sc` zugeordnet.

```
CREATE WORKLOAD work3_w1 CURRENT CLIENT_APPLNAME('CLP work3.db2') SERVICE CLASS work1_sc
```

Schritt 8: Erstellen eines Arbeitsaktionssets für Serviceklassen

Erstellen Sie ein Arbeitsaktionsset auf Serviceklassenebene, das Arbeitsaktionen für die spezifischen Arbeitsklassen enthält, die den Typ der einzugrenzenden Aktivitäten darstellen. In diesem Beispiel wollen wir Aktivitätsdaten für alle DDL-, READ- und LOAD-Aktivitäten erfassen, die unter der Serviceklasse `work1_sc` ausgeführt werden, und wir wollen auch READ-Aktivitäten einer separaten Serviceunterklasse zuordnen, sodass wir sie auf andere Weise behandeln können; in diesem Fall wird ein Schwellenwert auf die Serviceunterklasse angewendet, um die Ausführung aller großen SELECT-Anweisungen zu stoppen.

```
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)
```

Schritt 9: Erstellen eines Serviceklassenschwellenwerts

Damit Sie einen Effekt ähnlich der Arbeitsaktion `stop_large_read_wa` erhalten, die die Ausführung aller großen SELECT-Anweisungen verhinderte, erstellen Sie einen Schwellenwert vom Typ `ESTIMATEDSQLCOST` und wenden Sie ihn auf die Serviceunterklasse `work1_sc_read` an.

```
CREATE THRESHOLD stop_large_activities FOR SERVICE CLASS work1_sc_read
  UNDER work1_sc
  ACTIVITIES ENFORCEMENT DATABASE
  WHEN ESTIMATEDSQLCOST >10000 STOP EXECUTION
```

Schritt 10: Löschen der Aktivitätstabellen, Zurücksetzen der Statistikdaten und Ausführen von Aktivitäten

Löschen Sie den Inhalt aller Aktivitätstabellen, sodass Sie von vorn beginnen können, bevor Sie das Script erneut ausführen. Rufen Sie dann die gespeicherte Prozedur `wlm_collect_stats()` auf, um die Statistikdaten zurückzusetzen.

```
DELETE FROM activity_db2activities
DELETE FROM activitystmt_db2activities
DELETE FROM activityvals_db2activities

CALL wlm_collect_stats()
```

Führen Sie jetzt die Scripts work1.db2 und work3.db2 einmal aus.

```
db2 -o -tvf work1.db2
db2 -o -tvf work3.db2
```

Beachten Sie den Fehler SQL04712 für Aktivitäten, die einen Schwellenwertverstoß verursachten.

Schritt 11: Anzeigen von Statistikdaten für Arbeitsaktionssets

Verwenden Sie die Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS für den Zugriff auf die Statistikdaten des Arbeitsaktionssets im Speicher, um zu ermitteln, wie oft bestimmte Aktivitätstypen ausgeführt wurden. Bei der Ausführung der folgenden Abfrage werden nur die Arbeitsklassen load_wc, ddl_wc und read_wc angezeigt, da sie die einzigen drei Arbeitsklassen mit einer anwendbaren Arbeitsaktion sind. Alle anderen Aktivitäten werden unter "*" gezählt:

```
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
```

Die Ausgabe könnte dieses Mal folgendermaßen aussehen:

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 record(s) selected.

Schritt 12: Anzeigen der erfassten Aktivitätsdaten

Fragen Sie jetzt die Aktivitätstabellen erneut ab, um Informationen zu den einzelnen Aktivitäten zu erhalten. Beachten Sie die Serviceunterklasse, unter der die Aktivitäten ausgeführt wurden.

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

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_applname)	
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, syscat.tables, syscat.tables, syscat.tables, syscat.tables	

Beachten Sie wieder, dass eine der Aktivitäten einen SQLCODE von -4712 hat, dieses Mal aufgrund des in Schritt 9 erstellten Schwellenwerts für Serviceklassen stop_large_activities, gegen den verstoßen wurde, da der geschätzte Aufwand für die Anweisung SELECT zu groß war. Beachten Sie auch, dass alle Leseaktivitäten unter der Serviceunterklasse work1_sc_read ausgeführt werden.

Schritt 13: Zurücksetzen für die nächste Übung

Inaktivieren Sie den Ereignismonitor, löschen Sie den Serviceklassenschwellenwert und löschen Sie das Arbeitsaktionsset für Serviceklassen.

```
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
```

Löschen Sie den Inhalt aller Aktivitätstabellen, sodass Sie von vorn beginnen können, bevor Sie das Script erneut ausführen.

```
DELETE FROM activity_db2activities
DELETE FROM activitystmt_db2activities
DELETE FROM activityvals_db2activities
```

Inaktivieren Sie alle erstellten Workloads, sodass alle Aktivitäten unter der Standardbenutzerworkload ausgeführt und der Standardservicesuperklasse zugeordnet werden.

```
ALTER WORKLOAD work1_w1 DISABLE
ALTER WORKLOAD work2_w1 DISABLE
ALTER WORKLOAD work3_w1 DISABLE
ALTER WORKLOAD workth_w1 DISABLE
```

Rufen Sie die gespeicherte Prozedur wlm_collect_stats() auf, um die Statistikdaten zurückzusetzen.

```
CALL WLM_COLLECT_STATS()
```

Übung 5: Verwenden von Histogrammen für Serviceklassen

Diese Übung veranschaulicht, wie die Option COLLECT AGGREGATE ACTIVITY DATA BASE für eine Serviceklasse verwendet werden kann, um Histogramme der Laufzeiten, der Ausführungszeiten und der Wartezeiten von Koordinatoraktivitäten erzeugen zu können.

Geschätzte Zeit: 25-30 Minuten

Diese drei Histogramme sind nützlich, um mehr als nur die durchschnittliche Laufzeit, Ausführungszeit oder Wartezeit der auf dem System ausgeführten Aktivitäten zu erfahren, da sie zur Berechnung von Standardabweichungen verwendet werden können und somit Sonderfälle erkennen können. Weitere Informationen zu Histogrammen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

Auf Histogramme greifen Sie über den Statistikereignismonitor zu. In dieser Übung wird der in Übung 1 in Schritt 1 erzeugte Statistikereignismonitor verwendet.

Weitere Informationen: Der Statistikereignismonitor ist ein WRITE-TO-TABLE-Ereignismonitor und enthält logische Datengruppen. Die erste ist die logische Steuerungsdatengruppe, die jeder Ereignismonitor hat. Außerdem gibt es die logischen Datengruppen, die für den jeweiligen Statistikereignismonitortyp spezifisch sind. Es gibt die folgenden spezifischen logischen Datengruppen:

- histgrambin für Histogramminformationen
- qstats für Statistikdaten von Schwellenwertwarteschlangen
- scstats für Serviceklassenstatistikdaten
- wcstats für Arbeitsklassenstatistikdaten
- wlstats für Workloadstatistikdaten

Schritt 1: Erstellen von Sichten zum Anzeigen von Histogrammstatistikdaten

Erstellen Sie verschiedene Sichten, um das Abfragen der Tabelle HISTOGRAMBIN_DB2STATISTICS zu vereinfachen. Die erste Sicht enthält alle verfügbaren Histogrammtypen. In dieser Übung werden nur die drei Grundtypen dokumentiert: Laufzeit, Ausführungszeit und Wartezeit.

```
CREATE VIEW HISTOGRAMTYPES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) AS HISTOGRAM_TYPE
  FROM HISTOGRAMBIN_DB2STATISTICS
```

Eine zweite Sicht macht es einfacher, herauszufinden, für welche Serviceklassen Histogramme erfasst werden. Die Tabelle HISTOGRAMBIN_DB2STATISTICS gibt die Serviceklassen, für die Histogramme erfasst werden, mit der Serviceklassen-ID an. Durch die Verknüpfung dieser Tabelle mit der Katalogtabelle SERVICECLASSES können die Serviceklasseninformationen mit dem Servicesuperklassennamen und dem Serviceunterklassennamen statt mit der Serviceklassen-ID dargestellt werden.

```
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
```

Die dritte Sicht enthält alle Zeiten, zu denen ein Histogramm des angegebenen Typs für eine angegebene Serviceklasse erfasst wurde. Wie die Sicht histogramserviceclasses verknüpft sie auch die Tabelle HISTOGRAMBIN_DB2STATISTICS mit der Katalogtabelle SERVICECLASSES. Der Unterschied liegt darin, dass die Spalte STATISTICS_TIMESTAMP als eine der Spalten in der Sicht enthalten ist.

```
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
```

Die vierte und letzte Sicht enthält die Histogramme. Sie zeigt auch etwas, das oft im Zusammenhang mit Histogrammen erforderlich ist, nämlich die Zusammenfassung im Zeitverlauf. Diese Sicht zeigt den Anfang jedes Bins und die Anzahl der Aktivitäten, die für jeden Bin gezählt wurden. Für die drei Histogramme in dieser Übung misst das Feld BIN_TOP die Laufzeit, Ausführungszeit oder Wartezeit der Aktivität in Millisekunden. Wenn bei einem Laufzeithistogramm BIN_TOP z. B. 3000 Millisekunden und der BIN_TOP-Wert des vorherigen Bins 2000 Millisekunden beträgt und der Wert von NUMBER_IN_BIN 10 ist, dann wissen Sie, dass zehn Aktivitäten eine Laufzeit zwischen 2 und 3 Sekunden hatten.

```
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
```

Schritt 2: Aktivieren der Erfassung von Histogrammen

Die Histogramme der Laufzeit, Wartezeit und Ausführungszeit von Aktivitäten werden für eine Serviceunterklasse erfasst, wenn die Option COLLECT AGGREGATE ACTIVITY DATA BASE für die Unterklasse aktiviert ist. Aktivieren Sie die Basiserfassung der zusammengefassten Aktivitätsdaten für die Standardunterklasse unter der Standardbenutzerservicesuperklasse mit der Klausel COLLECT AGGREGATE ACTIVITY DATA.

Beachten Sie, dass alle Aktivitäten in der Standardbenutzerserviceklasse ausgeführt werden, da alle benutzerdefinierten Workloads am Ende der vorherigen Übung inaktiviert wurden.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS
  UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA BASE
```

Schritt 3: Aktivieren des Statistikereignismonitors

Aktivieren Sie den bereits erstellten Statistikereignismonitor, sodass er die Zusammenfassungsdaten bei deren Erfassung empfangen kann.

```
SET EVENT MONITOR DB2STATISTICS STATE 1
```

Schritt 4: Ausführen von Aktivitäten und Senden von Statistikdaten an den Statistikereignismonitor

Jetzt können einige Aktivitäten ausgeführt werden. Nachdem die Aktivitäten abgeschlossen sind, wird die gespeicherte Prozedur WLM_COLLECT_STATS aufgerufen, um die Statistikdaten (einschließlich der Histogramme der Aktivitätslaufzeit, -ausführungszeit und -wartezeit für die Standardbenutzerserviceklasse) an den aktiven Statistikereignismonitor zu senden. Diese Histogramme enthalten Daten zu allen Aktivitäten, die seit der Aktivierung von zusammengefassten Aktivitätsstatistikdaten in der Standardbenutzerserviceklasse ausgeführt wurden. Durch den Aufruf dieser gespeicherten Prozedur werden auch die Statistikdaten zurückgesetzt. Zum Anzeigen von Änderungen der Datenbankaktivität im Lauf der Zeit werden drei Erfassungsintervalle erstellt. Führen Sie im ersten Intervall zwei Scripts, work1.db2 und work2.db2, aus und lassen Sie dann die Statistikdaten erfassen und zurücksetzen.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS()
```

Führen Sie im zweiten Intervall nur das Script work1.db2 einmal aus und lassen Sie dann die Statistikdaten erfassen und zurücksetzen.

```
db2 -o -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS()
```

Führen Sie im dritten Intervall das Script work1.db2 zweimal und das Script work2.db2 einmal aus und lassen Sie dann die Statistikdaten erfassen und zurücksetzen.

```
db2 -o -tvf work1.db2
db2 -o -tvf work2.db2
db2 -o -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS()
```

Durch das periodische Erfassen von Daten können Sie überwachen, wie sich die Arbeit auf Ihrem System im Lauf der Zeit ändert.

Zusätzliche Informationen: Das periodische Erfassen von Daten muss nicht manuell erfolgen. Mit dem Datenbankkonfigurationsparameter WLM_COLLECT_INT kann das Intervall (in Minuten) festgelegt werden, in dem die Statistikerfassung und -zurücksetzung automatisch erfolgt.

Schritt 5: Abfragen von Sichten zum Anzeigen von Statistikdaten

Nachdem Statistikdaten erfasst wurden, können die zuvor erstellten Sichten zum Anzeigen der Statistikdaten verwendet werden. Die Sicht HISTOGRAMTYPES gibt nur die Typen der verfügbaren Histogramme zurück.

```
SELECT * FROM HISTOGRAMTYPES
```

```
HISTOGRAM_TYPE
```

```
-----
```

```
CoordActExecTime
CoordActLifetime
CoordActQueueTime
```

3 record(s) selected.

Da beim Ändern der Serviceklasse die Option BASE verwendet wurde, gibt es drei Histogramme: Laufzeit, Ausführungszeit und Wartezeit. In der Sicht HISTOGRAMSERVICECLASSES werden die Serviceklassen angezeigt, für die ein Histogramm erfasst wurde. Das folgende Beispiel schränkt die Ausgabe auf das Histogramm CoordActLifetime ein. Da die Erfassung zusammengefasster Aktivitätsinformationen nur für die Standardunterklasse der Standardbenutzerserviceklasse aktiviert wurde, wird bei der Auswahl aus der Sicht HISTOGRAMSERVICECLASSES nur diese Klasse angezeigt.

```
SELECT * FROM HISTOGRAMSERVICECLASSES
WHERE HISTOGRAM_TYPE = 'CoordActLifetime'
ORDER BY SERVICE_SUPERCLASS, SERVICE_SUBCLASS
```

HISTOGRAM_TYPE	SERVICE_SUPERCLASS	SERVICE_SUBCLASS
CoordActLifetime	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS

1 record(s) selected.

Die Sicht HISTOGRAMTIMES enthält die Zeiten, zu denen Histogramme erfasst wurden. Da die Prozedur WLM_COLLECT_STATS dreimal ausgeführt wurde, gibt es im Laufzeithistogramm drei Zeitmarken.

```
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 record(s) selected.

Die letzte Sicht HISTOGRAMS enthält die Histogramme. Im Gegensatz zur Sicht HISTOGRAMTIMES, die jedes Erfassungsintervall als eigene Zeile auflistet, fasst die Sicht HISTOGRAMS Histogrammdaten über mehrere Intervalle zusammen, um ein einzelnes Histogramm des angegebenen Typs für eine angegebene Serviceklasse zu erstellen.

```
SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
WHERE HISTOGRAM_TYPE = 'CoordActLifetime'
AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
ORDER BY BIN_TOP
```

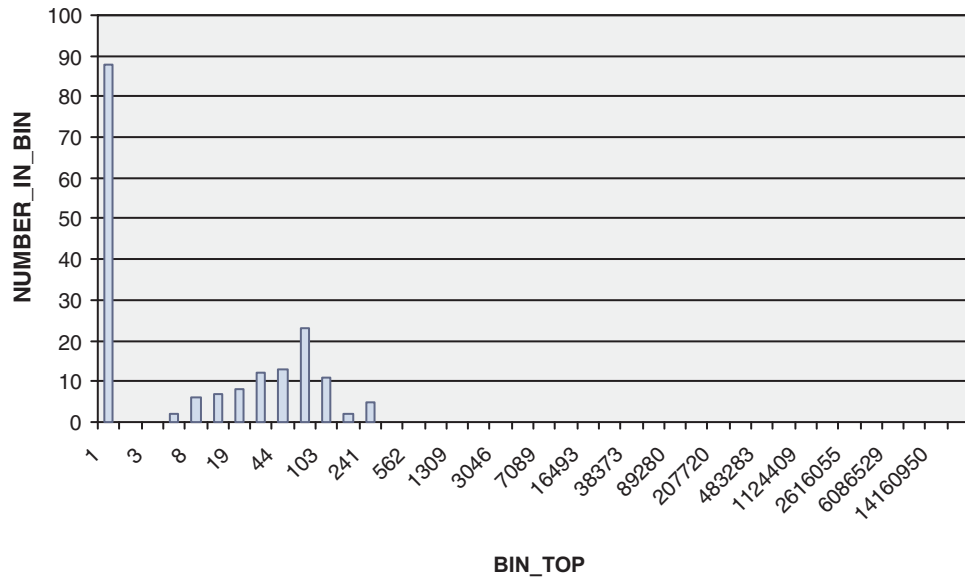
BIN_TOP	NUMBER_IN_BIN
-1	0
1	88
2	0

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 record(s) selected.

Die Ausgabe aus den Histogrammen kann als Eingabe in ein Diagrammtool zur Erzeugung eines Diagramms verwendet werden. Das folgende Diagramm wurde mit einer Ruby Graphing-Bibliothek namens Gruff Graphs erstellt.

Laufzeithistogramm für SYSDEFAULTUSERCLASS (CoordActLifetime):



Die Ausführung der Abfrage dieser Sicht sollte eine Ausgabe erzeugen, die nicht exakt dem oben abgebildeten Diagramm des Laufzeithistogramms entspricht, da die Aktivitätslaufzeiten von der Leistung des Systems abhängen. Die obige Ausgabe enthält 41 Bins, wobei alle größeren Bins leer sind. Am Anfang steht ein Bin, bei dem BIN_TOP -1 ist. Dieser Bin stellt alle Aktivitäten dar, deren Laufzeit zu groß für das Histogramm war. Wenn der Wert für NUMBER_OF_BIN größer als null und der Wert für BIN_TOP -1 ist, sollten Sie den oberen Binwert Ihres Histogramms hochsetzen. In der obigen Ausgabe ist NUMBER_IN_BIN 0, sodass hier keine solche Änderung erforderlich ist. Eine große Anzahl von Aktivitäten, in diesem Fall 88, wurden im Bin mit einem BIN_TOP-Wert von 1 gezählt. Dies ist der niedrigste Bin. Dies bedeutet, dass 88 Aktivitäten eine Laufzeit zwischen 0 und 1 Millisekunde hatten. Außerdem zeigt das Histogramm, dass die längste Laufzeit jeglicher Aktivität in den in diesem Histogramm erfassten Workloads zwischen 158 und 241 Millisekunden lag, da der größte BIN_TOP, für den Werte enthalten sind, 241 beträgt. Die Spalte COORD_ACT_LIFETIME_TOP in der Tabelle SCSTATS_DB2STATISTICS gibt eine genauere Messung der Laufzeit der Aktivität mit der längsten Laufzeit.

Dieselbe Abfrage kann mit einem Histogrammtyp von CoordActExecTime statt CoordActLifetime wiederholt werden. Das Histogramm der Ausführungszeit sollte ähnlich wie das Laufzeithistogramm sein, jedoch nicht identisch. Die Histogramme sind unterschiedlich, selbst wenn keine Warteschlange verwendet wird, weil die Ausführungszeit nicht die Initialisierungszeit oder die Cursorleerlaufzeit einschließt, die Laufzeit hingegen schon.

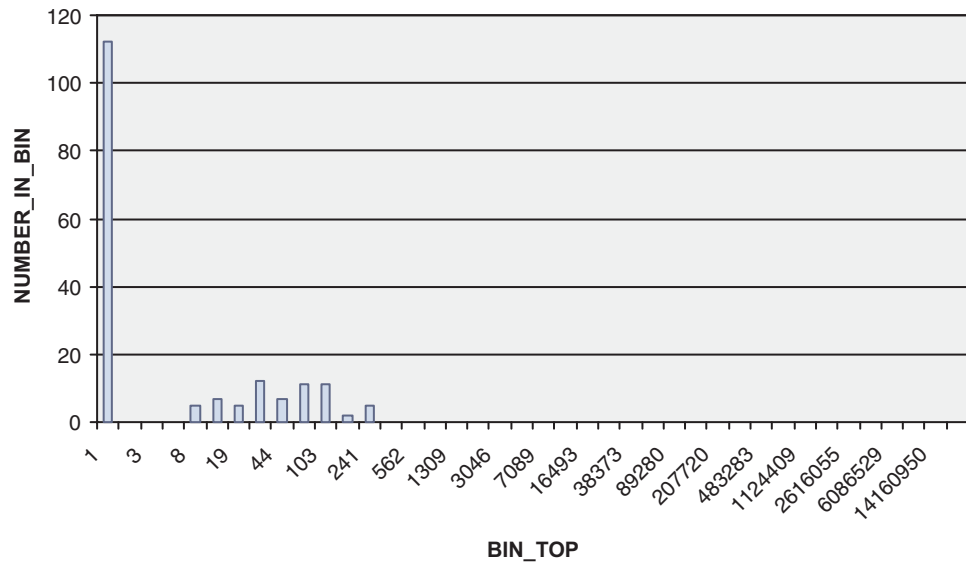
```
SELECT BIN_TOP, NUMBER_IN_BIN FROM HISTOGRAMS
  WHERE HISTOGRAM_TYPE = 'CoordActExecTime'
    AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
    AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
  ORDER BY BIN_TOP
```

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 record(s) selected.

Histogramm der Ausführungszeit für SYSDEFAULTUSERCLASS (CoordActExecTime):



Wieder wird eine große Anzahl von Aktivitäten im ersten Bin gezählt und die längste Ausführungszeit einer Aktivität beträgt höchstens 241 Millisekunden.

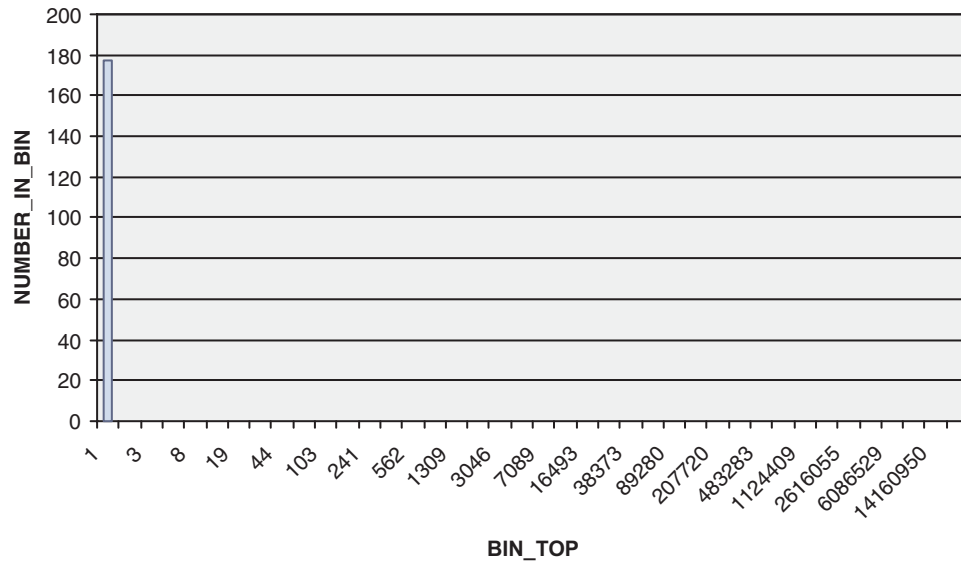
Schließlich wird das Histogramm CoordActQueueTime mit der Sicht HISTOGRAMS angezeigt. Dies ist das einfachste Histogramm, da keine Warteschlangenbildung auftritt, denn es wurden in dieser Übung keine Warteschlangenschwellenwerte erstellt oder aktiviert.

```
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 record(s) selected.

Histogramm der Wartezeit für SYSDEFAULTUSERCLASS (CoordActQueueTime):



Jede Aktivität wurde im Bin von 0 bis 1 Millisekunde gezählt, da jede Aktivität 0 Millisekunden in der Warteschlange wartete.

Die letzten Abfragen befassten sich mit den Aktivitätslaufzeiten, -ausführungszeiten und -wartezeiten, aufgeschlüsselt in Bins, jedoch zusammengefasst über mehrere Intervalle. Die folgende Abfrage stellt dieselben Informationen aus einer anderen Perspektive dar. Sie zeigt Durchschnittswerte anstelle von Histogrammen. Dabei werden die Intervalle getrennt und nicht zusammengefasst dargestellt. Außerdem wird die Anzahl abgeschlossener Aktivitäten dokumentiert, was zeigt, wie viele Aktivitäten in jedem Intervall abgeschlossen wurden. Hier wird die Tabelle SCSTATS_DB2STATISTICS statt der Tabelle HISTOGRAMBIN_DB2STATISTICS verwendet.

```
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 record(s) selected.

Das Ergebnis zeigt, dass die durchschnittlichen Laufzeiten für jedes Intervall etwas höher sind als die durchschnittlichen Ausführungszeiten und dass alle drei maximal etwas über einer halben Sekunde liegen. Die durchschnittliche Wartezeit ist erwartungsgemäß null. Die Anzahl abgeschlossener Aktivitäten in jedem Intervall ist wie erwartet, da im ersten Intervall die Workloads 1 und 2 ausgeführt wurden, was 77 erfasste Aktivitäten ergab, im zweiten Intervall Workload 1 allein ausge-

führt wurde, was 39 Aktivitäten ergab, und im dritten Intervall Workload 1 zweimal und Workload 2 einmal ausgeführt wurde, was 113 Aktivitäten ergab.

Schritt 6: Zurücksetzen für die nächste Übung

Als letzten Schritt müssen Sie die Erfassung von zusammengefassten Aktivitäten in der Standardbenutzerserviceklasse inaktivieren, die Sichten löschen und die Informationen in den Statistiktabellen löschen.

```
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
```

Übung 6: Untersuchen von Verzögerungen mit WLM-Tabellenfunktionen

Diese Übung veranschaulicht, wie Sie die Ursache einer Anwendungsverzögerung mit den DB2-WLM-Überwachungsfunktionen ermitteln können.

Geschätzte Zeit: 10-15 Minuten

Die DB2-WLM-Überwachungsfunktionen stellen Informationen und Statistikdaten für die Arbeit in einer Datenbank zur Verfügung. Sobald die Ursache einer Verzögerung ermittelt ist, können Sie Abhilfe schaffen.

Schritt 1: Ausführen von Aktivitäten

In dieser Übung werden zwei Anwendungen verwendet, app1.db2 und app2.db2. Beide Anwendungen führen DML-Operationen in der Datenbank SAMPLE aus. Führen Sie das Script app1.db2 in einem Fenster, sofort gefolgt vom Script app2.db2 in einem zweiten Fenster aus.

```
db2 -tvf app1.db2
db2 -tvf app2.db2
```

Schritt 2: Anzeigen der zurzeit aktiven Workloadvorkommen

Das Script app2.db2 sollte nun blockiert sein. Setzen Sie in einem dritten Fenster die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES ab, um die Status aller in der Datenbank ausgeführten Anwendungen herauszufinden. Für die Zwecke dieses Beispiels können Sie ein Workloadvorkommen als eine Anwendung betrachten. Diese Tabellenfunktion zeigt Informationen zu allen Workloadvorkommen in einer Serviceklasse an. Da wir alle Workloadvorkommen in der Datenbank sehen sollen, verwenden wir Platzhalterzeichen dargestellt durch " " für die Eingabeparameter **servicesuperklassenname** und **serviceunterklassenname**.

```
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))

```

Die Ausgabe könnte folgendermaßen aussehen:

APPL_HANDLE	APPL_NAME	USER_ID
12	CLP app1.db2	DB2USR1
17	CLP app2.db2	DB2USR1
18	-	DB2USR1
19	-	DB2USR1

4 record(s) selected.

Anhand der Ausgabe können wir feststellen, dass app2.db2 die Anwendungskennung 17 hat.

Schritt 3: Ermitteln des Agenten für die Anwendung

Verwenden Sie die Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS, um herauszufinden, was die Agenten für app2.db2 tun. Diese Tabellenfunktion zeigt Informationen zu Agenten an, die in einer Serviceklasse arbeiten. Da wir die Agenten sehen wollen, die für die Anwendungskennung 17 arbeiten, geben wir dies im Eingabeparameter **anwendungskennung** an. In diesem Beispiel sind wir nicht an Agenten für eine bestimmte Serviceklasse interessiert. Deshalb geben wir Platzhalterzeichen für die Eingabeparameter **servicesuperklassenname** und **serviceunterklassenname** an.

```

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))

```

Die Ausgabe könnte folgendermaßen aussehen:

APPL_HANDLE	UOW_ID	ACTIVITY_ID	AGENT_TYPE	AGENT_STATE	EVENT_TYPE
17	1	2	COORDINATOR	ACTIVE	ACQUIRE
LOCK	IDLE				

1 record(s) selected.

Anhand der Ausgabe können Sie sehen, dass der Koordinatoragent für Anwendung 17 inaktiv ist und darauf wartet, eine Sperre zu setzen. Dies ist die Ursache, warum app2.db2 blockiert zu sein scheint.

Schritt 4: Ermitteln der problematischen Anwendung und Beheben des Problems

Jetzt, da wir wissen, warum die Anwendung blockiert ist, können wir Abhilfe schaffen. Wir wissen, dass die Anwendung aufgrund einer Sperre wartet. Um herauszufinden, auf welche Sperre diese Anwendung wartet und welche Anwendung die Sperre hält, können wir das Tool db2pd verwenden. Zuerst müssen wir die aktuelle Transaktionsnummer für unsere blockierte Anwendung herausfinden: Setzen Sie 'db2pd -transactions' für Anwendungskennung 17 ab.


```
db2pd -db sample -transactions app=17
```

Die Ausgabe könnte folgendermaßen aussehen:

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		

Anhand der Ausgabe können wir feststellen, dass Anwendung 17 die Transaktionskennung 7 hat. Wir können jetzt ermitteln, auf welche Sperren diese Transaktion wartet, indem wir den Befehl 'db2pd -locks' für die Transaktionskennung 7 absetzen.

```
db2pd -db sample -locks 7 wait
```

Die Ausgabe könnte folgendermaßen aussehen:

Address	TranHdl	Lockname	Type	Mode	Sts
Owner	Dur	HoldCount	Att	ReleaseFlg	
0x07000000304013F0	7	00020010000000000640002D52	Row	.NS	W
2	1	0	0x00	0x00000002	

Die Ausgabe zeigt, dass die Anwendung aufgrund einer Zeilensperre wartet. Der Sperreneigentümer hat die Transaktionskennung 2. Diese Transaktion hält die Sperre und verursacht unsere Blockierung. Als letzter Schritt wird die entsprechende Anwendungskennung für die Transaktionskennung 2 ermittelt. Setzen Sie den Befehl 'db2pd -transactions' für die Transaktionskennung 2 ab.

```
db2pd -db sample -transactions 2
```

Die Ausgabe könnte folgendermaßen aussehen:

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		

Anhand der Ausgabe können wir sehen, dass die Transaktionskennung 2 der Anwendungskennung 12 entspricht. Anhand der Ergebnisse aus der Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES können Sie sehen, dass Anwendung 12 sich auf app1.db2 bezieht. Diese Anwendung hält eine Zeilensperre, die für app2.db2 erforderlich ist. Damit app2.db2 fortfahren kann, können Sie die UOW (Unit of Work) oder den Prozess über das Fenster, in dem app1.db2 ausgeführt wird, festschreiben, rückgängig machen oder beenden. Alternativ dazu können Sie app1.db2 zwangsweise abmelden, indem Sie den Befehl FORCE APPLICATION für die Anwendungskennung 12 absetzen.

```
db2 force application (12)
```

Weitere Informationen: Eine weitere Möglichkeit, aufgrund eines Zugriffskonflikts blockierte Anwendungen zu diagnostizieren, ist das Verwenden der Tabellenfunktion SNAPSHOT_LOCKWAIT des Überwachungsprogramms. Diese Tabellenfunktion stellt Informationen zu Sperreninhabern und wartenden Threads bereit. Damit Sie diese Tabellenfunktion verwenden können, muss der Konfigurationsparameter für den Monitorschalter DFT_MON_LOCK aktiviert sein, bevor die Datenbank gestartet wird. Dieser Schalter wirkt sich auf alle Datenbanken einer Instanz aus.

Übung 7: Abbrechen einer laufenden Aktivität

Diese Übung zeigt, wie eine laufende Aktivität mit der Prozedur WLM_CANCEL_ACTIVITY abgebrochen werden kann.

Geschätzte Zeit: 5-10 Minuten

Schritt 1: Absetzen einer lange laufenden Abfrage

Führen Sie in einem Befehlszeilenprozessorfenster das folgende Script aus, das eine lang laufende Abfrage absetzt:

```
db2 -tvf longquery.db2
```

Schritt 2: Abrufen der Anwendungskennung

Rufen Sie aus einem anderen Befehlszeilenprozessorfenster WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES auf, um die Anwendungskennung, die UOW-ID und die Aktivitäts-ID der Cursoraktivität abzurufen.

```
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')
```

Durch Verknüpfen des Ergebnisses der Tabellenfunktion mit der Verwaltungssicht APPLICATIONS können Sie die Cursoraktivität finden, die von longquery.db2 ausgeführt wird. Die Ausgabe könnte folgendermaßen aussehen:

```
APPLICATION_HANDLE  UOW_ID  ACTIVITY_ID  ACTIVITY_TYPE
-----
```

```
-----
```

```
267              1              1  READ_DML
```

```
1 record(s) selected.
```

Schritt 3: Abbrechen der Aktivität

Rufen Sie im selben Befehlszeilenprozessorfenster die gespeicherte Prozedur WLM_CANCEL_ACTIVITY auf, um mit der im vorherigen Schritt ermittelten Anwendungskennung, UOW-ID und Aktivitäts-ID die zuvor erhaltene Cursoraktivität abzubrechen:

```
CONNECT TO SAMPLE

CALL WLM_CANCEL_ACTIVITY (267, 1, 1)

CONNECT RESET
```

Beachten Sie, dass in Ihrem Fall die Anwendungskennung, UOW-ID und Aktivitäts-ID anders lauten können.

Im ersten Befehlszeilenprozessorfenster sehen Sie die folgende Ausgabe, die von der mit longquery.db2 abgesetzten lang laufenden Abfrage zurückgegeben wurde. SQL4725N Die Aktivität wurde abgebrochen. SQLSTATE=57014

Übung 8: Ermitteln der auf dem System aktiven Aktivitäten

Diese Übung veranschaulicht, wie Sie mithilfe der Überwachungstabellenfunktionen und Arbeitsaktionssets des DB2-Workload-Managements ermitteln können, welche Typen von Aktivitäten auf Ihrem System aktiv sind.

Geschätzte Zeit: 15-20 Minuten

Sie wollen beispielsweise wissen, wie viele große Aktivitäten oder Ladedienstprogramme zurzeit gleichzeitig auf Ihrem System ausgeführt werden. Zu wissen, welche Typen von Arbeit auf dem System ausgeführt werden, ist wichtig, da unterschiedliche Typen von Arbeit unterschiedlichen Ressourcenbedarf haben und sich unterschiedlich auf die Systemleistung auswirken werden.

Schritt 1: Ermitteln der Anzahl von Aktivitäten jedes Typs, die auf Ihrem System ausgeführt werden

Bevor Sie beginnen, wollen Sie eventuell die Anzahl der zurzeit ausgeführten Aktivitäten eines bestimmten Typs mit der Tabellenfunktion `WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES` anzeigen:

```
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
```

Die Ausgabe dieser Abfrage könnte folgendermaßen aussehen:

ACTIVITY_TYPE	NUMBER_RUNNING
-----	-----
READ_DML	1

Damit Sie Informationen zu den unterschiedlichen Typen von Aktivitäten erhalten, die in einem bestimmten Zeitraum auf Ihrem System ausgeführt wurden, können Sie Arbeitsklassensets und Arbeitsaktionen verwenden.

Schritt 2: Erstellen eines Arbeitsaktionssets für Datenbanken mit Arbeitsaktionen zum Zählen der Aktivitäten

Zum Zählen, wie oft eine Aktivität eines bestimmten Typs in einem bestimmten Zeitraum ausgeführt wurde, muss ein Arbeitsaktionsset erstellt werden. Da wir in diesem Beispiel an den Aktivitäten interessiert sind, die auf dem gesamten System ausgeführt werden, wird das Arbeitsaktionsset auf Datenbankebene erstellt und dem Arbeitsklassenset `all_class_types` zugeordnet, das in Übung 4 in Schritt 1 erstellt wurde. Dieses Arbeitsklassenset enthält Arbeitsklassen für alle Typen erkannter Aktivitäten. Wären wir nur an den Aktivitäten in einer bestimmten Serviceklasse interessiert, würden wir ein Arbeitsaktionsset auf Serviceklassenebene erstellen. In diesem Beispiel sind wir auch an den Informationen zu allen Typen von Aktivitäten interessiert, sodass das Arbeitsaktionsset eine Arbeitsaktion `COUNT ACTIVITY` für jede Arbeitsklasse im Arbeitsklassenset `all_class_types` enthält.

```
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)

```

Weitere Informationen: Jedes Mal, wenn auf eine Aktivität, die einer Arbeitsklasse entspricht, mindestens eine Arbeitsaktion angewendet wird, wird ein Zähler für die Arbeitsklasse um 1 erhöht. Die Arbeitsaktion COUNT ACTIVITY ist eine Möglichkeit, mit der effektiv sichergestellt werden kann, dass der Zähler aktualisiert wird. Wenn Sie außer der Zählung der ausgeführten Aktivitäten dieses Typs keine andere Aktion durchführen wollen, ist die Methode mit der Arbeitsaktion COUNT ACTIVITY am besten geeignet.

Schritt 3: Ausführen einiger Aktivitäten

Führen Sie das Script work1.db2 ein Mal aus.

```
db2 -tvf work1.db2
```

Schritt 4: Anzeigen von Statistikdaten für Arbeitsaktionssets

Sie können die Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS für den Zugriff auf die Statistikdaten des Arbeitsaktionssets im Speicher verwenden, um zu ermitteln, wie oft bestimmte Aktivitätstypen ausgeführt wurden. Die folgende Abfrage gibt z. B. die Anzahl der Aktivitäten zurück, die jeder Arbeitsklasse im Arbeitsklassenset zugeordnet wurden, dem eine Arbeitsaktion zugeordnet ist:

```
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, LAST_RESET

```

Weitere Informationen: Das Leerzeichen in der Anweisung bedeutet, dass das Ergebnis nicht durch das Argument eingeschränkt ist. (In diesem Beispiel wollen wir die Informationen für alle Arbeitsaktionssets). Der Wert des letzten Arguments, *member*, ist das Platzhalterzeichen -2, was bedeutet, dass Daten von allen Datenbankmitgliedern zurückgegeben werden.

Die Ausgabe dieser Abfrage könnte folgendermaßen aussehen, wobei "*" alle Aktivitäten darstellt, die in keine der definierten Arbeitsklassen bzw. in Arbeitsklassen ohne Arbeitsaktionen fallen.

```

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 record(s) selected.

Schritt 5: Unterscheiden von Aktivitäten nach mehr als ihrem Typ und anderen Attributen

Sie können Aktivitäten nicht nur nach ihren Typen aufteilen. Beispielsweise wollen Sie wissen, wie viele große Abfragen ausgeführt werden.

Ändern Sie das Arbeitsklassenset, indem Sie eine neue Lesearbeitsklasse hinzufügen, die große Abfragen darstellt. In diesem Beispiel gilt jede Abfrage mit einer Kardinalität von mehr als 40 als große Abfrage.

```
ALTER WORK CLASS SET all_class_types
  ADD WORK CLASS large_wc WORK TYPE READ FOR CARDINALITY FROM 41 POSITION AT 1
```

Weitere Informationen: Beachten Sie, dass wir diese Arbeitsklasse an Position 1 platziert haben. Wird die Klausel POSITION AT nicht angegeben, wird die Arbeitsklasse am Ende des Arbeitsklassensets platziert. Bei der Entscheidung, zu welcher Arbeitsklasse eine Aktivität gehört, werden die Arbeitsklassen in der Reihenfolge geprüft, in der sie angeordnet sind, und die erste Arbeitsklasse, deren Attribute den Attributen der Aktivität entsprechen, ist die Arbeitsklasse, der die Aktivität zugeordnet wird. Wenn in diesem Fall die Arbeitsklasse large_wc am Ende der Liste platziert wäre, würden die großen Aktivitäten der Arbeitsklasse read_wc zugeordnet, da diese dann vor large_wc platziert wäre.

Ändern Sie das Arbeitsaktionsset, indem Sie eine Arbeitsaktion COUNT ACTIVITY hinzufügen, und wenden Sie es auf die neue Arbeitsklasse an.

```
ALTER WORK ACTION SET work1_was
  ADD WORK ACTION count_large_reads ON WORK CLASS large_wc COUNT ACTIVITY
```

Schritt 6: Zurücksetzen der Statistikdaten und Ausführen einiger Aktivitäten

Rufen Sie die gespeicherte Prozedur WLM_COLLECT_STATS auf, um die gespeicherten Statistikdaten zurückzusetzen, sodass Sie von vorn beginnen und beim Abrufen der Workload-Management-Statistikdaten im Speicher die Informationen zu Aktivitäten erhalten, die ab diesem Zeitpunkt ausgeführt werden.

```
CALL WLM_COLLECT_STATS()
```

Führen Sie das Script work1.db2 ein Mal aus.

```
db2 -tvf work1.db2
```

Schritt 7: Anzeigen von Statistikdaten für Arbeitsaktionssets

Verwenden Sie die Tabellenfunktion WLM_GET_WORK_ACTION_SET_STATS erneut für den Zugriff auf die Statistikdaten des Arbeitsaktionssets im Speicher, um zu ermitteln, wie oft bestimmte Aktivitätstypen ausgeführt wurden.

```
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
```

Die Ausgabe könnte folgendermaßen aussehen:

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 record(s) selected.

Beachten Sie, dass dieses Mal vier der Aktivitäten aus dem Script als große Aktivitäten gelten.

Schritt 8: Zurücksetzen für die nächste Übung

Löschen Sie das Arbeitsaktionsset:

```
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;
```

Übung 9: Erfassen von ausführlichen Informationen zu einer zurzeit ausgeführten Aktivität

Diese Übung veranschaulicht, wie Sie zur späteren Protokollanalyse mit der Prozedur `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` detaillierte Informationen zu einer in Ausführung befindlichen Aktivität erfassen können.

Geschätzte Zeit: 5-10 Minuten

Von Ihnen erfasste Aktivitätsinformationen werden an den aktiven Ereignismonitor für Aktivitäten gesendet. In den vorherigen Aufgaben wurde gezeigt, wie die Klausel `COLLECT ACTIVITY DATA` für Workloads, Serviceklassen, Arbeitsaktionen und Schwellenwerte verwendet wird, um detaillierte Aktivitätsinformationen zu erfassen. Diese Klausel muss im Voraus angegeben werden, bevor die Ausführung einer Aktivität beginnt. Die erfassten Informationen werden dann nach Abschluss der Aktivität an den Aktivitätsereignismonitor gesendet. Mit der Prozedur `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` können Sie Informationen reaktiv erfassen, wenn Sie ein Problem mit einer bereits ablaufenden Aktivität bemerken. Bei dieser Prozedur werden Informationen zu einer Aktivität sofort an den Aktivitätsereignismonitor gesendet. Es werden sowohl grundlegende als auch Anweisungsaktivitätsinformationen erfasst, jedoch keine Eingabedaten.

Schritt 1: Aktivieren des Aktivitätsereignismonitors

Aktivieren Sie den vorhandenen in Übung 1 erstellten Ereignismonitor für Aktivitäten.

```
CONNECT TO SAMPLE
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

Schritt 2: Absetzen einer lange laufenden Abfrage

Führen Sie im Befehlszeilenprozessor das folgende Script aus, das eine lang laufende Abfrage mit einem problematischen Cursor absetzt:

```
db2 -tvf longquery.db2
```

Schritt 3: Abrufen der Anwendungskennung

Rufen Sie in einem zweiten Befehlszeilenprozessorfenster WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES auf, um die Anwendungskennung, die UOW-ID und die Aktivitäts-ID der Aktivität abzurufen. Durch Verknüpfen des Ergebnisses der Tabellenfunktion mit der Verwaltungssicht APPLICATIONS können Sie die Cursoraktivität finden, die von longquery.db2 ausgeführt wird.

```
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')
```

Die Ausgabe könnte folgendermaßen aussehen:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	ACTIVITY_TYPE
267	1	1	READ_DML

1 record(s) selected.

Schritt 4: Erfassen von Informationen zur Aktivität

Rufen Sie im selben Befehlszeilenprozessorfenster die gespeicherte Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS mit der im vorherigen Schritt ermittelten Anwendungskennung, UOW-ID und Aktivitäts-ID auf:

```
CONNECT TO SAMPLE

CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS (267, 1, 1)

CONNECT RESET
```

Dieser Schritt sendet Informationen zur Aktivität an den aktiven Ereignismonitor für Aktivitäten. Beachten Sie, dass in Ihrem Fall die Anwendungskennung, UOW-ID und Aktivitäts-ID anders lauten können.

Schritt 5: Anzeigen der Aktivitätsinformationen

Sehen Sie sich die für die Aktivität erfassten Informationen mit einer Anweisung wie der folgenden an:

```
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

```

Die Ausgabe könnte folgendermaßen aussehen:

```

APPL_NAME      CLIENT_APP_NAME      APPL_ID
ACTIVITY_ID    UOW_ID              PARTIAL_RECORD TIME_STARTED
TIME_COMPLETED          STMT_TEXT
-----
-----
-----
-----
-----
-----
-----
-----
db2bp          CLP longquery.db2    *LOCAL.swalkty.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

```

Anmerkung: Aktivitäten, die mit der Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS erfasst werden, enthalten etwas weniger Informationen wie bei der Erfassung mit der Klausel COLLECT ACTIVITY DATA, da die Aktivität erfasst wird, bevor sie abgeschlossen ist. Insbesondere finden Felder wie die Fertigstellungszeitmarke (die nur Nullen enthält) und der SQLCODE keine Anwendung. Sie können ermitteln, ob eine Aktivität mit der Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS erfasst wurde, indem Sie sich die Spalte PARTIAL_RECORD in der Tabelle ACTIVITY ansehen. Wenn die Spalte PARTIAL_RECORD den Wert 1 hat (siehe Ausgabe oben), wurden Aktivitätsinformationen mit WLM_CAPTURE_ACTIVITY_IN_PROGRESS erfasst. Wenn die Spalte PARTIAL_RECORD den Wert 0 hat, wurden Aktivitätsinformationen nach Abschluss der Aktivität mit der Klausel COLLECT ACTIVITY DATA erfasst.

Übung 10: Generieren von Protokolldaten und Berichten

Diese Übung veranschaulicht die Verwendung des Beispiel-WLM-Protokollanalyse-tools.

Geschätzte Zeit: 20-25 Minuten

DB2-Datenbankprodukte schließen Perl-Beispielscripts ein, die Informationen liefern, die der WLM-Aktivitätsereignismonitor für die Protokollanalyse erfasst hat. Sie können diese Scripts zur Erstellung zusätzlicher Protokollanalyseberichte für Ihre Bedürfnisse prüfen oder ändern. Es gibt folgende Perl-Scripts:

- wlmhist.pl: generiert Protokolldaten
- wlmhistrep.pl: erzeugt Berichte aus den Protokolldaten

Schritt 1: Erstellen der EXPLAIN-Tabellen

Zur Generierung einiger Protokolldaten müssen die EXPLAIN-Tabellen unter dem Schema des Benutzers vorhanden sein, der das Tool ausführt. Wechseln Sie in das Verzeichnis /sql1lib/misc und führen Sie die folgenden Befehle aus, um die EXPLAIN-Tabellen zu erstellen:

```
db2 CONNECT TO SAMPLE
```

```
db2 -tvf EXPLAIN.DDL
```

Schritt 2: Ändern der Serviceklasse zum Erfassen von Aktivitätsdaten

Aktivieren Sie die Erfassung von Aktivitätsdaten, indem Sie die Klausel COLLECT ACTIVITY DATA für das gewünschte WLM-Objekt angeben. In dieser Übung wollen wir Protokolldaten für Aktivitäten generieren, die in der Standardserviceunterklasse der Standardbenutzerservicesuperklasse ausgeführt werden:

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS  
COLLECT ACTIVITY DATA ON COORDINATOR WITH DETAILS
```

Schritt 3: Aktivieren des Aktivitätsereignismonitors

Da der Aktivitätsereignismonitor in Übung 1 in Schritt 1 erstellt wurde, aktivieren Sie ihn jetzt, sofern er nicht bereits aktiviert ist.

```
SET EVENT MONITOR DB2ACTIVITIES STATE 1
```

Schritt 4: Ausführen einiger Aktivitäten

Führen Sie einige Aktivitäten aus, damit Aktivitätsdaten erfasst werden, zu denen Protokolldaten generiert werden können.

```
db2 -tvf work1.db2
```

```
db2 -tvf work2.db2
```

Schritt 5: Inaktivieren des Ereignismonitors für Aktivitäten

Es wird dringend empfohlen, den Ereignismonitor für Aktivitäten vor dem Generieren von Protokolldaten zu inaktivieren. Wenn Sie dies nicht tun, könnten DML-Aktivitäten, die als Ergebnis der Generierung der Protokolldaten ausgeführt werden, ebenfalls erfasst und in den DB2-Ereignismonitortabellen gespeichert werden, wodurch sich die Anzahl der tatsächlichen Aktivitäten stark erhöht, zu denen Aktivitätsdaten generiert werden.

```
CONNECT TO SAMPLE
```

```
SET EVENT MONITOR DB2ACTIVITIES STATE 0
```

Schritt 6: Generieren von Protokolldaten

Führen Sie das Script zum Generieren von Protokolldaten, wlmhist.pl, für Aktivitäten aus, die in den Tabellen des Aktivitätsereignismonitors erfasst sind. Das Format sieht folgendermaßen aus:

```
wlmhist.pl datenbankname benutzer kennwort [von_zeit bis_zeit workload-id  
serviceklassenname serviceunterklassenname aktivitätstabelle aktivitätsanweisungstabelle]
```

Verwenden Sie einen Gedankenstrich (-), um optionale Parameter zu übergangen.

Weitere Informationen: Das Protokolladatengenerator-Script (wlmhist.pl) generiert nur Protokolldaten für DML. Wenn Sie das Protokolladatengenerator-Script (wlmhist.pl) bereits mindestens einmal ausgeführt haben, empfiehlt es sich, vor der erneuten Ausführung die Tabellen aktivitätstabelle und aktivitätsanweisungstabelle zu bereinigen, um Datenduplikate zu vermeiden. Wenn Sie diese beiden Tabellen nicht bereinigt haben, müssen Sie die Eingabeparameter **von_zeit** und **bis_zeit** verwenden, um sicherzustellen, dass Sie keine Protokolldaten für Aktivitäten generieren, für die bereits Protokolldaten generiert wurden.

In diesem Beispiel generieren wir Protokolldaten für alle Aktivitäten, die im Aktivitätsereignismonitor erfasst wurden.

```
Perl wlmhist.pl sample db2inst1 kennwort
```

Es können einige Fehler ähnlich dem folgenden auftreten:

```
Error running explain [IBM][CLI Driver][DB2/LINUX8664] SQL0418N
In einer Anweisung wird eine Parametermarke in nicht zulässiger Weise verwendet.
SQLSTATE-Wert=42610
  for statement VALUES (TABLE_SCHEMA(:H00002 , :H00003  )) INTO :H00007
```

```
DBD::DB2::db do failed: [IBM][CLI Driver][DB2/LINUX8664] SQL0418N
In einer Anweisung wird eine Parametermarke in nicht zulässiger Weise verwendet.
SQLSTATE-Wert=42610
```

Beim Generieren von Protokolldaten wird EXPLAIN für die Anweisung selbst ausgeführt. In einigen Fällen kann EXPLAIN für einige Anweisungen mit Parametermarken nicht ausgeführt werden und es wird ein Fehler zurückgegeben. Für Aktivitäten, die einen solchen Fehler ergeben, werden keine Protokolldaten generiert.

Sobald das Tool die Protokolldatengenerierung erfolgreich abgeschlossen hat, sagt es Ihnen, für wie viele Aktivitäten erfolgreich Protokolldaten generiert wurden.

Schritt 7: Generieren von Protokolldatenberichten

Führen Sie das Protokolldatenberichtsscript, wlmhistrep.pl, aus, um Berichte basierend auf den in Schritt 1 generierten Daten zu generieren. Das Format sieht folgendermaßen aus:

```
wlmhistrep.pl datenbankaliasname benutzer-id kennwort
[ausgabedatei bericht schemaname von_zeit bis_zeit übergebende_funktion]
```

Verwenden Sie einen Gedankenstrich (-), um optionale Parameter zu übergehen.

Der Parameter **bericht** kann eine Kombination der folgenden Buchstaben sein:

- A: Übereinstimmende Tabellen
- B: Nicht übereinstimmende Tabellen
- C: Übereinstimmende Indizes
- D: Nicht übereinstimmende Indizes
- E: Übergebende Funktionen

Wenn der angegebene Parameter **benutzer-id** nicht der Benutzer-ID entspricht, mit der das Script wlmhist.pl beim Erstellen der Tabelle wlmhist ausgeführt wurde, müssen Sie den korrekten Wert für **schemaname** angeben. Die Parameter **von_zeit** und **bis_zeit** müssen im Zeitmarkenformat angegeben werden (z. B. 2007-06-06-17.00.00).

In dieser Übung generieren wir Berichte für übereinstimmende Tabellen und nicht übereinstimmende Indizes:

```
Perl wlmhistrep.pl sample db2inst1 kennwort - AD
```

Die Ausgabe könnte folgendermaßen aussehen:

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
:				
:				

Schritt 8: Zurücksetzen für die nächste Übung

Inaktivieren Sie die Aktivitätserfassung für die Standardserviceunterklasse der Standardbenutzerservicesuperklasse und bereinigen Sie die Aktivitätstabellen.

```
ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
COLLECT ACTIVITY DATA NONE
```

```
DELETE FROM ACTIVITY_DB2ACTIVITIES
DELETE FROM ACTIVITYSTMT_DB2ACTIVITIES
```

Übung 11: Verwenden von erweiterten Zusammenfassungen für Serviceklassen

Diese Übung veranschaulicht, wie die Option COLLECT AGGREGATE ACTIVITY DATA EXTENDED für eine Serviceklasse verwendet werden kann, um Histogramme der Zwischenankunftszeiten und des geschätzten Aufwands von Koordinatoraktivitäten erzeugen zu können.

Geschätzte Zeit: 25-30 Minuten

Die *Zwischenankunftszeit* (Inter-Arrival Time) ist das Zeitintervall zwischen dem Eingang einer Aktivität im System und dem Eingang der nächsten Aktivität. Der

geschätzte Aufwand einer Aktivität stellt eine Schätzung des SQL-Compilers der Systemressourcen dar, die bei der Ausführung der Aktivität verwendet werden. Er bezieht sich nur auf DML-Aktivitäten.

Histogramme der Zwischenankunftszeit oder Histogramme des geschätzten Aufwands können miteinander, mit Laufzeithistogrammen oder mit anderen Laufzeitstatistiken korreliert werden, um zu ermitteln, ob Änderungen in Laufzeithistogrammen oder in Laufzeitstatistiken das Ergebnis eines der folgenden Ereignisse sein können:

- Eine Änderung in der Komplexität der Workload, beispielsweise eine Änderung der Verteilung des geschätzten Aufwands
- Eine Änderung in der Ankunftsrate von Aktivitäten, die durch die Verteilung der Zwischenankunftszeit bestimmt wird
- Eine Änderung auf dem System selbst, beispielsweise die Einführung eines neuen Schwellenwerts, eine Änderung in der Priorität einer Serviceklasse oder eine geänderte Hardware

Weitere Informationen zu Histogrammen finden Sie in „Histogramme beim Workload-Management“ auf Seite 276.

Der Zugriff auf Histogramme erfolgt über den Statistikereignismonitor. In dieser Übung wird wieder der in Übung 1 in Schritt 1 erstellte Statistikereignismonitor verwendet.

Schritt 1: Erstellen von Sichten zum Anzeigen von Histogrammstatistikdaten

Erstellen Sie verschiedene Sichten, um die Abfrage der Tabelle HISTOGRAMBIN_DB2STATISTICS zu vereinfachen. Die erste Sicht enthält alle verfügbaren Histogrammtypen. In dieser Übung gibt es nur die drei Grundtypen: Laufzeit, Ausführungszeit und Wartezeit.

```
CREATE VIEW HISTOGRAMTYPES AS
  SELECT DISTINCT SUBSTR(HISTOGRAM_TYPE,1,24) HISTOGRAM_TYPE
  FROM HISTOGRAMBIN_DB2STATISTICS
```

Eine zweite Sicht macht es einfacher, herauszufinden, für welche Serviceklassen Histogramme erfasst werden. Die Tabelle HISTOGRAMBIN_DB2STATISTICS listet die Serviceklasse, für die das Histogramm erfasst wird, mit der Serviceklassen-ID auf. Durch die Verknüpfung dieser Tabelle mit der Katalogtabelle SERVICECLASSES können die Serviceklasseninformationen mit dem Servicesuperklassennamen und dem Serviceunterklassennamen statt mit der Serviceklassen-ID dargestellt werden.

```
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
```

Die dritte Sicht enthält alle Zeiten, bei denen ein Histogramm des angegebenen Typs für eine angegebene Serviceklasse erfasst wurde. Wie die Sicht HISTOGRAMSERVICECLASSES verknüpft sie die Tabelle HISTOGRAMBIN_DB2STATISTICS mit der Katalogtabelle SERVICECLASSES. Der Unterschied liegt darin, dass die Spalte STATISTICS_TIMESTAMP als eine der Spalten in der Sicht enthalten ist.

```

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

```

Die vierte und letzte Sicht enthält die Histogramme. Sie zeigt auch eine allgemeine Task im Zusammenhang mit Histogrammen, nämlich die Zusammenfassung über die Zeit. Diese Sicht zeigt den Anfang jedes Bins und die Anzahl der Aktivitäten, die für jeden Bin gezählt wurden. Bei den beiden folgenden Histogrammen misst das Feld BIN_TOP die Zwischenankunftszeit der Aktivität (in Millisekunden) und den geschätzten Aufwand (in Timeron). Wenn bei einem Zwischenankunftszeithistogramm BIN_TOP z. B. 3000 Millisekunden und der BIN_TOP-Wert des vorherigen Bin 2000 Millisekunden beträgt und der Wert von NUMBER_IN_BIN 10 ist, dann wissen Sie, dass es zehn Aktivitäten gab, wobei jede 2 bis 3 Sekunden nach der vorherigen Aktivität in das System einging.

```

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

```

Schritt 2: Aktivieren der Erfassung von Histogrammen

Das Aktivieren der Histogrammerfassung erfolgt für die Standardbenutzerserviceklasse durch Ändern ihrer Standardunterklasse, sodass mit der Option EXTENDED zusammengefasste Aktivitätsdaten erfasst werden. Dadurch werden die drei mit der Option BASE verfügbaren Histogramme (Laufzeit, Ausführungszeit und Wartezeit) sowie die beiden Histogramme bereitgestellt, die nur über die Option EXTENDED verfügbar sind (Zwischenankunftszeit und geschätzter Aufwand).

```

ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
  COLLECT AGGREGATE ACTIVITY DATA EXTENDED

```

Schritt 3: Aktivieren des Statistikereignismonitors

Aktivieren Sie den bereits erstellten Ereignismonitor, wenn er noch nicht aktiv ist, sodass er die Zusammenfassungsdaten bei deren Erfassung empfangen kann.

```

SET EVENT MONITOR DB2STATISTICS STATE 1

```

Schritt 4: Ausführen von Aktivitäten und Senden von Statistikdaten an den Statistikereignismonitor

Führen Sie zuerst einige Aktivitäten aus. Nach Abschluss der Aktivitäten wird die gespeicherte Prozedur WLM_COLLECT_STATS aufgerufen, um die Serviceklassenstatistikdaten an den aktiven Statistikereignismonitor zu senden (einschließlich Laufzeit, Ausführungszeit, Wartezeit, Zwischenankunftszeit und geschätzter Aufwand der Aktivitäten für die Standardbenutzerserviceklasse). Diese Histogramme enthalten Daten zu allen Aktivitäten, die seit der Aktivierung von zusammengefassten Aktivitätsstatistikdaten in der Standardbenutzerserviceklasse ausgeführt wurden. Durch den Aufruf der gespeicherten Prozedur werden die Statistikdaten

auch zurückgesetzt. Zum Anzeigen von Änderungen der Datenbankaktivität im Lauf der Zeit werden drei Erfassungsintervalle erstellt.

Führen Sie im ersten Intervall zwei Scripts, work1.db2 und work2.db2, aus und lassen Sie dann die Statistikdaten erfassen und zurücksetzen.

```
db2 -o- -tvf work1.db2
db2 -o- -tvf work2.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS
```

Führen Sie im zweiten Intervall nur das Script work1.db2 einmal aus und lassen Sie dann die Statistikdaten erfassen und zurücksetzen.

```
db2 -o- -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS
```

Führen Sie im dritten Intervall das Script work1.db2 zweimal und das Script work2.db2 einmal aus und lassen Sie dann die Statistikdaten erfassen und zurücksetzen.

```
db2 -o- -tvf work1.db2
db2 -o- -tvf work2.db2
db2 -o- -tvf work1.db2
```

```
CONNECT TO SAMPLE
```

```
CALL WLM_COLLECT_STATS
```

Durch das periodische Erfassen von Daten können Sie überwachen, wie sich die Arbeit auf Ihrem System im Lauf der Zeit ändert.

Zusätzliche Informationen: Das periodische Erfassen von Daten muss nicht manuell erfolgen. Mit dem Datenbankkonfigurationsparameter WLM_COLLECT_INT kann das Intervall (in Minuten) festgelegt werden, in dem die Statistikerfassung und -rücksetzung erfolgt.

Schritt 5: Abfragen von Sichten zum Anzeigen von Statistikdaten

Nachdem Statistikdaten erfasst wurden, können die zuvor erstellten Sichten zum Anzeigen der Statistikdaten verwendet werden. Die Sicht HISTOGRAMTYPES gibt nur die Typen der verfügbaren Histogramme zurück.

```
SELECT * FROM HISTOGRAMTYPES
```

```
HISTOGRAM_TYPE
-----
CoordActEstCost
CoordActExecTime
CoordActInterArrivalTime
CoordActLifetime
CoordActQueueTime
```

```
3 record(s) selected.
```

Da beim Ändern der Serviceklasse die Option EXTENDED verwendet wurde, gibt es fünf Histogramme:

In der Sicht HISTOGRAMSERVICECLASSES werden die Serviceklassen angezeigt, für die ein Histogramm erfasst wurde. Das folgende Beispiel schränkt die Ausgabe auf das Histogramm CoordActInterArrivalTime ein. Da die Erfassung zusammengefasster Aktivitätsinformationen nur für die Standardunterklasse der Standardbenutzerserviceklasse aktiviert wurde, wird bei der Auswahl aus der Sicht HISTOGRAMSERVICECLASSES nur diese Klasse angezeigt.

```
SELECT * FROM HISTOGRAMSERVICECLASSES
WHERE HISTOGRAM_TYPE = 'CoordActInterArrivalTime'
ORDER BY SERVICE_SUPERCLASS, SERVICE_SUBCLASS
```

HISTOGRAM_TYPE	SERVICE_SUPERCLASS	SERVICE_SUBCLASS
CoordActInterArrivalTime	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS

1 record(s) selected.

Die Sicht HISTOGRAMTIMES enthält die Zeiten, zu denen Histogramme erfasst wurden. Da die Prozedur WLM_COLLECT_STATS dreimal ausgeführt wurde, gibt es im Zwischenankunftszeithistogramm drei Zeitmarken.

```
SELECT * FROM HISTOGRAMTIMES
WHERE HISTOGRAM_TYPE = 'CoordActInterArrivalTime'
AND SERVICE_SUPERCLASS = 'SYSDEFAULTUSERCLASS'
AND SERVICE_SUBCLASS = 'SYSDEFAULTSUBCLASS'
ORDER BY TIMESTAMP
```

HISTOGRAM_TYPE TIMESTAMP	SERVICE_SUPERCLASS	SERVICE_SUBCLASS
CoordActInterArrivalTime 2007-08-08-13.41.38.870298	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS
CoordActInterArrivalTime 2007-08-08-13.41.42.802855	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS
CoordActInterArrivalTime 2007-08-08-13.41.53.577835	SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS

Die letzte Sicht HISTOGRAMS enthält die Histogramme. Im Gegensatz zur Sicht HISTOGRAMTIMES, die jedes Erfassungsintervall als eigene Zeile auflistet, fasst die Sicht HISTOGRAMS Histogrammdaten über mehrere Intervalle zusammen, um ein einzelnes Histogramm des angegebenen Typs für eine angegebene Serviceklasse zu erstellen.

```
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 record(s) selected.

Die Ausführung dieser Abfrage erzeugt eine Ausgabe, die nicht exakt der im obigen Beispiel gezeigten entspricht, da die Zwischenankunftszeiten der Aktivitäten von der Leistung des Systems abhängen. Die obige Ausgabe enthält 41 Bins, wobei alle größeren Bins leer sind. Am Anfang steht ein Bin, bei dem BIN_TOP -1 ist. Dieser Bin stellt alle Aktivitäten dar, deren Zwischenankunftszeit zu groß für das Histogramm war. Wenn der Wert für NUMBER_OF_BIN größer als null und der Wert für BIN_TOP -1 ist, sollten Sie den oberen Binwert Ihres Histogramms hochsetzen. In der obigen Ausgabe ist NUMBER_IN_BIN 0, sodass hier keine solche Änderung erforderlich ist. Die Mehrzahl der Aktivitäten traf mit einem zeitlichen Abstand von weniger als 1309 Millisekunden ein. Vier Aktivitäten trafen mit einem zeitlichen Abstand zwischen 7089 und 16493 Millisekunden ein.

Dieselbe Abfrage kann mit einem Histogrammtyp von CoordActEstCost statt CoordActInterArrivalTime wiederholt werden.

```
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 record(s) selected.

Ein Histogramm wie dieses ist typisch für eine kleine Workload. Bei einer kleinen Workload gibt es keine große Schwankung in der Größe der Aktivitäten, sodass Aktivitäten nur in drei unterschiedlichen Bins gezählt wurden. Etwas mehr als 60 % der Aktivitäten hatten einen geschätzten Aufwand zwischen 5 und 19 Timeron, während der Rest einen geschätzten Aufwand von weniger als 1 Timeron hatte.

Schritt 6: Zurücksetzen für andere Übungen

Als letzten Schritt müssen Sie die Erfassung von zusammengefassten Aktivitäten in der Standardbenutzerserviceklasse inaktivieren, die Sichten löschen und die Informationen in den Statistiktabelle löschen.

```
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
```

Kapitel 7. Workload-Management-Szenarios

Workload-Management-Beispielanwendung

In Ihren DB2-Datenserver wurden mit dem DB2-Workload-Management umfassende Workload-Management-Features integriert, was Ihnen eine bessere Steuerung der Aktivitäten, Ressourcen und der Leistung ermöglicht sowie aussagekräftige Informationen über die Funktionsweise Ihres Systems bereitstellt. Eine Workload-Management-Beispielanwendung ist jetzt unter developerWorks verfügbar.

Die Workload-Management-Beispielanwendung veranschaulicht, wie Sie DB2-Workload-Management-Features für die folgenden Ziele einsetzen können:

Schutz des Systems vor nicht mehr steuerbaren Abfragen

Nicht mehr steuerbare Abfragen sind aufwendig und verschlechtern die Leistung. Die Workload-Management-Beispielanwendung ermittelt Abfragen, die potenziell nicht mehr steuerbar werden, und stoppt die Ausführung dieser Abfragen, nachdem sie gegen einen bestimmten Schwellenwert verstoßen haben.

Beschränken der gleichzeitigen Ressourcenauslastung durch einzelne Anwendungen

Die Beispielanwendung zeigt, wie Sie mit DB2-Workload-Management-Features verhindern können, dass Anwendungen, die große Mengen gleichzeitig ablaufender Arbeit übergeben, sich negativ auf die Leistung anderer Anwendungen auswirken.

Erreichen einer bestimmten Antwortzeit

Mit Workload-Management-Features können Sie eine bestimmte angestrebte Antwortzeit der Form "Transaktion X von Anwendung Y soll in 90 % aller Fälle innerhalb 1 Sekunde abgeschlossen sein" erreichen, unabhängig davon, welche andere Aktivität gleichzeitig auf dem System ausgeführt wird. Die Beispielanwendung zeigt, wie Sie eine angestrebte Antwortzeit erreichen können.

Konsistente Antwortzeit für kurze Abfragen

Abfragen, die typischerweise eine Antwortzeit von weniger als 1 Sekunde haben, sollten eine relativ konsistente Antwortzeit haben, unabhängig davon, welche anderen Workloads gerade auf dem System ausgeführt werden. Die Beispielanwendung verwendet zur Überwachung der Konsistenz das Histogramm der Abfrageausführungszeit.

Schutz des Systems in Zeiträumen mit Lastspitzen

Features für Workload-Management-Richtlinien schützen das System vor Überlastung bei Lastspitzen, indem Arbeit in die Warteschlange gestellt wird, sobald das System ausreichend ausgelastet ist.

Aktivieren von gleichzeitig ablaufender Extraktions-, Transformations- und Ladeverarbeitung und von Benutzerabfragen

Mit Workload-Management-Features können Sie Extraktions-, Transformations- und Ladejobs (wie das Laden von Daten in Tabellen) ausführen und dabei den Leistungseinfluss für Benutzer steuern, die gleichzeitig Abfragen ausführen.

Informationen zum Herunterladen der Beispielanwendung finden Sie auf der developerWorks-Website in Workload Management Samples.

Szenario: Untersuchen einer workloadbezogenen Leistungsminderung des Systems

Wenn Sie eine Leistungsminderung des Systems bemerken (z. B. dauert die Ausführung einiger Anwendungen viel länger als erwartet) und Sie nicht sicher sind, ob das Problem mit der Konfiguration der Workloads zusammenhängt, können Sie mit Tabellenfunktionsdaten das Problem untersuchen und erforderlichenfalls beheben.

Erstellen Sie zuerst eine Abfrage, die Daten unter Verwendung der von der Tabellenfunktion `WLM_GET_SERVICE_SUBCLASS_STATS` zurückgegebenen Daten serviceklassen- und datenbankmemberübergreifend zusammenfasst. Definieren Sie für das erste und das zweite Argument leere Zeichenfolgen und stellen Sie das dritte Argument auf `-2` ein, um anzugeben, dass Daten für alle Serviceklassen in allen Datenbankmembern zusammengestellt werden sollen.

Ihre Abfrage könnte der folgenden ähneln:

```
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

Im Beispiel oben führt die Serviceunterklasse `SUB1` in der Servicesuperklasse `SUP1` gleichzeitig mehr Aktivitäten als gewöhnlich aus. Zur weiteren Untersuchung wollen Sie eventuell die Statistikdaten für Workloads prüfen, die dieser Serviceklasse zugeordnet sind. Ihre Abfrage könnte der folgenden ähneln:

```
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

Die Ausgabe zeigt, dass eine Anwendung in der Workload `LYNNSALES` gleichzeitig 8 Aktivitäten übergeben hat. Sie können erwägen, einen Schwellenwert hinzuzufügen, der den gemeinsamen Zugriff von Koordinatoraktivitäten für jedes Workloadvorkommen beschränkt.

Szenario: Zusammenfassen von Aktivitätsmessdaten über untergeordnete Aktivitäten hinweg

Jede Zeile in einem Aktivitätsereignismonitor enthält Informationen und Überwachungsmessdaten, die die Ausführung einer bestimmten Aktivität beschreiben. Die Aktivitätsinformationen umfassen die Monitorelemente **parent_uow_id** und **parent_activity_id**. Diese Monitorelemente sind ungleich null, wenn die Aktivität eine untergeordnete Aktivität einer anderen Aktivität (verschachtelt) ist und die Elemente die übergeordnete Aktivität angeben.

Bei der Verwendung von rekursivem SQL können die Überwachungsmessdaten im Aktivitätsereignismonitor über alle untergeordneten Aktivitäten hinweg für die entsprechenden übergeordneten Aktivitäten hinweg zusammengefasst werden. Die Anzeige der zusammengefassten Messdaten für die übergeordnete Aktivität ist beim Ermitteln der Teile einer gespeicherten Prozedur hilfreich, für die eine Optimierung vorteilhaft sein kann. Wenn beispielsweise die gespeicherte Prozedur P1 die gespeicherten Prozeduren P2 und P3 aufruft, können die zusammengefassten Messdaten anzeigen, dass 90 % der CPU-Lastung während eines Aufrufs an P1 während der Verarbeitung von Anweisungen in der verschachtelten Prozedur P3 auftritt. So können Sie den Schwerpunkt Ihrer Untersuchungen entsprechend legen.

Sie können mithilfe des folgenden Beispiels Zusammenfassungen verschachtelter Messdaten von einem Aktivitätsereignismonitor abrufen.

Anmerkung: Das Zeichen @ wird in diesem Beispiel als Anweisungsabschlusszeichen verwendet.

Die folgenden Anweisungen können in eine Datei (beispielsweise `test.clp`) kopiert und mit `db2 -td@ -f dateiname` ausgeführt werden, wie im folgenden Beispielbefehl gezeigt:

```
db2 -td@ -f test.clp
```

Nehmen wir für dieses Beispiel an, dass die folgenden Tabellen und gespeicherten Prozeduren vorhanden sind:

```
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@

```

Prozedur

1. Erstellen Sie einen Aktivitätseignismonitor und aktivieren Sie die Aktivitätserfassung. In diesem Beispiel wird die Aktivitätserfassung für die aktuelle Verbindung mit der Prozedur WLM_SET_CONN_ENV aktiviert. Führen Sie die Prozedur TEST.P1 mit derselben Verbindung aus und inaktivieren Sie dann den Aktivitätseignismonitor. Die Aktivitätsinformationen werden für die Prozedur TEST.P1 und alle ihre untergeordneten Aktivitäten erfasst.

```

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. Führen Sie die folgende Abfrage aus, um die erfassten Anweisungen mit ihren übergeordneten Identifikationsinformationen, der jeweiligen CPU-Belegung und der zusammengefassten CPU-Belegung anzuzeigen. Die zusammengefasste CPU-Belegung wird rekursiv über alle untergeordneten Aktivitäten hinweg addiert. Diese Abfrage kann auf einfache Art erweitert werden, sodass alle vom Aktivitätseignismonitor erfassten Überwachungsmessdaten unterstützt werden.

Anmerkung: Zur besseren Lesbarkeit wird hier nur die CPU-Belegung ausgegeben.

```

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@

```

Diese Abfrage erzeugt die folgende Ausgabe. Die Spalte STMT_CPU gibt die CPU-Belegung für die Anweisung an, jedoch nicht die CPU-Belastung der untergeordneten Aktivitäten. Die Spalte AGG_CPU gibt die zusammengefasste CPU-Belegung für die Aktivität und alle ihre untergeordneten Aktivitäten an.

Anmerkung: Wenn Abfragen zu schnell ausgeführt werden, sodass sie nicht gemessen werden können, können die Spalten STMT_CPU und AGG_CPU den Wert 0 enthalten.

UOW_ID	ACTIVITY_ID	PARENT_UOW_ID	PARENT_ACTIVITY_ID	STMT_CPU	AGG_CPU	STMT_TEXT
576	1	0		0	5353	84064 CALL TEST.P1()
576	2	576		1	7444	52043 CALL TEST.P3()
576	3	576		2	1869	1869 INSERT INTO T1 VALUES(1)
576	4	576		2	11727	26935 CALL TEST.P4()
576	5	576		4	2017	2017 INSERT INTO T1 VALUES(5)
576	6	576		4	6602	6602 INSERT INTO T1 VALUES(6)
576	7	576		4	6589	6589 INSERT INTO T1 VALUES(7)
576	8	576		2	15795	15795 SELECT COUNT(*) INTO :HV00
576	9	576		1	11727	20314 CALL TEST.P2()
576	10	576		9	1941	1941 INSERT INTO T1 VALUES(2)
576	11	576		9	6646	6646 INSERT INTO T1 VALUES(3)
576	12	576		1	6354	6354 INSERT INTO T1 VALUES(4)

12 record(s) selected.

Szenario: Ermitteln von Aktivitäten, deren Ausführung zu lange dauert

Workload-Management-Tabellenfunktionen erleichtern die Ermittlung einer bestimmten Aktivität im Datenserver und gegebenenfalls deren Abbruch, ohne dass dabei die gesamte Anwendung beendet werden muss.

Ermitteln einer Aktivität, deren Ausführung zu lange dauert

Im Folgenden sehen Sie ein Beispiel für die Ermittlung einer lange laufenden Abfrage. Nehmen wir an, dass ein Benutzer der Vertriebsabteilung (SALES), der die Anwendung SalesReport ausführt, sich beklagt, dass die Ausführung der Anwendung zu lange dauert.

Ermitteln Sie die Anwendungskennung und zeigen Sie dann mit der Tabellenfunktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES alle Aktivitäten an, die zurzeit in dieser Anwendung ausgeführt werden. Wenn die Anwendungskennung z. B. 1 ist, könnte Ihre Abfrage der folgenden ähneln:

```

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(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

Für die Aktivität wird die UOW-ID 2 und die Aktivitäts-ID 5 ermittelt. Mit der Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS können Sie dann erkennen, was die Agenten tun, die an dieser Aktivität arbeiten:

```
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
```

Die Aktivität könnte z. B. in der Warteschlange stehen, ausgeführt werden oder aufgrund einer Sperre warten. Wenn die Aktivität in der Warteschlange steht, sieht das Ergebnis folgendermaßen aus:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	REQUEST_TYPE	EVENT_TYPE	EVENT_OBJECT
1	2	5	OPEN	WAIT	WLM_QUEUE

Wenn die Aktivität ausgeführt wird, sieht das Ergebnis folgendermaßen aus:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	REQUEST_TYPE	EVENT_TYPE	EVENT_OBJECT
1	2	5	OPEN	PROCESS	REQUEST

Wenn die Aktivität aufgrund einer Sperre wartet, sieht das Ergebnis folgendermaßen aus:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	REQUEST_TYPE	EVENT_TYPE	EVENT_OBJECT
1	2	5	OPEN	ACQUIRE	LOCK

Wenn Sie wissen, was die Aktivität gerade tut, können Sie entsprechend fortfahren:

- Wenn die Aktivität in der Warteschlange steht und der Benutzer angibt, dass die Abfrage so lange dauert, dass das Ergebnis nicht mehr von Interesse ist, oder wenn Sie meinen, dass die Abfrage zu viele Ressourcen verbraucht, können Sie sie abbrechen.
- Wenn die Aktivität wichtig ist und sich in der Warteschlange befindet, können Sie erwägen, andere weniger wichtige gerade ausgeführte Arbeit abzubrechen (und dadurch den gemeinsamen Zugriff verringern, sodass Aktivitäten aus der Warteschlange genommen werden). Vielleicht reicht es dem Benutzer auch, zu wissen, dass die Arbeit nicht blockiert ist, sondern nur auf die Gelegenheit wartet, ausgeführt zu werden.
- Wenn die Aktivität aufgrund einer Sperre wartet, können Sie mit Snapshot Monitor untersuchen, aufgrund welcher Sperren die Anwendung wartet.
- Wenn die Aktivität aufgrund einer Sperre wartet, die von einer Aktivität mit niedrigerer Priorität gehalten wird, können Sie erwägen, diese Aktivität abzubrechen.

Vielleicht finden Sie es auch hilfreich, die DML-Anweisung zu kennen, die die Aktivität 5 ausführt. Wenn Sie einen aktiven Aktivitätsereignismonitor haben, können Sie die Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS ausführen, um Informationen zur DML-Anweisung sowie andere Informationen zur Aktivität 5

während ihrer Ausführung zu erfassen. Im Gegensatz zum Anweisungsereignismonitor können Sie mit der Prozedur `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` Informationen zu einer bestimmten Abfrage statt zu jeder zu diesem Zeitpunkt ausgeführten Anweisung erfassen. Sie können mit `MON_GET_ACTIVITY_DETAILS_COMPLETE` auch den Anweisungstext erhalten.

Wenn Sie beschließen, dass Sie die Aktivität abbrechen müssen, können Sie mit der Routine `WLM_CANCEL_ACTIVITY` die Aktivität abbrechen, ohne die absetzende Anwendung beenden zu müssen:

```
CALL WLM_CANCEL_ACTIVITY (1, 2, 5)
```

Die Anwendung, die die Aktivität abgesetzt hat, empfängt einen Fehler `SQL4725N`. Jede Anwendung, die mit negativem SQL-Code umgehen kann, kann diesen SQL-Code handhaben.

Ermitteln einer aufgrund eines Sperrenkonflikts blockierten Aktivität

Nehmen wir an, dass Sie eine Situation haben, in der sich ein Benutzer über eine Anwendung beklagt, deren Ausführung zu lange dauert. Nehmen wir weiter an, dass Sie entweder den Anwendungsnamen oder die Berechtigungs-ID der Anwendung mit langer Laufzeit kennen. Mithilfe dieser Informationen können Sie mit dem Befehl **LIST APPLICATIONS** die Anwendungskennung ermitteln. Angenommen, die vom Befehl **LIST APPLICATIONS** zurückgegebene Anwendungskennung lautet 2. Mit der Tabellenfunktion `WLM_GET_SERVICE_CLASS_AGENTS` können Sie dann ermitteln, welche Agenten an dieser Aktivität arbeiten. Ihre Abfrage könnte der folgenden ähneln:

```
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

Die Ergebnisse zeigen, dass der Agent 1 auf eine ferne Antwort wartet. Wenn Sie sich den Agenten in dem fernen Member ansehen, der an derselben Aktivität arbeitet, zeigt das Feld `EVENTOBJECT` an, dass der Agent darauf wartet, eine Sperre zu erhalten.

Als Nächstes muss ermittelt werden, wer der Eigentümer der Sperre ist. Sie können diese Information durch Aktivieren der Monitorschalter und Verwenden der Tabellenfunktion von Snapshot Monitor erhalten (siehe folgendes Beispiel):

```
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

Sie können den Sperreneigentümer auch mit der folgenden Befehlsfolge ermitteln:

```
db2pd -db aliasname-der-datenbank -locks
db2pd -db aliasname-der-datenbank -transactions
```

Wenn Sie die lange laufende Aktivität abbrechen wollen, können Sie die Prozedur WLM_CANCEL_ACTIVITY verwenden. Wenn der erfolgreiche Abschluss der lange laufenden Anwendung wichtiger ist als der erfolgreiche Abschluss der Anwendung, die Eigentümer der Sperre ist, können Sie die Beendigung dieser Anwendung erzwingen.

Szenario: Abbrechen von Aktivitäten, die länger als eine Stunde in der Warteschlange stehen

Mithilfe der hier beschriebenen Beispielscripts können Sie eine Prozedur zum Abbrechen von Aktivitäten erstellen, die länger als eine Stunde in der Warteschlange stehen. Darüber hinaus wird ein Beispielscript bereitgestellt, mit dem Sie die Prozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten mithilfe von DB2 Administrative Task Scheduler so terminieren können, dass sie alle 10 Minuten ausgeführt wird.

Die Prozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten erfasst auch Informationen zu den abgebrochenen Aktivitäten (sofern ein Aktivitätsereignismonitor aktiv ist) und verwaltet eine kleine Protokolltabelle der abgebrochenen Aktivitäten. Diese beiden Informationskomponenten sind optional und Kommentare im Beispielscript geben an, wo die Komponenten, wenn sie nicht erforderlich sind, auf Kommentar gesetzt werden können.

Die in der Beispielprozedur enthaltenen Anweisungen sind selbst Aktivitäten und werden (abhängig von der Konfiguration von Schwellenwerten auf Ihrem System) vom Schwellenwert gesteuert. Sie können erwägen, die Beispielprozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten in einer Serviceklasse auszuführen, für die keine Warteschlangenschwellenwerte angewendet wurden.

1. Kopieren Sie das folgende Beispielscript, das die Prozedur zum Abbrechen von Aktivitäten erstellt, die länger als eine Stunde in der Warteschlange stehen, in eine von Ihnen erstellte Datei (beispielsweise in eine Datei mit dem Namen `x.clp`):

```
-- Simple history table to track cancelled
-- activities

CREATE TABLE SAMPLE.CANCELED_ACTIVITIES(
  APPLICATION_HANDLE BIGINT,
  UOW_ID BIGINT,
  ACTIVITY_ID BIGINT )@

-- Cancel any activities that have been queued
-- for more than 1 hour

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 TIMESTAMPDIFF(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;

-- Ignore errors for activity not found and activity event
-- monitor does not exist.
DECLARE CONTINUE HANDLER FOR SQLSTATE '5U035', SQLSTATE '01H53'
        BEGIN
        END;

-- Find all activities that are queued by WLM
-- thresholds where (ACTIVITY_STATE = 'QUEUED')
OPEN QUEUEDAPPS;
FETCH QUEUEDAPPS INTO APPHANDLE, UOWID, ACTIVITYID;

WHILE AT_END = 0 DO

-- Now use activity entry time to estimate the time spend queued.
-- Queuing occurs before an activity begins execution, so queue
-- time is approximated using current time - entry time
        OPEN QTIMECUR;
        FETCH QTIMECUR INTO QUEUETIME;
        CLOSE QTIMECUR;

        IF ( QUEUETIME >= 1 ) THEN

-- Optional: Insert a record into a table to record the
-- cancellation of the statement (for monitoring purposes, to
-- understand how many statements were cancelled). Modify this
-- insert as required to capture more info such as the name of
-- the application that submitted the cancelled query. Comment out
-- these 2 lines if the monitoring is not important to you.
                INSERT INTO SAMPLE.CANCELED_ACTIVITIES VALUES ( APPHANDLE,
                        UOWID, ACTIVITYID );

-- Optional: Send details about activity to any activity activities
-- event monitor before cancelling. Comment out
-- this line if you don't care about the details of the
-- statements that were cancelled
                CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS( APPHANDLE, UOWID,
                        ACTIVITYID );

-- Cancel the activity
                CALL WLM_CANCEL_ACTIVITY( APPHANDLE, UOWID, ACTIVITYID );

-- Explicit commit, required for the insert statement above. The
-- admin task scheduler will not perform a commit. Comment out this
-- line if the insert statement is removed.
                COMMIT;

        END IF;

        FETCH QUEUEDAPPS INTO APPHANDLE, UOWID, ACTIVITYID;

END WHILE;

```

```
CLOSE QUEUEDAPPS;
```

```
END@
```

2. Erstellen Sie die Prozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten, indem Sie das Script `x.clp` mit dem folgenden Befehl ausführen:

```
db2 -td@ -f x.clp
```

3. Führen Sie die Prozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten mit dem folgenden Befehl aus:

```
db2 "call sample.cancel_queued_activities()"
```

Alle Aktivitäten, die länger als eine Stunde in der Warteschlange stehen, werden abgebrochen.

4. Das folgende Beispielscript terminiert die Prozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten mithilfe von DB2 Administrative Task Scheduler so, dass sie alle 10 Minuten ausgeführt wird. Kopieren Sie das Beispielscript in eine von Ihnen erstellte Datei (beispielsweise in eine Datei mit dem Namen `y.clp`):

```
-----  
-- Enable DB2 Admin Task Scheduler if  
-- not already enabled.  
-----  
  
!db2set DB2_ATS_ENABLE=YES@  
  
-----  
-- Create SYSTOOLSPACE tablespace.  
-- Enable if SYSTOOLSPACE does not already  
-- exist on your database.  
-----  
  
-- CREATE TABLESPACE SYSTOOLSPACE IN IBMCATGROUP MANAGED BY AUTOMATIC STORAGE  
-- EXTENTSIZE 4@  
  
-----  
-- Add a task to automatically cancel  
-- activities that have been queued  
-- for more than 1 hour. Task is scheduled  
-- to run every 10 minutes. Adjust the  
-- schedule as necessary using the  
-- schedule input parameter (specified in  
-- cron format).  
-----  
  
CALL SYSPROC.ADMIN_TASK_ADD(  
    'CANCEL ACTIVITIES QUEUED FOR MORE 1 HOUR',  
    NULL,  
    NULL,  
    NULL,  
    '* / 10 * * * *',  
    'SAMPLE',  
    'CANCEL_QUEUED_ACTIVITIES',  
    NULL,  
    NULL,  
    NULL )@
```

5. Terminieren Sie die Prozedur zum Abbrechen von in der Warteschlange stehenden Aktivitäten so, dass sie alle 10 Minuten ausgeführt wird, indem Sie das Script `y.clp` mit dem folgenden Befehl ausführen:

```
db2 -td@ -f y.clp
```

Szenario: Erkennen von Aktivitäten mit niedrigem geschätztem Aufwand und langer Laufzeit

Das folgende Beispiel zeigt, wie Sie mit Arbeitsklassen, Arbeitsaktionssets, Schwellenwerten und der Aktivitätserfassung Aktivitäten ermitteln können, die einen geringen geschätzten Aufwand aber eine hohe Laufzeit haben. Diese Situation könnte darauf hindeuten, dass der geschätzte Aufwand (in Timeron) aufgrund von veralteten Tabellen- und Indexstatistikdaten ungenau ist.

Als Erstes muss ein Arbeitsklassenset mit einer Arbeitsklasse erstellt werden, mit der Aktivitäten mit einem niedrigen geschätzten Aufwand ermittelt werden. Beispiel:

```
CREATE WORK CLASS SET WCS1 (WORK CLASS SMALLDML WORK TYPE DML FOR TIMERONCOST FROM 0 TO 500)
```

Als Nächstes erstellen Sie dann ein Datenbankarbeitsaktionsset mit einer Arbeitsaktion, die einen Schwellenwert für die Gesamtzeit der Aktivität auf die Arbeitsklasse SMALLDML anwendet. Die Schwellenwertaktion ist CONTINUE und die Option COLLECT ACTIVITY DATA ist angegeben, sodass eine Aktivität, die einen Schwellenwertverstoß verursacht, bei ihrem Abschluss an den Aktivitätsereignismonitor gesendet wird:

```
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)
```

Zuletzt erstellen und aktivieren Sie einen Ereignismonitor für Schwellenwertverstöße und einen Aktivitätsereignismonitor:

```
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
```

Wenn jetzt eine DML-Aktivität mit einem geschätzten Aufwand von weniger als 500 Timeron länger als 15 Minuten ausgeführt wird, wird ein Schwellenwertverstoßeintrag in den THVIOLATIONS-Ereignismonitor geschrieben (um anzugeben, dass gegen den Gesamtzeitschwellenwert verstoßen wurde) und Details zur DML-Aktivität werden beim Abschluss der Aktivität erfasst und an den DB2ACTIVITIES-Ereignismonitor gesendet. Mit den im DB2ACTIVITIES-Ereignismonitor erfassten Aktivitätsinformationen können Sie die weitere Untersuchung vornehmen. Sie können z. B. die Anweisung EXPLAIN auf der Abfrage ausführen und den Zugriffsplan untersuchen. Sie sollten auch die Systembelastung und die Warteschlangensteuerung zum Zeitpunkt der Aktivitätserfassung berücksichtigen, da eine lange Laufzeit ein Ergebnis von unzureichenden Systemressourcen oder des Einstellens der Aktivität in die Warteschlange sein kann. Eine lange Laufzeit weist nicht notwendigerweise auf veraltete Statistikdaten hin.

Szenario: Abbrechen aller in einer Serviceunterklasse ausgeführten Aktivitäten

Das folgende Beispiel zeigt eine gespeicherte Prozedur, die zum Abbrechen aller Aktivitäten verwendet werden kann, die zurzeit in einer bestimmten Serviceunterklasse ausgeführt werden.

Führen Sie die folgenden Schritte aus, um die Prozedur CANCELALL zu erstellen:

1. Kopieren Sie die folgende Anweisung CREATE PROCEDURE in eine Datei (zum Beispiel 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. Führen Sie den folgenden CLP-Befehl aus:

```
db2 -td@ -f cancelall.ddl
```

Nachdem die Prozedur erstellt wurde, führen Sie sie mithilfe der folgenden Anweisung aus und brechen Sie dabei beispielsweise alle Aktivitäten in der Serviceunterklasse ab, die die ID = 15 hat:

```
CALL CANCELALL( 15 )
```

Anmerkung: Die Prozedur CANCELALL muss in einer anderen Serviceunterklasse als das Ziel, das als Eingabe übergeben wird, ausgeführt werden, da sich die Prozedur andernfalls selbst abbricht.

Szenario: Trennen der Verbindung aller Anwendungen, die Aktivitäten in einer Serviceklasse zugeordnet sind oder solche Aktivitäten ausführen

Das folgende Beispiel zeigt eine gespeicherte Prozedur, die zum Trennen der Verbindung (zum Erzwingen des Endes) aller Anwendungen verwendet werden kann, die entweder einer bestimmten Serviceklasse zugeordnet sind oder zurzeit Aktivitäten in einer bestimmten Serviceklasse ausführen.

Führen Sie die folgenden Schritte aus, um die Prozedur FORCEALLINSC zu erstellen:

1. Kopieren Sie die folgende Anweisung CREATE PROCEDURE in eine Datei (zum Beispiel 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;

    -- First force any applications that have an activity running in
    -- the specified service class

    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;

    -- Now force any connections that are mapped to the service class, but which
    -- don't currently have any activities running

    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. Führen Sie den folgenden CLP-Befehl aus:

```
db2 -td@ -f forceall.ddl
```

Nachdem die Prozedur erstellt wurde, führen Sie sie mithilfe der folgenden Anweisung aus (und trennen Sie dabei beispielsweise die Verbindung aller Anwendungen, die entweder Aktivitäten in einer bestimmten Serviceklasse mit der ID = 15 zugeordnet sind oder solche Aktivitäten zurzeit ausführen):

```
CALL FORCEALLINSC( 15 )
```

Anmerkung: Die Prozedur FORCEALLINSC muss in einer anderen Serviceklasse als das Ziel, das als Eingabe übergeben wird, ausgeführt werden, da sich die Prozedur andernfalls selbst abbricht.

Szenario: Optimieren einer DB2-Workload-Management-Konfiguration, wenn Kapazitätsplanungsinformationen verfügbar sind

Wenn Sie Kapazitätsplanung durchgeführt haben, sollten Sie Informationen zu den Typen von Benutzern und ihren erwarteten Antwortzeiten haben. Mithilfe dieser Informationen können Sie Ihre DB2-Workload-Management-Konfiguration aufbauen, deren Effektivität bestimmen und sie optimieren.

Nehmen wir an, dass Sie Kapazitätsplanung durchgeführt haben und dass die Daten in der folgenden Tabelle die entsprechenden Ergebnisse für Arbeitstypen und die angestrebten Antwortzeiten darstellen:

Tabelle 63. Ergebnisse der Kapazitätsplanung

Arbeitstyp	Anwendung	Ziel	Wichtigkeit	Erwarteter Durchsatz
Auftragserfassung	orderentryapp.exe	Durchschnittliche Antwortzeit < 1 Sekunde	Hoch	10 000 (sowohl Einfügungen als auch Aktualisierungen) pro Tag
Business-Intelligence-Abfragen	businessobjects.exe	Durchschnittliche Antwortzeit < 10 Sekunden	Hoch	100 Abfragen pro Tag
Stapelverarbeitung	batchapp.exe	Maximierung des Durchsatzes	Niedrig	5000 Aktualisierungen pro Tag
Sonstige	Alle anderen Anwendungen	Beste Leistung	Niedrig	100 Aktivitäten pro Tag

Basierend auf den Daten in der Tabelle oben könnten Sie drei Serviceklassen (ORDER_ENTRY_SC, BI_QUERIES_SC und BATCH_SC) und drei Workloads (ORDER_ENTRY_WL, BI_QUERIES_WL und BATCH_WL) erstellen, um die Arbeit den Serviceklassen zuzuordnen. Nach der Erstellung der Serviceklassen und der Workloads könnten Sie einen Statistikereignismonitor erstellen, um zusammengefasste Aktivitätsinformationen wie das Histogramm der Aktivitätslaufzeit für jede Serviceklasse zu erfassen. Nehmen wir an, dass die Daten in der folgenden Tabelle einen Vergleich der durchschnittlichen täglichen Anzahl von Aktivitäten in jeder Serviceklasse (berechnet aus dem Histogramm der Aktivitätslaufzeit) mit den Volumen darstellen, die in der Kapazitätsplanungsübung vorhergesagt wurden:

Tabelle 64. Aktivitäten pro Tag

Serviceklasse	Vorhergesagte Aktivitäten pro Tag	Tatsächliche Aktivitäten pro Tag
ORDER_ENTRY_SC	10 000	9700

Tabelle 64. Aktivitäten pro Tag (Forts.)

Serviceklasse	Vorhergesagte Aktivitäten pro Tag	Tatsächliche Aktivitäten pro Tag
BI_QUERIES_SC	100	115
BATCH_SC	5000	5412
SYSDEFAULTUSERCLASS	100	85

Die beobachteten Daten zeigen, dass die Kapazitätsplanungsschätzungen präzise waren. Die Daten in der folgenden Tabelle vergleichen die durchschnittlichen Aktivitätslaufzeiten (aus dem Histogramm der Aktivitätslaufzeit) mit den bei der Kapazitätsplanung festgelegten angestrebten Antwortzeiten und zeigen, dass die Aktivitäten in der Serviceklasse BI_QUERIES_SC nicht die zugehörigen Antwortzeitziele erreichen.

Tabelle 65. Antwortzeiten

Serviceklasse	Angestrebte Antwortzeit	Tatsächliche durchschnittliche Laufzeit
ORDER_ENTRY_SC	< 1 Sekunde	0,8 Sekunden
BI_QUERIES_SC	< 10 Sekunden	30 Sekunden
BATCH_SC		2 Sekunden
SYSDEFAULTUSERCLASS		10 Minuten

Mit dem DB2-Workload-Management können Sie unterschiedliche Strategien verwenden, wenn Sie sich mit dem Problem befassen, dass Business-Intelligence-Abfragen nicht die angestrebten Antwortzeiten erreichen:

- Beschränken des gemeinsamen Zugriffs von Serviceklassen mit niedrigerer Priorität
- Zulassen, dass der Betriebssystem-Workload-Manager weniger wichtigen Serviceklassen weniger Prozessorressourcen zur Verfügung stellt
- Ändern der Agenten- und E/A-Vorablesezugriffsprioritäten für die Serviceklassen
- Verwenden einer beliebigen Kombination der vorherigen drei Methoden

Nehmen wir an, die Prozessorzeit ist die Ressource, wegen der die Business-Intelligence-Abfragen ihre Ziele nicht erreichen können. Nehmen wir weiterhin an, dass Sie den Betriebssystem-Workload-Manager verwenden, um der Serviceklasse SYSDEFAULTUSERCLASS weniger Prozessorressourcen als anderen Serviceklassen zur Verfügung zu stellen. Sie können dann zusammengefasste Aktivitätsinformationen über einen Zeitraum von mehreren Tagen erfassen, um zu beobachten, ob die Änderung der CPU-Zuordnung die erwarteten Ergebnisse bringt. Die Daten in der folgenden Tabelle zeigen einen weiteren Vergleich zwischen den angestrebten Antwortzeiten und den tatsächlichen durchschnittlichen aus den Histogrammen berechneten Laufzeiten, nachdem Sie die Änderungen über den Betriebssystem-Workload-Manager vorgenommen haben. Alle Serviceklassen erreichen jetzt ihre angestrebten Antwortzeiten. Durch die Neuordnung der Prozessorzeit haben sich die Antwortzeiten von Aktivitäten in der Serviceklasse SYSDEFAULTUSERCLASS verdoppelt.

Tabelle 66. Antwortzeiten nach der Rekonfiguration

Serviceklasse	Angestrebte Antwortzeit	Tatsächliche durchschnittliche Laufzeit
ORDER_ENTRY_SC	< 1 Sekunde	0,6 Sekunden
BI_QUERIES_SC	< 10 Sekunden	9,5 Sekunden
BATCH_SC		1,5 Sekunden
SYSDEFAULTUSERCLASS		20 Minuten

Szenario: Optimieren einer DB2-Workload-Management-Konfiguration, wenn keine Kapazitätsplanungsinformationen verfügbar sind

Mit den DB2-Workload-Management-Tools können Sie eine Workload-Management-Konfiguration entwerfen, überwachen und optimieren, selbst wenn Sie keine Kapazitätsanalysedaten für die Entwicklung der Konfiguration haben.

Nehmen wir an, dass Sie anfangs nicht wissen, welche Workloads und Serviceklassen erstellt werden müssen, weil Sie entweder die Arbeit auf dem System nicht genau kennen oder noch nicht wissen, welche Workloads für stabile Ausführungsergebnisse erforderlich sind. Nehmen wir weiter an, dass Sie wissen, dass einige Anwendungen Anforderungen bezüglich der Antwortzeit haben, aber Sie noch nicht wissen, wie viele andere Anwendungen mit solchen zeitkritischen Anwendungen um Ressourcen konkurrieren. Sie können die Workload-Management-Überwachungsfunktionen verwenden, um dies zu ermitteln.

Gehen Sie wie folgt vor, um eine DB2-Workload-Management-Konfiguration einzurichten, die Überwachungsdaten als Grundlage verwendet:

1. Klassifizieren Sie die Anwendungen, von denen Sie wissen, dass sie wichtig sind. Sie müssen diese Anwendungen eingrenzen und ihnen einen ausreichenden Anteil der Systemressourcen zuteilen.
2. Für die restliche Arbeit erfassen Sie Statistikdaten für die größten Aktivitäten, da diese Aktivitäten die größte Auswirkung pro Aktivität auf das System haben.
3. Analysieren Sie die in Schritt 2 erfassten Aktivitätsinformationen.
4. Wiederholen Sie Schritt 1 bis 3 für den noch nicht klassifizierten Teil der Arbeit. Wiederholen Sie diesen Schritt, bis Sie wissen, dass es sich nicht mehr lohnt, die verbleibende noch nicht klassifizierte Arbeit weiter zu klassifizieren.

Die folgenden Abschnitte enthalten Informationen zur Durchführung dieser Schritte.

Schritt 1. Eingrenzen der als wichtig bekannten Anwendungen und Zuteilen der angemessenen Ressourcenmenge für diese Anwendungen

Nehmen wir an, dass Sie zwei wichtige Business-Intelligence-Anwendungen, BI1 und BI2, haben und dass Sie die Antwortzeiten für diese Anwendungen minimieren müssen. Sie können für diese beiden Anwendungen Workloads erstellen und sie einer Serviceklasse namens MOSTIMPORTANT zuordnen, für die Sie Systemressourcen zuordnen können.

Unter dem Betriebssystem AIX können Sie AIX Workload Manager verwenden, um eine Serviceklasse namens MOSTIMPORTANT zu erstellen und dieser Serviceklasse eine garantierte Menge an Ressourcen zuzuordnen.

Auf dem DB2-Datenserver erstellen Sie die erforderlichen Serviceklassen und Workloads wie folgt:

```
CREATE SERVICE CLASS MOSTIMPORTANT OUTBOUND CORRELATOR 'MOSTIMPORTANT'  
CREATE WORKLOAD BI1WORKLOAD APPLNAME ('BI1') SERVICE CLASS MOSTIMPORTANT  
CREATE WORKLOAD BI2WORKLOAD APPLNAME ('BI2') SERVICE CLASS MOSTIMPORTANT
```

Für die Zwecke dieses Beispiels nehmen wir an, dass noch ein beträchtlicher Teil der Systemauslastung nicht berücksichtigt ist, selbst nachdem Sie die bekannten Anwendungen berücksichtigt haben. Sie müssen daher diese Arbeit besser verstehen und möglicherweise steuern.

Schritt 2. Erfassen von Statistikdaten für die größten Aktivitäten in der restlichen nicht klassifizierten Arbeit

Eine lang laufende Aktivität hat eine größere individuelle Auswirkung auf das System als eine Aktivität mit kurzer Laufzeit, da die lang laufende Aktivität über einen längeren Zeitraum Systemressourcen belegt. Das Erfassen von Informationen zu lang laufenden Aktivitäten erzeugt jedoch keinen höheren Systemaufwand als das Erfassen von Informationen zu Aktivitäten mit kurzer Laufzeit. Die beste Methode zum Erfassen von Informationen zum Großteil der Arbeit ist daher, zuerst Informationen zu den am längsten laufenden Aktivitäten zu erfassen.

Beginnen Sie mit dem Erfassen von Aktivitätsinformationen, indem Sie zuerst eine Aktivitätslaufzeit festlegen, ab der Sie Aktivitätsinformationen erfassen wollen. Sie können dies vereinfachen, indem Sie die Erfassung für einen Teil der nicht klassifizierten Aktivitäten, z. B. 30 %, auswählen und dann das Histogramm der Aktivitätslaufzeit für diese Aktivitäten beobachten. Lassen Sie das System arbeiten, sodass die Statistikdaten aktualisiert werden. Führen Sie dann die Prozedur WLM_COLLECT_STATS aus, um die Statistikdaten an einen aktiven Statistikereignismonitor zu senden.

Mit der folgenden Abfrage können Sie das Histogramm der Aktivitätslaufzeit für die Serviceklasse SYSDEFAULTUSERCLASS in Form einer Tabelle erhalten, die den Anteil der Gesamtaktivitäten darstellt, der in den jeweiligen Laufzeitbereich fiel. Bei dieser Abfrage wird davon ausgegangen, dass die Datenbank nicht mehrere Member hat.

```
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
0.000	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

Die folgende Abbildung zeigt die Ergebnisse der obigen Abfrage in Form eines Diagramms:

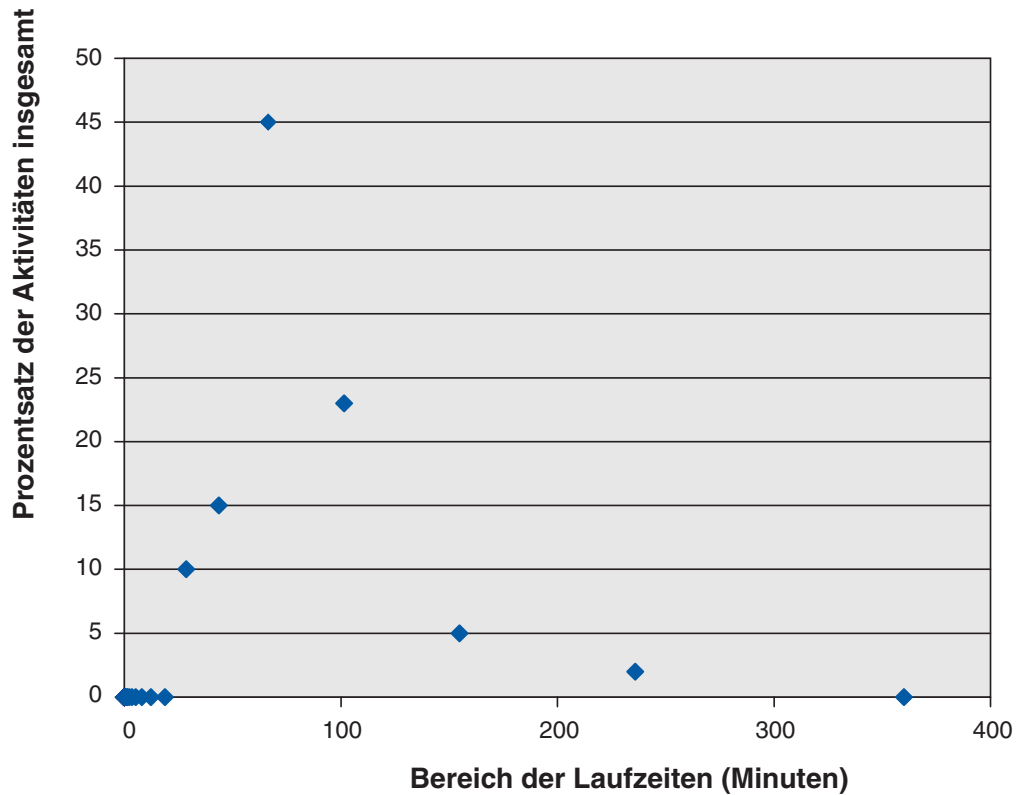


Abbildung 56. Histogramm der Aktivitätslaufzeit nicht klassifizierter Aktivitäten

In diesem Beispiel fallen 30 % der Aktivitäten in den Bereich einer Laufzeit von 101 Minuten oder mehr. Zum Erfassen von Informationen zu diesen Aktivitäten erstellen Sie einen Aktivitätslaufzeitschwellenwert von 100 Minuten mit den Optionen CONTINUE und COLLECT ACTIVITY DATA (siehe folgendes Beispiel). Wenn gegen diesen Schwellenwert verstoßen wird, werden Aktivitätsinformationen an einen aktiven Aktivitätsereignismonitor gesendet.

```
CREATE THRESHOLD COLLECTLONGESTRUNNING30PERCENT
FOR SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
ACTIVITIES ENFORCEMENT DATABASE ENABLE
WHEN ACTIVITYTOTALTIME > 100 MINUTES COLLECT ACTIVITY DATA CONTINUE
```

Lassen Sie das System arbeiten, damit Daten erfasst werden.

Wenn der Systemaufwand für das Erfassen von Informationen zu den 30 % am längsten laufenden Aktivitäten akzeptabel ist, dann können Sie erlauben, dass die Datenerfassung einige Stunden oder einige Tage fortgesetzt wird. Mit den erfassten Daten können Sie ermitteln, welche Benutzer und Anwendungen die am längsten laufenden noch nicht klassifizierten 30 % der DML-Aktivitäten erzeugen. Unter diesen Aktivitäten könnten einige sein, die zeitkritisch sind. Sie könnten auch einige Überraschungen erleben, z. B. Anwendungen mit niedriger Priorität, die eine beträchtliche Anzahl großer Aktivitäten ausführen. Wenn Sie mit der Erfassung und der Analyse der Daten fertig sind, können Sie den Schwellenwert löschen.

Schritt 3. Analysieren der im vorherigen Schritt erfassten Aktivitätsinformationen

Sie können die Informationen, die Sie im vorherigen Schritt zu Aktivitäten erfasst haben, nach der übergebenden Anwendung analysieren. Sie könnten die folgende Abfrage absetzen:

```

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

Eine Analyse der Aktivitäten nach der übergebenden Anwendung zeigt, dass eine große Zahl der am längsten laufenden Aktivitäten von der Anwendung UNIMPOR-TANTAPP, einer relativ unwichtigen Anwendung, übergeben wurden. Sie können eine Workload verwenden, um diese Anwendung von den anderen nicht klassifi-zierten Anwendungen abzugrenzen und sie einer Serviceklasse namens BESTEF-FORT zuordnen, die Ressourcen nur dann erhält, wenn der Ressourcenbedarf aller anderen Aktivitäten erfüllt ist.

Gemäß den obigen Ergebnissen scheinen die verbleibenden Anwendungen in der Standardserviceklasse wenige große Aktivitäten zu übergeben. Vielleicht finden Sie es lohnend, die Erfassung von Aktivitäten zu wiederholen, die in der Standardser-viceklasse ausgeführt werden, ohne die Erfassung auf lang laufende Aktivitäten einzuschränken.

Schritt 4. Wiederholen von Schritt 1 bis 3 für den noch nicht klassifizierten Teil der Arbeit, bis es nicht mehr lohnt, die verbleibende nicht klassifizierte Arbeit weiter zu klassifizieren

Jetzt, da die zwei wichtigen Anwendungen in der Serviceklasse MOSTIMPOR-TANT und die unwichtige Anwendung in der Serviceklasse BESTEFFORT ausge-führt werden, wird viel weniger Arbeit in der Standardbenutzerserviceklasse aus-geführt. In dieser Situation kann es weniger aufwendig sein, Informationen zu jeder Aktivität in dieser Serviceklasse zu erfassen. Alternativ brauchen Sie mögli-cherweise die Arbeit nicht weiter zu unterteilen und können an diesem Punkt auf-hören. Nehmen wir an, Sie wollen Informationen zu den verbleibenden Aktivitäten erfassen, für den Fall, dass die verbleibende Arbeit mit Überraschungen aufwartet. Sie können dies erreichen, indem Sie COLLECT ACTIVITY DATA für die Standard-benutzerserviceklasse angeben und einen Aktivitätsereignismonitor erstellen:

```

ALTER SERVICE CLASS SYSDEFAULTSUBCLASS UNDER SYSDEFAULTUSERCLASS
COLLECT ACTIVITY DATA ON COORDINATOR WITHOUT DETAILS

```

Lassen Sie das System arbeiten, damit Daten erfasst werden. Sie können die Ergeb-nisse wie in Schritt 3 analysieren.

```

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

Die Ergebnisse zeigen, dass die Anwendung ONLYSMALL die Mehrheit der nicht klassifizierten Aktivitäten erzeugt. Da diese Anwendung nicht in den Ergebnissen enthalten war, als Sie Informationen zu den größten Aktivitäten erfasst haben, können Sie annehmen, dass ONLYSMALL während des Zeitraums der Datenerfassung keine großen Abfragen erzeugte.

Kapitel 8. Referenz

Prozeduren und Tabellenfunktionen

WLM_CANCEL_ACTIVITY - Aktivität abbrechen

Diese Prozedur bricht eine gegebene Aktivität ab. Bei Abbruch wird eine Fehlermeldung an die Anwendung zurückgegeben, welche die abgebrochene Aktivität übergeben hatte.

Syntax

```
►►—WLM_CANCEL_ACTIVITY—(—anwendungskennung—,—uow-id—,—aktivitäts-id—)————►◄
```

Das Schema ist SYSPROC.

Prozedurparameter

anwendungskennung

Ein Eingabeargument des Typs BIGINT, das die Anwendungskennung angibt, deren Aktivität abgebrochen werden soll. Wird dieses Argument nicht angegeben, wird keine Aktivität gefunden und die Fehlermeldung SQL4702N wird mit dem SQLSTATE-Wert 5U035 zurückgegeben.

uow-id

Ein Eingabeargument des Typs INTEGER, das die UOW-ID der Aktivität angibt, die abgebrochen werden soll. Wird dieses Argument nicht angegeben, wird keine Aktivität gefunden und die Fehlermeldung SQL4702N wird mit dem SQLSTATE-Wert 5U035 zurückgegeben.

aktivitäts-id

Ein Eingabeargument des Typs INTEGER, das die Aktivitäts-ID angibt, die die Aktivität innerhalb der UOW (Unit of Work) eindeutig angibt, die abgebrochen werden soll. Wird dieses Argument nicht angegeben, wird keine Aktivität gefunden und die Fehlermeldung SQL4702N wird mit dem SQLSTATE-Wert 5U035 zurückgegeben.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Ein Administrator kann die Tabellenfunktion `WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES` verwenden, um die Anwendungskennung, die UOW-ID und die Aktivitäts-ID einer Aktivität zu finden. Wenn Sie eine Aktivität mit der Anwendungskennung 1, UOW-ID 2 und Aktivitäts-ID 3 abbrechen möchten, müssen Sie Folgendes eingeben:

```
CALL WLM_CANCEL_ACTIVITY(1, 2, 3)
```

Hinweise zur Verwendung

- Wenn keine Aktivität gefunden werden kann, wird die Fehlermeldung `SQL4702N` mit dem `SQLSTATE`-Wert `5U035` zurückgegeben.
- Wenn die Aktivität nicht abgebrochen werden kann, weil sie sich nicht im korrekten Status befindet (nicht initialisiert), wird die Fehlermeldung `SQL4703N` (Ursachencode 1) mit dem `SQLSTATE`-Wert `5U016` zurückgegeben.
- Wenn die Aktivität erfolgreich abgebrochen wird, wird die Fehlermeldung `SQL4725N` mit dem `SQLSTATE`-Wert `57014` an die abgebrochene Anwendung zurückgegeben.
- Wenn der Koordinator zum Zeitpunkt des Abbruchs eine Anforderung für eine andere Aktivität verarbeitet oder inaktiv ist, wird die Aktivität in den Status `CANCEL_PENDING` versetzt. Sie wird erst abgebrochen, wenn der Koordinator die nächste Anforderung für die Aktivität verarbeitet.

WLM_CAPTURE_ACTIVITY_IN_PROGRESS - Aktivitätsinformationen für Aktivitätsereignismonitor erfassen

Die Prozedur `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` stellt Informationen zu einer bestimmten Aktivität zusammen und schreibt die Informationen in den aktiven Aktivitätsereignismonitor.

Wenn Sie diese Prozedur auf eine Aktivität anwenden, die über untergeordnete Aktivitäten verfügt, generiert diese Prozedur rekursiv bis zur untersten Ebene einen Datensatz für jede untergeordnete Aktivität. Diese Informationen werden erfasst und gesendet, sobald diese Prozedur aufgerufen wird. Sie wartet nicht, bis die Ausführung der übergeordneten Aktivität abgeschlossen ist. Der Datensatz der Aktivität im Ereignismonitor wird als Teildatensatz gekennzeichnet.

Syntax

```
▶▶—WLM_CAPTURE_ACTIVITY_IN_PROGRESS—(—anwendungskennung—, —————▶  
▶—uow-id—, —aktivitäts-id—)—————▶▶
```

Das Schema ist `SYSPROC`.

Prozedurparameter

Wenn Sie nicht alle nachfolgenden Parameter angeben, wird keine Aktivität gefunden und die Fehlermeldung `SQL4702N` wird mit dem `SQLSTATE`-Wert `5U035` zurückgegeben.

anwendungskennung

Ein Eingabeargument des Typs `BIGINT`, das die Anwendungskennung angibt, deren Aktivität erfasst werden soll.

uow-id

Ein Eingabeargument des Typs INTEGER, das die UOW-ID der Aktivität angibt, die erfasst werden soll.

aktivitäts-id

Ein Eingabeargument des Typs INTEGER, das die Aktivitäts-ID angibt, die die Aktivität innerhalb der UOW (Unit of Work) eindeutig angibt, deren Informationen erfasst werden sollen.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Eine einzelne Prozedur MYSCHEMA.MYSLOWSTP wird möglicherweise langsamer ausgeführt als normal. Ein Benutzer beklagt sich und der Administrator möchte nun die Ursache der Leistungsminderung untersuchen. Die Untersuchung während der Ausführung der gespeicherten Prozedur ist nicht praktikabel. Deshalb beschließt der Administrator, die Aktivität der gespeicherten Prozedur und aller darin verschachtelter Aktivitäten zu erfassen.

Wenn beispielsweise ein Ereignismonitor für DB2-Aktivitäten mit dem Namen DB2ACTIVITIES vorhanden ist und aktiviert wurde, kann der Administrator die Funktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES verwenden, um die Anwendungskennung, die UOW-ID und die Aktivitäts-ID für den Aufruf dieser gespeicherten Prozedur abzurufen. Wenn die Aktivität mit einer Anwendungskennung von 1, eine UOW-ID von 2 und eine Aktivitäts-ID von 3 angegeben wird, kann der Administrator wie folgt einen Aufruf für WLM_CAPTURE_ACTIVITY_IN_PROGRESS absetzen:

```
CALL WLM_CAPTURE_ACTIVITY_IN_PROGRESS(1,2,3)
```

Sobald die Prozedur abgeschlossen ist, kann der Administrator die folgende Tabellenfunktion verwenden, um zu ermitteln, an welcher Position die Aktivität die Zeit verbracht hat. Die Funktion ruft die Informationen aus dem Ereignismonitor the DB2ACTIVITIES ab.

```
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, CHILDSTMT.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

```

Die folgende Beispielabfrage verwendet die Tabellenfunktion:

```

SELECT * FROM TABLE(SHOWCAPTUREDACTIVITY(1, 2, 3))
AS ACTS ORDER BY UOW_ID, ACTIVITY_ID

```

Hinweise zur Verwendung

Wenn kein aktiver Aktivitätsereignismonitor vorhanden ist, wird die Fehlernachricht SQL1633W mit dem SQLSTATE-Wert 01H53 zurückgegeben.

Aktivitätsinformationen werden nur im Koordinatormember der Aktivität erfasst.

WLM_COLLECT_STATS - Workload-Management-Statistikdaten erfassen und zurücksetzen

Die Prozedur WLM_COLLECT_STATS stellt Statistikdaten für Serviceklassen, Workloads, Arbeitsklassen und Schwellenwertwarteschlangen zusammen und schreibt die Daten in den Statistikereignismonitor. Die Prozedur setzt auch die Statistikdaten für Serviceklassen, Workloads, Arbeitsklassen und Schwellenwertwarteschlangen zurück.

Wenn kein Statistikereignismonitor aktiv ist, setzt die Prozedur nur die Statistikdaten zurück.

Syntax

```

▶▶ WLM_COLLECT_STATS ( ( wartestatus , zeitmarke_der_statistikdaten ) ) ▶▶

```

Das Schema ist SYSPROC.

Prozedurparameter

wartestatus

Ein optionales Eingabeargument des Typs CHAR, das angibt, ob diese Prozedur sofort nach der Initialisierung einer Statistikerfassung und Zurücksetzung

zurückkehrt. Wurde 'Y' angegeben, kehrt diese Prozedur nicht zurück, bevor alle Statistikdaten geschrieben und mittels Flushoperation in die Ereignismonitorstabellen übertragen wurden. Ansonsten kehrt diese Prozedur unverzüglich nach der Initialisierung einer Statistikerfassung und Zurücksetzung zurück.

zeitmarke_der_statistikdaten

Ein optionales Ausgabeargument des Typs `TIMESTAMP`, das den Zeitmarkenwert für den Beginn der Statistikerfassung zurückgibt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht `EXECUTE` für die Routine
- Berechtigung `DATAACCESS`
- Berechtigung `DBADM`
- Berechtigung `SQLADM`
- Berechtigung `WLMADM`

Standardzugriffsrecht für PUBLIC

Ohne

Beispiele

Beispiel 1: `WLM_COLLECT_STATS` zur Initialisierung der Statistikerfassung und Zurücksetzung aufrufen.

```
CALL WLM_COLLECT_STATS()
```

Nachfolgend finden Sie ein Beispiel für die Ausgabe dieser Abfrage.

```
Rückgabestatus = 0
```

Beispiel 2: `WLM_COLLECT_STATS` zur Erfassung und Zurücksetzung von Statistikdaten aufrufen, wobei die Prozedur erst nach dem Schreiben aller Daten in die Statistikereignismonitorstabellen zurückkehrt.

```
CALL WLM_COLLECT_STATS('Y', ::collect_timestamp)
```

Nachfolgend finden Sie ein Beispiel für die Ausgabe dieser Abfrage.

```
Rückgabestatus = 0
```

Beispiel 3: `WLM_COLLECT_STATS` zur Erfassung und Zurücksetzung von Statistikdaten aufrufen, während ein anderer Aufruf in Bearbeitung ist.

```
CALL WLM_COLLECT_STATS()
```

Nachfolgend finden Sie ein Beispiel für die Ausgabe dieser Abfrage.

```
SQL1632W Die Anforderung zum Erfassen und Zurücksetzen von Statistiken wurde ignoriert, da eine andere Anforderung zum Erfassen und Zurücksetzen von Statistiken bereits in Bearbeitung ist.
```

Hinweise zur Verwendung

Die Prozedur `WLM_COLLECT_STATS` führt dieselben Operationen zum Erfassen (Senden der Statistiken an den aktiven Statistikereignismonitor) und Zurücksetzen aus, die in dem mit dem Datenbankkonfigurationsparameter `wlm_collect_int` definierten Intervall automatisch auftreten.

Wenn die Prozedur zu einem Zeitpunkt aufgerufen wird, an dem eine andere Anforderung zum Erfassen und Zurücksetzen bereits in Bearbeitung ist (z. B. wenn die Prozedur zu einem Zeitpunkt aufgerufen wird, an dem ein anderer Aufruf der Prozedur oder eine automatische Erfassung ausgeführt wird), wird eine Warnung (SQL1632W mit dem SQLSTATE-Wert 01H53) zurückgegeben und die Anforderung wird ignoriert.

Im asynchronen Modus startet die Prozedur WLM_COLLECT_STATS nur den Prozess der Erfassung und Zurücksetzung. Die Prozedur kehrt möglicherweise zum Aufrufenden zurück, bevor alle Statistikdaten in den aktiven Statistikereignismonitor geschrieben wurden. Je nach dem, wie schnell die Erfassung und Zurücksetzung der Statistikdaten erfolgt, wird der Aufruf der Prozedur WLM_COLLECT_STATS (welche selbst eine Aktivität ist) in den Statistikdaten entweder für das vorherige Erfassungsintervall oder für das gerade gestartete neue Erfassungsintervall protokolliert.

Im synchronen Modus kehrt die Prozedur WLM_COLLECT_STATS nicht zurück, bevor die Erfassung der Statistikdaten abgeschlossen ist und alle Statistikdaten in die Tabellen eines aktiven Statistikereignismonitors geschrieben wurden. Die Zeitmarke, an der die Statistikerfassung begann, wird über den Ausgabeparameter *statistics_timestamp* zurückgegeben.

WLM_GET_ACTIVITY_DETAILS - Detaillierte Informationen zu einer bestimmten Aktivität zurückgeben

Diese Funktion gibt detaillierte Informationen zu einer bestimmten Aktivität zurück, die durch ihre Anwendungskennung, UOW-ID und Aktivitäts-ID angegeben wird. Diese Informationen enthalten Details zu allen Schwellenwertverstößen der Aktivität.

Anmerkung: Diese Tabellenfunktion wird nicht weiter unterstützt und wurde ersetzt durch: MON_GET_ACTIVITY_DETAILS (Tabellenfunktion).

Diese Funktion gibt Basisstatistikdaten für mindestens eine Serviceunterklasse zurück.

Syntax

```
►► WLM_GET_ACTIVITY_DETAILS (—anwendungskennung—, —uow-id—, —————►  
► —aktivitäts-id—, —member—) —————►◀
```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

anwendungskennung

Ein Eingabeargument des Typs BIGINT, das eine gültige Anwendungskennung angibt. Wird dieses Argument nicht angegeben, gibt diese Funktion keine Zeilen zurück. Wird das Argument nicht angegeben, wird der Fehler SQL171N zurückgegeben.

uow-id

Ein Eingabeargument des Typs INTEGER, das eine gültige UOW-ID angibt, die innerhalb der Anwendung eindeutig ist. Wird dieses Argument nicht angege-

ben, gibt diese Funktion keine Zeilen zurück. Wird das Argument nicht angegeben, wird der Fehler SQL171N zurückgegeben.

aktivitäts-id

Ein Eingabeargument des Typs INTEGER, das eine gültige Aktivitäts-ID angibt, die innerhalb der UOW (Unit of Work) eindeutig ist. Wird dieses Argument nicht angegeben, gibt diese Funktion keine Zeilen zurück. Wird das Argument nicht angegeben, wird der Fehler SQL171N zurückgegeben.

member

Ein Eingabeargument des Typs INTEGER, das beim Aufruf dieser Funktion eine gültige Membernummer in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Datenbankmember oder -2 für alle Datenbankmember an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Detaillierte Informationen zu einer einzelnen Aktivität können Sie mithilfe der Tabellenfunktion WLM_GET_ACTIVITY_DETAILS abrufen. Diese Tabellenfunktion gibt Aktivitätsinformationen wie Name/Wert-Paare für jedes Member zurück. Dieses Beispiel ist so beschränkt, dass nur eine aus elf Elementen bestehende Untergruppe der Name/Wert-Paare für jedes Member für eine Aktivität gezeigt wird, die durch eine Anwendungskennung von 1, eine UOW-ID von 1 und eine Aktivitäts-ID von 5 angegeben wird. Eine vollständige Liste der Name/Wert-Paare finden Sie in Tabelle 68 auf Seite 400 und Tabelle 69 auf Seite 402.

```
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
```

Nachfolgend finden Sie ein Beispiel für die Ausgabe dieser Abfrage.

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

Hinweis zur Verwendung:

Der ACTIVITY_STATE-Wert QUEUED bedeutet, dass die Koordinatoraktivität versucht, die Schwellenwerttickets über einen Fernprozeduraufruf abzurufen, aber bisher keine Antwort erhalten hat. Wenn dieser Status angezeigt wird, kann dies bedeuten, dass die Aktivität von WLM in die Warteschlange gestellt wurde. Wird er nur kurzzeitig angezeigt, kann dies darauf hindeuten, dass die Aktivität gerade ihre Tickets abrufen. Wenn Sie sich ein genaueres Bild darüber machen wollen, ob die Aktivität tatsächlich in die Warteschlange gestellt wird, können Sie ermitteln, welcher Agent die Aktivität verarbeitet (mithilfe der Tabellenfunktion WLM_GET_SERVICE_CLASS_AGENTS) und feststellen, ob das Ereignisobjekt dieses Agenten in der Katalogpartition den Wert WLM_QUEUE aufweist.

Zurückgegebene Informationen

Tabelle 67. Für WLM_GET_ACTIVITY_DETAILS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
NAME	VARCHAR(256)	Elementname. Mögliche Werte finden Sie in Tabelle 68 und Tabelle 69 auf Seite 402.
VALUE	VARCHAR(1024)	Elementwerte. Mögliche Werte finden Sie in Tabelle 68 und Tabelle 69 auf Seite 402.

Tabelle 68. Zurückgegebene Elemente

Elementname	Beschreibung
ACTIVITY_ID	activity_id - Aktivitäts-ID (Monitorelement)
ACTIVITY_STATE	activity_state - Aktivitätsstatus (Monitorelement)
ACTIVITY_TYPE	activity_type - Aktivitätstyp (Überwachungselement)
APPLICATION_HANDLE	application_handle - Anwendungskennung (Monitorelement)

Table 68. Zurückgegebene Elemente (Forts.)

Elementname	Beschreibung
COORD_PARTITION_NUM	coord_partition_num - Koordinatorpartitionsnummer (Monitorelement)
DATABASE_WORK_ACTION_SET_ID	Wurde diese Aktivität einem Arbeitsaktionsset zugeordnet, das auf die Datenbank angewendet wurde, enthält diese Spalte die ID des Arbeitsaktionssets. Diese Spalte enthält den Wert 0, wenn die Aktivität nicht einem Arbeitsaktionsset zugeordnet wurde, das auf die Datenbank angewendet wurde.
DATABASE_WORK_CLASS_ID	Wurde diese Aktivität einem Arbeitsaktionsset zugeordnet, das auf die Datenbank angewendet wurde, enthält diese Spalte die ID der Arbeitsklasse dieser Aktivität. Diese Spalte enthält den Wert 0, wenn die Aktivität nicht einem Arbeitsaktionsset zugeordnet wurde, das auf die Datenbank angewendet wurde.
EFFECTIVE_ISOLATION	effective_isolation - Effektive Isolation (Monitorelement)
EFFECTIVE_LOCK_TIMEOUT	effective_lock_timeout - Effektives Zeitlimit für Sperren (Monitorelement)
EFFECTIVE_QUERY_DEGREE	effective_query_degree - Effektiver Abfrageparallelitätsgrad (Monitorelement)
ENTRY_TIME	entry_time - Eintrittszeit (Monitorelement)
INVOCATION_ID	invocation_id - Aufruf-ID (Monitorelement)
LAST_REFERENCE_TIME	last_reference_time - Letzte Verweiszeit (Monitorelement)
LOCAL_START_TIME	local_start_time - Lokale Startzeit (Monitorelement)
NESTING_LEVEL	nesting_level - Verschachtelungsebene (Monitorelement)
PACKAGE_NAME	package_name - Paketname (Monitorelement)
PACKAGE_SCHEMA	package_schema - Paketschema (Monitorelement)
PACKAGE_VERSION_ID	package_version_id - Paketversion (Monitorelement)
PARENT_ACTIVITY_ID	parent_activity_id - ID der übergeordneten Aktivität (Monitorelement)
PARENT_UOW_ID	Eindeutige Kennung der UOW (Unit of Work) innerhalb einer Anwendung. Verweist auf die UOW, in der die übergeordnete Aktivität dieser Aktivität ursprünglich gestartet wurde. Gibt eine leere Zeichenfolge zurück, wenn die Aktivität keine übergeordnete Aktivität aufweist oder in einem fernen Member stattfindet.
QUERY_COST_ESTIMATE	query_cost_estimate - Geschätzter Abfrageaufwand (Monitorelement)
ROUTINE_ID	routine_id - Routinen-ID (Monitorelement)
ROWS_FETCHED	rows_fetched - Abgerufene Zeilen (Monitorelement)
ROWS_MODIFIED	rows_modified - Modifizierte Zeilen (Monitorelement)
SECTION_NUMBER	section_number - Abschnittsnummer (Monitorelement)
SERVICE_CLASS_ID	service_class_id - Serviceklassen-ID (Monitorelement)
SERVICE_CLASS_WORK_ACTION_SET_ID	Wurde diese Aktivität einem Arbeitsaktionsset zugeordnet, das auf eine Serviceklasse angewendet wurde, enthält diese Spalte die ID des Arbeitsaktionssets. Diese Spalte enthält den Wert 0, wenn die Aktivität nicht einem Arbeitsaktionsset zugeordnet wurde, das auf eine Serviceklasse angewendet wurde.

Tabelle 68. Zurückgegebene Elemente (Forts.)

Elementname	Beschreibung
SERVICE_CLASS_WORK_CLASS_ID	Wurde diese Aktivität einem Arbeitsaktionsset zugeordnet, das auf eine Serviceklasse angewendet wurde, enthält diese Spalte die ID der Arbeitsklasse dieser Aktivität. Diese Spalte enthält den Wert 0, wenn die Aktivität nicht einem Arbeitsaktionsset zugeordnet wurde, das auf eine Serviceklasse angewendet wurde.
STMT_PKG_CACHE_ID	stmt_pkgcache_id - Anweisungspaketcache-ID (Monitorelement)
STMT_TEXT	stmt_text - Text der SQL-Anweisung (Monitorelement)
SYSTEM_CPU_TIME	system_cpu_time - System-CPU-Zeit (Monitorelement)
UOW_ID	uow_id - UOW-ID (Monitorelement)
USER_CPU_TIME	user_cpu_time - Benutzer-CPU-Zeit (Monitorelement)
UTILITY_ID	utility_id - Dienstprogramm-ID (Monitorelement)

Wichtig: Die Tabellenfunktion WLM_GET_ACTIVITY_DETAILS zeigt nur die Schwellenwerte an, die derzeit auf eine Aktivität angewendet werden.

Die folgenden Elemente werden nur zurückgegeben, wenn die entsprechenden Schwellenwerte auf die Aktivität angewendet werden.

Tabelle 69. Zurückgegebene Elemente (sofern zutreffend)

Elementname	Beschreibung
ACTIVITYTOTALTIME_THRESHOLD_ID	activitytotaltime_threshold_id - Kennung des Schwellenwerts für Gesamtzeit der Aktivität (Monitorelement)
ACTIVITYTOTALTIME_THRESHOLD_VALUE	activitytotaltime_threshold_value - Wert des Schwellenwerts für Gesamtzeit der Aktivität (Monitorelement)
ACTIVITYTOTALTIME_THRESHOLD_VIOLATED	activitytotaltime_threshold_violated - Verstoß gegen den Schwellenwert für Gesamtzeit der Aktivität (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_ID	concurrentdbcoordactivities_db_threshold_id - Kennung des Datenbankschwellenwerts für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_QUEUED	concurrentdbcoordactivities_db_threshold_queued - Durch Datenbankschwellenwert für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VALUE	concurrentdbcoordactivities_db_threshold_value - Wert des Datenbankschwellenwerts für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_DB_THRESHOLD_VIOLATED	concurrentdbcoordactivities_db_threshold_violated - Verstoß gegen den Datenbankschwellenwert für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_ID	concurrentdbcoordactivities_subclass_threshold_id - Kennung des Schwellenwerts für Serviceunterklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_QUEUED	'Yes' gibt an, dass die Aktivität durch den Schwellenwert CONCURRENTDBCOORDACTIVITIES_SUBCLASS in eine Warteschlange gestellt wurde. Der Wert 'No' gibt an, dass die Aktivität nicht in die Warteschlange gestellt wurde.

Tabelle 69. Zurückgegebene Elemente (sofern zutreffend) (Forts.)

Elementname	Beschreibung
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_VALUE	concurrentdbcoordactivities_subclass_threshold_value - Schwellenwert für Serviceunterklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)
CONCURRENTDBCOORDACTIVITIES_SUBCLASS_THRESHOLD_VIOLATED	'Yes' gibt an, dass die Aktivität gegen den Schwellenwert CONCURRENTDBCOORDACTIVITIES_SUBCLASS verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_ID	Die ID des Schwellenwerts CONCURRENTDBCOORDACTIVITIES_SUPERCLASS, der auf die Aktivität angewendet wurde.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_QUEUED	'Yes' gibt an, dass die Aktivität durch den Schwellenwert CONCURRENTDBCOORDACTIVITIES_SUPERCLASS in eine Warteschlange gestellt wurde.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_VALUE	Die Obergrenze des Schwellenwerts CONCURRENTDBCOORDACTIVITIES_SUPERCLASS, der auf die Aktivität angewendet wurde.
CONCURRENTDBCOORDACTIVITIES_SUPERCLASS_THRESHOLD_VIOLATED	'Yes' gibt an, dass die Aktivität gegen den Schwellenwert CONCURRENTDBCOORDACTIVITIES_SUPERCLASS verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_ID	Die ID des Schwellenwerts CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET, der auf die Aktivität angewendet wurde.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_QUEUED	'Yes' gibt an, dass die Aktivität durch den Schwellenwert CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET in eine Warteschlange gestellt wurde. Der Wert 'No' gibt an, dass die Aktivität nicht in die Warteschlange gestellt wurde.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VALUE	Die Obergrenze des Schwellenwerts CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET, der auf die Aktivität angewendet wurde.
CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET_THRESHOLD_VIOLATED	'Yes' gibt an, dass die Aktivität gegen den Schwellenwert CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_ID	Die Kennung (ID) des Schwellenwerts für CONCURRENTWORKLOADACTIVITIES, der auf die Aktivität angewendet wurde.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VALUE	Die Obergrenze des Schwellenwerts für CONCURRENTWORKLOADACTIVITIES, der auf die Aktivität angewendet wurde.
CONCURRENTWORKLOADACTIVITIES_THRESHOLD_VIOLATED	'Yes' gibt an, dass die Aktivität gegen den Schwellenwert CONCURRENTWORKLOADACTIVITIES verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.
ESTIMATEDSQLCOST_THRESHOLD_ID	estimatedsqlcost_threshold_id - Kennung des Schwellenwerts für geschätzten SQL-Aufwand (Monitorelement)

Tabelle 69. Zurückgegebene Elemente (sofern zutreffend) (Forts.)

Elementname	Beschreibung
ESTIMATEDSQLCOST_THRESHOLD_VALUE	estimatedsqlcost_threshold_value - Wert des Schwellenwerts für geschätzten SQL-Aufwand (Monitorelement)
ESTIMATEDSQLCOST_THRESHOLD_VIOLATED	estimatedsqlcost_threshold_violated - Verstoß gegen den Schwellenwert für geschätzten SQL-Aufwand (Monitorelement)
SQLROWSRETURNED_THRESHOLD_ID	sqlrowsreturned_threshold_id - Kennung des Schwellenwerts für zurückgegebene gelesene SQL-Zeilen (Monitorelement)
SQLROWSRETURNED_THRESHOLD_VALUE	sqlrowsreturned_threshold_value - Wert des Schwellenwerts für zurückgegebene gelesene SQL-Zeilen (Monitorelement)
SQLROWSRETURNED_THRESHOLD_VIOLATED	sqlrowsreturned_threshold_violated - Verstoß gegen den Schwellenwert für zurückgegebene gelesene SQL-Zeilen (Monitorelement)
SQLTEMPSPACE_THRESHOLD_ID	sqltempespace_threshold_id - Kennung des Schwellenwerts für temporären SQL-Speicherbereich (Monitorelement)
SQLTEMPSPACE_THRESHOLD_VALUE	sqltempespace_threshold_value - Wert des Schwellenwerts für temporären SQL-Speicherbereich (Monitorelement)
SQLTEMPSPACE_THRESHOLD_VIOLATED	sqltempespace_threshold_violated - Verstoß gegen den Schwellenwert für temporären SQL-Speicherbereich (Monitorelement)

WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten von Schwellenwertwarteschlangen zurückgeben

Die Funktion WLM_GET_QUEUE_STATS gibt Basisstatistikdaten für mindestens eine Schwellenwertwarteschlange in allen aktiven Membern zurück. Diese Funktion gibt für jede Schwellenwertwarteschlange eine Zeile mit Statistikdaten zurück.

Syntax

```

▶▶—WLM_GET_QUEUE_STATS—(—schwellenwertvergleichselement—,—schwellenwertdomäne—,——————▶
▶—schwellenwertname—,—schwellenwert-id—)——————▶▶

```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

schwellenwertvergleichselement

Ein Eingabeargument des Typs VARCHAR(27), das ein Schwellenwertvergleichselement angibt. Die folgenden Werte sind möglich:

CONCDBC

Schwellenwert für gleichzeitig ablaufende Datenbankkoordinatoraktivitäten

DBCONN

Schwellenwert für die Gesamtzahl der Verbindungen zu Datenbankmembern

SCCONN

Schwellenwert für die Gesamtzahl der Verbindungen zu Serviceklassenmembern

Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Schwellenwerte zurückgegeben, die die anderen Kriterien erfüllen.

Die Werte für *schwellenwertvergleichselement* entsprechen denen der Spalte THRESHOLDPREDICATE in der Sicht SYSCAT.THRESHOLDS.

schwellenwertdomäne

Ein Eingabeargument des Typs VARCHAR(18), das eine Schwellenwertdomäne angibt. Die folgenden Werte sind möglich:

<i>DB</i>	Datenbank
<i>SB</i>	Serviceunterklasse
<i>SP</i>	Servicesuperklasse
<i>WA</i>	Arbeitsaktionsset

Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Schwellenwerte zurückgegeben, die die anderen Kriterien erfüllen.

Die Werte für *schwellenwertdomäne* entsprechen denen der Spalte DOMAIN in der Sicht SYSCAT.THRESHOLDS.

schwellenwertname

Ein Eingabeargument des Typs VARCHAR(128), das einen Schwellenwertnamen angibt. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Schwellenwerte zurückgegeben, die die anderen Kriterien erfüllen. Die Werte für *schwellenwertname* entsprechen denen der Spalte THRESHOLDNAME in der Sicht SYSCAT.THRESHOLDS.

schwellenwert-id

Ein Eingabeargument des Typs INTEGER, das eine Schwellenwert-ID angibt. Wird das Argument weggelassen oder wird -1 angegeben, werden Daten für alle Schwellenwerte zurückgegeben, die die anderen Kriterien erfüllen. Die Werte für *schwellenwert-id* entsprechen denen der Spalte THRESHOLDID in der Sicht SYSCAT.THRESHOLDS.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Die folgende Abfrage zeigt die Basisstatistikdaten für alle Warteschlangen in allen Mitgliedern auf einem System an:

```

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

```

Die Beispielausgabe sieht wie folgt aus:

```

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

```

Hinweis zur Verwendung:

Die Funktion fasst Daten nicht warteschlangenübergreifend (für ein Member) oder memberübergreifend (für mindestens eine Warteschlange) zusammen. Sie können Daten jedoch wie im vorherigen Beispiel gezeigt mithilfe von SQL-Abfragen zusammenfassen.

Zurückgegebene Informationen

Tabelle 70. Für WLM_GET_QUEUE_STATS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
THRESHOLD_PREDICATE	VARCHAR(27)	threshold_predicate - Vergleichselement für Schwellenwert (Monitorelement)
THRESHOLD_DOMAIN	VARCHAR(18)	threshold_domain - Schwellenwertdomäne (Monitorelement)
THRESHOLD_NAME	VARCHAR(128)	threshold_name - Name des Schwellenwerts (Monitorelement)
THRESHOLD_ID	INTEGER	thresholdid - Schwellenwert-ID (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Name der Servicesuperklasse (Monitorelement)
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Name der Serviceunterklasse (Monitorelement)
WORK_ACTION_SET_NAME	VARCHAR(128)	work_action_set_name - Name des Arbeitsaktionssets (Monitorelement)
WORK_CLASS_NAME	VARCHAR(128)	work_class_name - Name der Arbeitsklasse (Monitorelement)
WORKLOAD_NAME	VARCHAR(128)	workload_name - Name der Workload (Monitorelement)

Tabelle 70. Für WLM_GET_QUEUE_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
LAST_RESET	TIMESTAMP	last_reset - Zeitmarke für letzte Zurücksetzung (Monitorelement)
QUEUE_SIZE_TOP	INTEGER	queue_size_top - Maximaler Wert für Warteschlangenlänge (Monitorelement)
QUEUE_TIME_TOTAL	BIGINT	queue_time_total - Gesamtwarteschlangenzeit (Monitorelement)
QUEUE_ASSIGNMENTS_TOTAL	BIGINT	queue_assignments_total - Gesamtzahl der Warteschlangenzuordnungen (Monitorelement)
QUEUE_SIZE_CURRENT	INTEGER	Anzahl der Verbindungen oder Aktivitäten in der Warteschlange.
QUEUE_TIME_LATEST	BIGINT	Zeit, die die letzte Verbindung oder Aktivität, die die Warteschlange verlassen hat, in der Warteschlange verbracht hat. Die Zeit wird in Millisekunden gemessen.
QUEUE_EXIT_TIME_LATEST	TIMESTAMP	Zeit, zu der die letzte Verbindung oder Aktivität die Warteschlange verließ.
THRESHOLD_CURRENT_CONCURRENCY	INTEGER	Anzahl der Verbindungen bzw. Aktivitäten, die derzeit entsprechend dem Schwellenwert ausgeführt werden.
THRESHOLD_MAX_CONCURRENCY	INTEGER	Maximale Anzahl der Verbindungen bzw. Aktivitäten, die gemäß dem Schwellenwert gleichzeitig ausgeführt werden können.
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse aktive Agenten auflisten

Die Funktion WLM_GET_SERVICE_CLASS_AGENTS gibt eine Liste der Agenten, der im abgeschirmten Modus ausgeführten Prozesse (db2fmp-Prozesse) und der Systementitäten in einem angegebenen Member zurück, die in einer angegebenen Serviceklasse oder für eine angegebene Anwendung ausgeführt werden. Die Systementitäten sind Threads und Prozesse, die keine Agenten sind, beispielsweise Seitenlöschfunktionen und Vorablesefunktionen.

Eine vollständige Liste der Informationen, die zurückgegeben werden können, finden Sie in Tabelle 71 auf Seite 410.

Syntax

```

▶▶—WLM_GET_SERVICE_CLASS_AGENTS—(—servicesuperklassenname—, —————→
▶—serviceunterklassenname—, —anwendungskennung—, —member—)————→▶▶

```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

servicesuperklassenname

Ein Eingabeargument des Typs VARCHAR(128), das den Namen einer Service-superklasse in der Datenbank angibt, zu der zurzeit eine Verbindung besteht. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Superklassen in der Datenbank abgerufen.

serviceunterklassenname

Ein Eingabeargument des Typs VARCHAR(128), das auf eine bestimmte Unterklasse in einer Superklasse verweist. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Unterklassen in der Datenbank abgerufen.

anwendungskennung

Ein Eingabeargument des Typs BIGINT, das die Anwendungskennung angibt, für die Agenteninformationen zurückgegeben werden sollen. Wird das Argument weggelassen, werden Daten für alle Anwendungen in der Datenbank abgerufen. Ist die Anwendungskennung 0, werden nur Systementitäten zurückgegeben.

member

Ein Eingabeargument des Typs INTEGER, das die Membroffsetnummer in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Datenbankmember oder -2 für alle Datenbankmember an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel 1

Die folgende Abfrage gibt eine Liste der Agenten zurück, die für alle Datenbankmember der Anwendungskennung 1 zugeordnet sind. Zur Ermittlung der Anwendungskennung können Sie den Befehl **LIST APPLICATIONS** oder die Tabellenfunktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES verwenden.

```
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(CHAR(AGENT_STATE),1,10) AS AGENTSTATE,  
       SUBSTR(CHAR(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
```

Die Beispielausgabe sieht wie folgt aus:

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

Die Ausgabe zeigt einen Koordinatoragenten und einen Subagenten in Member 0 sowie einen Subagenten in Member 1, die für eine Aktivität mit der UOW-ID 1 und der Aktivitäts-ID 5 aktiv sind. In derselben Zeile, in der die Spalte AGENTTYPE den Wert COORDINATOR hat, hat die Spalte REQTYPE (welche den ursprünglichen Anforderungstyp oder den Hauptanforderungstyp angibt), den Wert FETCH. Das bedeutet, dass der Anforderungstyp für den Koordinatoragenten eine Abrufanforderung ist.

Beispiel 2

Die folgende Abfrage ermittelt, auf welche Sperre ein Agent wartet:

```
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
```

Die Beispielausgabe sieht wie folgt aus:

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	02000500000000000000000000054
ROUTINE	PROCESS	EXECUTING	-
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	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-

21 Satz/Sätze ausgewählt.

Eine spätere Verwendung derselben Abfrage zeigt, dass der WLM-Schwellenwert einen Agenten in eine Warteschlange gestellt hat:

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	PROCESS	EXECUTING	-
REQUEST	PROCESS	EXECUTING	-

WLM_QUEUE	WAIT	IDLE	MYCONCDBCOORDTH
ROUTINE	PROCESS	EXECUTING	-
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	PROCESS	EXECUTING	-

21 Satz/Sätze ausgewählt.

Hinweis zur Verwendung:

Die Parameter werden über AND verknüpft. Wenn Sie also Datensätze angeben, die miteinander in Konflikt stehen, beispielsweise einen Servicesuperklassennamen SUP_A und einen Unterklassennamen SUB_B, wobei SUB_B keine Unterklasse von SUP_A ist, werden keine Zeilen zurückgegeben.

Zurückgegebene Informationen

Table 71. Durch WLM_GET_SERVICE_CLASS_AGENTS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Name der Servicesuperklasse (Monitorelement)
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Name der Serviceunterklasse (Monitorelement)
APPLICATION_HANDLE	BIGINT	application_handle - Anwendungskennung (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
ENTITY	VARCHAR (32)	Einer der folgenden Werte: <ul style="list-style-type: none"> Ist der Typ der Entität ein Agent, ist der Wert 'db2agent'. Ist der Typ der Entität ein im abgeschirmten Modus ausgeführter Prozess, ist der Wert 'db2fmp (prozess-id)', wobei <i>prozess-id</i> die Prozess-ID des im abgeschirmten Modus ausgeführten Prozesses ist. Ansonsten ist der Wert der Name der Systementität.
WORKLOAD_NAME	VARCHAR(128)	workload_name - Name der Workload (Monitorelement)
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ID des Workloadvorkommens (Monitorelement)
UOW_ID	INTEGER	uow_id - UOW-ID (Monitorelement)
ACTIVITY_ID	INTEGER	activity_id - Aktivitäts-ID (Monitorelement)
PARENT_UOW_ID	INTEGER	parent_uow_id - UOW-ID der übergeordneten Aktivität (Monitorelement)
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - ID der übergeordneten Aktivität (Monitorelement)
AGENT_TID	BIGINT	agent_tid - Agententhread-ID (Monitorelement)

Tabelle 71. Durch WLM_GET_SERVICE_CLASS_AGENTS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
AGENT_TYPE	VARCHAR (32)	Agententyp. Es gibt folgende Agententypen: <ul style="list-style-type: none"> • COORDINATOR • OTHER • PDBSUBAGENT • SMPSUBAGENT Wenn der Wert COORDINATOR ist, kann sich die Agenten-ID in Konzentratoren umgebungen ändern.
SMP_COORDINATOR	INTEGER	Gibt an, ob der Agent ein SMP-Koordinator ist (1 für 'ja' und 2 für 'nein').
AGENT_SUBTYPE	VARCHAR (32)	Subtyp des Agenten. Die folgenden Subtypen sind möglich: <ul style="list-style-type: none"> • DSS • OTHER • RPC • SMP
AGENT_STATE	VARCHAR (32)	Gibt an, ob ein Agent zugeordnet oder aktiv ist. Folgende Werte sind möglich: <ul style="list-style-type: none"> • ASSOCIATED • ACTIVE
EVENT_TYPE	VARCHAR (32)	Typ des zuletzt von diesem Agenten verarbeiteten Ereignisses. Die folgenden Werte sind möglich: <ul style="list-style-type: none"> • ACQUIRE • PROCESS • WAIT Weitere Informationen zu möglichen Werten für diese Spalte finden Sie in Tabelle 72 auf Seite 414.
EVENT_OBJECT	VARCHAR (32)	Objekt des zuletzt von diesem Agenten verarbeiteten Ereignisses. Die folgenden Werte sind möglich: <ul style="list-style-type: none"> • COMPRESSION_DICTIONARY_BUILD • IMPLICIT_REBIND • INDEX_RECREATE • LOCK • LOCK_ESCALATION • QP_QUEUE • REMOTE_REQUEST • REQUEST • ROUTINE • WLM_QUEUE Weitere Informationen zu möglichen Werten für diese Spalte finden Sie in Tabelle 72 auf Seite 414.
EVENT_STATE	VARCHAR (32)	Status des zuletzt von diesem Agenten verarbeiteten Ereignisses. Die folgenden Werte sind möglich: <ul style="list-style-type: none"> • EXECUTING • IDLE Weitere Informationen zu möglichen Werten für diese Spalte finden Sie in Tabelle 72 auf Seite 414.

Tabelle 71. Durch WLM_GET_SERVICE_CLASS_AGENTS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
REQUEST_ID	VARCHAR (64)	Anforderungs-ID. Dieser Wert ist nur in Kombination mit dem Wert für <i>anwendungskennung</i> eindeutig. Mithilfe dieser Kombination können Sie zwischen zeitintensiven Einzelanforderungen und mehreren Anforderungen unterscheiden, um beispielsweise einen Abruf, der viel Zeit in Anspruch nimmt, von mehreren Abrufen zu unterscheiden.
REQUEST_TYPE	VARCHAR (32)	Anforderungstyp. Die folgenden Werte sind möglich: <ul style="list-style-type: none"> • Für Koordinatoragenten: <ul style="list-style-type: none"> – CLOSE – COMMIT – COMPILE – DESCRIBE – EXCSQLSET – EXECIMMD – EXECUTE – FETCH – INTERNAL <i>zahl</i> (Dabei ist <i>zahl</i> der Wert der internen Konstante.) – OPEN – PREPARE – REBIND – REDISTRIBUTE – REORG – ROLLBACK – RUNSTATS • Für Subagenten mit dem AGENT_SUBTYPE-Wert DSS oder SMP: <ul style="list-style-type: none"> – Wenn die Nummer des Teilbereichs ungleich null ist, wird die Teilbereichsnummer im Format SUBSECTION:<i>teilbereichsnummer</i> angegeben; ansonsten wird NULL zurückgegeben.

Tabelle 71. Durch WLM_GET_SERVICE_CLASS_AGENTS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
REQUEST_TYPE (Fortsetzung)	VARCHAR (32)	<ul style="list-style-type: none"> • Für Subagenten mit dem AGENT_SUBTYPE-Wert RPC: <ul style="list-style-type: none"> - ABP - CATALOG - INTERNAL - REORG - RUNSTATS - WLM • Für Subagenten mit dem SUBTYPE-Wert OTHER: <ul style="list-style-type: none"> - ABP - APP_RBSVPT - APP_RELSVPT - BACKUP - CLOSE - EXTERNAL_RBSVPT - EVMON - FORCE - FORCE_ALL - INTERNAL <i>zahl</i> (Dabei ist <i>zahl</i> der Wert der internen Konstante.) - INTERRUPT - NOOP (wenn keine Anforderung vorhanden) - QP - REDISTRIBUTE - STMT_RBSVPT - STOP_USING - UPDATE_DBM_CFG - WLM
NESTING_LEVEL	INTEGER	nesting_level - Verschachtelungsebene (Monitorelement)
INVOCATION_ID	INTEGER	invocation_id - Aufruf-ID (Monitorelement)
ROUTINE_ID	INTEGER	routine_id - Routinen-ID (Monitorelement)
EVENT_OBJECT_NAME	VARCHAR(1024)	Name des Ereignisobjekts. Wenn EVENT_OBJECT den Wert LOCK hat, ist der Wert dieser Spalte der Name der Sperre, auf die der Agent wartet. Wenn EVENT_OBJECT den Wert WLM_QUEUE hat, ist der Wert dieser Spalte der Name des WLM-Schwellenwerts, der den Agenten in die Warteschlange gestellt hat. Ansonsten ist der Wert NULL.
APPLICATION_NAME	VARCHAR(128)	appl_name - Anwendungsname
APPLICATION_ID	VARCHAR(128)	appl_id - Anwendungs-ID
CLIENT_PID	BIGINT	client_pid - Clientprozess-ID
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - Berechtigungs-ID für Sitzung
REQUEST_START_TIME	TIMESTAMP	Zeit, zu der der Agent die Verarbeitung der Anforderung begann, die er zurzeit verarbeitet

Tabelle 71. Durch WLM_GET_SERVICE_CLASS_AGENTS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
AGENT_STATE_LAST_UPDATE_TIME	TIMESTAMP	Der Zeitpunkt der letzten Änderung des Ereignisses, das zurzeit vom Agenten verarbeitet wird. Das zurzeit vom Agenten verarbeitete Ereignis wird durch die Spalten EVENT_TYPE, EVENT_OBJECT und EVENT_STATE angegeben.
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - Ausführbare ID (Monitorelement)
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

Anmerkung: Die möglichen Kombinationen der Spalten EVENT_STATE, EVENT_TYPE, EVENT_OBJECT und EVENT_OBJECT_NAME sind in der folgenden Tabelle aufgelistet.

Tabelle 72. Mögliche Kombinationen der Spaltenwerte EVENT_STATE, EVENT_TYPE, EVENT_OBJECT und EVENT_OBJECT_NAME

Beschreibung des Ereignisses	Wert EVENT_STATE	Wert EVENT_TYPE	Wert EVENT_OBJECT	Wert EVENT_OBJECT_NAME
Sperre anfordern	IDLE	ACQUIRE	LOCK (Sperre)	Sperrenname
Sperre eskalieren	EXECUTING	PROCESS	LOCK_ESCALATION	NULL
Anforderung verarbeiten	EXECUTING	PROCESS	REQUEST	NULL
Auf neue Anforderung warten	IDLE	WAIT	REQUEST	NULL
Auf Verarbeitung einer Anforderung im fernen Member warten	IDLE	WAIT	REMOTE_REQUEST	NULL
Auf WLM-Schwellenwertwarteschlange warten	IDLE	WAIT	WLM_QUEUE	Schwellenwertname
Routine verarbeiten	EXECUTING	PROCESS	ROUTINE	NULL
Index erneut erstellen	EXECUTING	PROCESS	INDEX_RECREATE	NULL
Komprimierungswörterverzeichnis erstellen	EXECUTING	PROCESS	COMP_DICT_BLD	NULL
Impliziten Rebind durchführen	EXECUTING	PROCESS	IMPLICIT_REBIND	NULL

WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES - Workloadvorkommen auflisten

Die Funktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES gibt die Liste aller Workloadvorkommen zurück, die in einer angegebenen Serviceklasse in einem bestimmten Member ausgeführt werden. Ein Workloadvorkommen ist eine bestimmte Datenbankverbindung, deren Attribute mit der Definition einer Workload übereinstimmen und die somit der Workload zugehörig oder zugeordnet ist.

Eine vollständige Liste der Informationen, die zurückgegeben werden können, finden Sie in Tabelle 73 auf Seite 416.

Syntax

```
►—WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES—(—servicesuperklassenname—,——►  
►—serviceunterklassenname—,—member—)——►
```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

servicesuperklassenname

Ein Eingabeargument des Typs VARCHAR(128), das den Namen einer Service-superklasse in der Datenbank angibt, zu der zurzeit eine Verbindung besteht. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Superklassen in der Datenbank abgerufen, für die die übrigen Parameter übereinstimmen.

serviceunterklassenname

Die Zielserviceunterklasse für das Workloadvorkommen. Mit Ausnahme von Aktivitäten, die einer anderen Unterklasse zugeordnet oder erneut zugeordnet werden, wird jede von diesem Workloadvorkommen übergebene Arbeit in dieser Serviceunterklasse unter der Zielserviceunterklasse ausgeführt.

member

Ein Eingabeargument des Typs INTEGER, das die Nummer eines Members in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Datenbankmember oder -2 für alle Datenbankmember an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Wenn ein Administrator anzeigen möchte, welche Workloadvorkommen auf dem System insgesamt aktiv sind, kann er die Funktion WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES aufrufen, indem er einen Nullwert oder eine leere Zeichenfolge für *servicesuperklassenname* und *serviceunterklassenname* und -2 für *member* angibt:

```
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

```

Wenn das System über vier Datenbankmember verfügt und derzeit zwei Workloads ausführt, gibt die vorstehende Abfrage Ergebnisse ähnlich den folgenden zurück:

```

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
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2
... 1          SYSDEFAULTUSERWORKLOAD 1
... 2          SYSDEFAULTUSERWORKLOAD 2

```

Hinweis zur Verwendung:

Die Parameter werden über AND verknüpft. Wenn Sie also Datensätze angeben, die miteinander in Konflikt stehen, beispielsweise einen Servicesuperklassennamen SUP_A und einen Unterklassennamen SUB_B, wobei SUB_B keine Unterklasse von SUP_A ist, werden keine Zeilen zurückgegeben.

Anmerkung: Die für das Workloadvorkommen erfassten Statistikdaten (beispielsweise coord_act_completed_total) werden zu Beginn jeder UOW (Unit of Work) zurückgesetzt, wenn sie mit den entsprechenden Workloadstatistikdaten kombiniert werden.

Zurückgegebene Informationen

Tabelle 73. Für WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Name der Servicesuperklasse (Monitorelement)
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Name der Serviceunterklasse (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)

Tabelle 73. Für WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - Koordinatorpartitionsnummer (Monitorelement)
APPLICATION_HANDLE	BIGINT	application_handle - Anwendungskennung (Monitorelement)
WORKLOAD_NAME	VARCHAR(128)	workload_name - Name der Workload (Monitorelement)
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - ID des Workloadvorkommens (Monitorelement)
UOW_ID	INTEGER	uow_id - UOW-ID (Monitorelement)
WORKLOAD_OCCURRENCE_STATE	VARCHAR(32)	workload_occurrence_state - Status des Workloadvorkommens (Monitorelement)
SYSTEM_AUTH_ID	VARCHAR(128)	system_auth_id - Systemberechtigungs-ID (Monitorelement)
SESSION_AUTH_ID	VARCHAR(128)	session_auth_id - Berechtigungs-ID für Sitzung (Monitorelement)
APPLICATION_NAME	VARCHAR(128)	appl_name - Anwendungsname (Monitorelement)
CLIENT_WRKSTNNAME	VARCHAR(255)	client_wrkstnname - Name der Client-Workstation (Monitorelement)
CLIENT_ACCTNG	VARCHAR(255)	client_acctng - Abrechnungszeichenfolge des Clients (Monitorelement)
CLIENT_USER	VARCHAR(255)	Der aktuelle Wert des Sonderregisters CLIENT_USERID für dieses Workloadvorkommen.
CLIENT_APPLNAME	VARCHAR(255)	client_applname - Name der Clientanwendung (Monitorelement)
COORD_ACT_COMPLETED_TOTAL	INTEGER	coord_act_completed_total - Gesamtzahl abgeschlossener Koordinatoraktivitäten (Monitorelement)
COORD_ACT_ABORTED_TOTAL	INTEGER	coord_act_aborted_total - Gesamtzahl abgebrochener Koordinatoraktivitäten (Monitorelement)
COORD_ACT_REJECTED_TOTAL	INTEGER	coord_act_rejected_total - Gesamtzahl zurückgewiesener Koordinatoraktivitäten (Monitorelement)
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - Maximaler Wert für gleichzeitige Aktivitäten (Monitorelement)
ADDRESS	VARCHAR(255)	address - IP-Adresse, von der aus die Verbindung initialisiert wurde
APPL_ID	VARCHAR(128)	appl_id - Anwendungs-ID
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)
COORD_MEMBER	SMALLINT	coord_member - Koordinatormember (Monitorelement)

WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben

Die Funktion WLM_GET_SERVICE_SUBCLASS_STATS gibt Basisstatistikdaten zu mindestens einer Serviceunterklasse zurück.

Eine vollständige Liste der Informationen, die zurückgegeben werden können, finden Sie in Tabelle 74 auf Seite 420.

Syntax

```
► WLM_GET_SERVICE_SUBCLASS_STATS (—servicesuperklassenname—, —serviceunterklassenname—, —member—)
```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

servicesuperklassenname

Ein Eingabeargument des Typs VARCHAR(128), das den Namen einer Service-superklasse in der Datenbank angibt, zu der zurzeit eine Verbindung besteht. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Superklassen in der Datenbank abgerufen.

serviceunterklassenname

Ein Eingabeargument des Typs VARCHAR(128), das den Namen einer Service-unterklasse in der Datenbank angibt, zu der zurzeit eine Verbindung besteht. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Unterklassen in der Datenbank abgerufen.

member

Ein Eingabeargument des Typs INTEGER, das eine gültige Membrnummer in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Member oder -2 für alle Datenbank-member an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiele

Beispiel 1: Da jede Aktivität vor ihrer Ausführung einer DB2-Serviceklasse zugeordnet werden muss, können Sie den globalen Status des Systems überwachen, indem Sie die Tabellenfunktionen für die Serviceklassenstatistik verwenden und alle Ser-

viceklassen in allen Mitgliedern abfragen. Im folgenden Beispiel wird für *servicesuperklassenname* und *serviceunterklassenname* ein Nullwert übergeben, um Statistikdaten für alle Serviceklassen zurückzugeben. Für *member* wird der Wert -2 angegeben, um Statistikdaten für alle Mitglieder zurückzugeben:

```
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
```

Die Anweisung gibt Serviceklassenstatistikdaten wie durchschnittliche Laufzeit einer Aktivität und Standardabweichung in Sekunden zurück:

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

Beispiel 2: Mit derselben Tabellenfunktion kann auch der höchste Wert für den durchschnittlichen gemeinsamen Zugriff von Koordinatoraktivitäten zurückgegeben werden, die in der Serviceklasse in den einzelnen Mitgliedern aktiv sind.

```
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
```

Die Beispielausgabe sieht wie folgt aus:

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

Sie können in der Ausgabe dieser Tabellenfunktion die durchschnittlichen Ausführungszeiten und die Anzahl der Aktivitäten überprüfen und erhalten so für eine bestimmte Datenbank einen guten Überblick über die Arbeitslast in jedem Mitglied. Erhebliche Abweichungen der Wertangaben auf hoher Ebene, die von dieser Tabellenfunktion zurückgegeben werden, geben möglicherweise eine Änderung der Arbeitslast auf dem System an.

Beispiel 3: Wenn eine Aktivität für REMAP ACTIVITY TO-Aktivitäten Schwellenwerte verwendet, verbringt die Aktivität während ihrer Laufzeit möglicherweise in

mehr als einer Serviceklasse Zeit. Sie können die Spalten ACT_REMAPPED_IN und ACT_REMAPPED_OUT überprüfen, um zu ermitteln, wie viele Aktivitäten ein Serviceklassenset passiert haben:

```
SELECT SUBSTR(SERVICE_SUPERCLASS_NAME,1,19) AS SUPERCLASS_NAME,
       SUBSTR(SERVICE_SUBCLASS_NAME,1,18) AS SUBCLASS_NAME,
       ACT_REMAPPED_IN AS MAPPED_IN,
       ACT_REMAPPED_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
```

Die Beispielausgabe sieht wie folgt aus:

SUPERCLASS_NAME	SUBCLASS_NAME	MAPPED_IN	MAPPED_OUT
SYSDEFAULTUSERCLASS	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SYSDEFAULTSUBCLASS	0	0
SUPERCLASS1	SUBCLASS1	0	7
SUPERCLASS1	SUBCLASS2	7	0

Hinweise zur Verwendung

Einige Statistikdaten werden nur zurückgegeben, wenn Sie die Parameter COLLECT AGGREGATE ACTIVITY DATA und COLLECT AGGREGATE REQUEST DATA für die entsprechende Serviceunterklasse auf einen anderen Wert als NONE setzen.

Die Tabellenfunktion WLM_GET_SERVICE_SUBCLASS_STATS gibt eine Datenzeile pro Serviceunterklasse und pro Member zurück. Die Funktion fasst Daten nicht serviceklassenübergreifend (für eine Partition) oder partitionsübergreifend (für mindestens eine Serviceklasse) zusammen. Sie können Daten jedoch mithilfe von SQL-Abfragen zusammenfassen.

Die Parameter werden über AND verknüpft. Wenn Sie also Datensätze angeben, die miteinander in Konflikt stehen, beispielsweise einen Superklassennamen SUPA und einen Unterklassennamen SUBB, wobei SUBB keine Unterklasse von SUPA ist, werden keine Zeilen zurückgegeben.

Zurückgegebene Informationen

Tabelle 74. Für WLM_GET_SERVICE_SUBCLASS_STATS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Name der Servicesuperklasse (Monitorelement)
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - Name der Serviceunterklasse (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
LAST_RESET	TIMESTAMP	last_reset - Zeitmarke für letzte Zurücksetzung (Monitorelement)
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - Gesamtzahl abgeschlossener Koordinatoraktivitäten (Monitorelement)
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - Gesamtzahl abgebrochener Koordinatoraktivitäten (Monitorelement)

Tabelle 74. Für WLM_GET_SERVICE_SUBCLASS_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - Gesamtzahl zurückgewiesener Koordinatoraktivitäten (Monitorelement)
CONCURRENT_ACT_TOP	INTEGER	concurrent_act_top - Maximaler Wert für gleichzeitige Aktivitäten (Monitorelement)
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - Maximaler Wert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - Durchschnittswert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)
COORD_ACT_LIFETIME_STDDEV	DOUBLE	<p>Die Standardabweichung der Laufzeit von Koordinatoraktivitäten auf der Verschachtelungsebene 0, die der Serviceunterklasse seit dem letzten Zurücksetzen zugeordnet wurden. Wenn der Parameter COLLECT AGGREGATE ACTIVITY DATA der Serviceklasse auf NONE gesetzt ist, ist der Wert der Spalte Null. Die Zeit wird in Millisekunden gemessen.</p> <p>Diese Standardabweichung wird anhand des Histogramms für die Laufzeit der Koordinatoraktivität berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird -1 zurückgegeben.</p> <p>Aktivitäten, die die Serviceunterklasse passieren, aber vor Abschluss ihrer Ausführung einer anderen Unterklasse zugeordnet werden, beeinflussen den Wert COORD_ACT_LIFETIME_STDDEV einer Serviceunterklasse nicht.</p>
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - Durchschnittswert für die Ausführungszeit von Koordinatoraktivitäten (Monitorelement)

Tabella 74. Für WLM_GET_SERVICE_SUBCLASS_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	<p>Die Standardabweichung der Ausführungszeiten von Koordinatoraktivitäten auf der Verschachtelungsebene 0, die der Serviceunterklasse seit dem letzten Zurücksetzen zugeordnet wurden. Die Zeit wird in Millisekunden gemessen.</p> <p>Diese Standardabweichung wird anhand des Histogramms für die Ausführungszeit der Koordinatoraktivität berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird -1 zurückgegeben.</p> <p>Aktivitäten, die die Unterklasse passieren, aber vor Abschluss ihrer Ausführung einer anderen Unterklasse zugeordnet werden, beeinflussen die Standardabweichung der Ausführungszeit einer Serviceunterklasse nicht.</p>
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	<p>coord_act_queue_time_avg - Durchschnittliche Warteschlangenzeit für Koordinatoraktivitäten (Monitorelement)</p>
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	<p>Die Standardabweichung der Wartezeit von Koordinatoraktivitäten auf der Verschachtelungsebene 0, die der Serviceunterklasse seit dem letzten Zurücksetzen zugeordnet wurden. Wenn der Parameter COLLECT AGGREGATE ACTIVITY DATA der Serviceklasse auf NONE gesetzt ist, ist der Wert der Spalte Null. Die Zeit wird in Millisekunden gemessen.</p> <p>Diese Standardabweichung wird anhand des Histogramms für die Wartezeit der Koordinatoraktivität berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird -1 zurückgegeben.</p> <p>Die Standardabweichung der Wartezeit wird nur für die Serviceunterklasse gezählt, in der die Aktivität in die Warteschlange gestellt wurde.</p>
NUM_REQUESTS_ACTIVE	BIGINT	<p>Anzahl der Anforderungen, die während der Ausführung dieser Tabellenfunktion in der Serviceunterklasse aktiv sind.</p>

Table 74. For WLM_GET_SERVICE_SUBCLASS_STATS returned information (Forts.)

Spaltenname	Datentyp	Beschreibung
NUM_REQUESTS_TOTAL	BIGINT	<p>Die Anzahl der Anforderungen, deren Ausführung in dieser Serviceunterklasse seit dem letzten Zurücksetzen abgeschlossen wurde. Dies gilt für alle Anforderungen unabhängig von ihrer Zugehörigkeit zu einer Aktivität. Wenn der Parameter COLLECT AGGREGATE ACTIVITY DATA der Serviceklasse auf NONE gesetzt ist, ist der Wert der Spalte Null.</p> <p>Anforderungen, die die Serviceunterklasse passieren, deren Ausführung aber nicht in ihr abgeschlossen wird, beeinflussen den Wert NUM_REQUESTS_TOTAL einer Serviceunterklasse nicht.</p>
REQUEST_EXEC_TIME_AVG	DOUBLE	<p>request_exec_time_avg - Durchschnittswert für die Ausführungszeit von Anforderungen (Monitorelement)</p>
REQUEST_EXEC_TIME_STDDEV	DOUBLE	<p>Die Standardabweichung der Ausführungszeiten von Anforderungen, die dieser Serviceunterklasse seit dem letzten Zurücksetzen zugeordnet wurden. Die Zeit wird in Millisekunden gemessen. Wenn der Parameter COLLECT AGGREGATE REQUEST DATA der Serviceklasse auf NONE gesetzt ist, ist der Wert dieser Spalte NULL.</p> <p>Diese Standardabweichung wird anhand des Histogramms für die Ausführungszeit der Anforderung berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird -1 zurückgegeben.</p> <p>Anforderungen, die die Unterklasse passieren, deren Ausführung aber nicht in ihr abgeschlossen wurde, beeinflussen die Standardabweichung der Ausführungszeit von Anforderungen einer Serviceunterklasse nicht.</p>

Tabelle 74. Für WLM_GET_SERVICE_SUBCLASS_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
REQUEST_EXEC_TIME_TOTAL	BIGINT	<p>Die Summe der Ausführungszeiten von Anforderungen, die dieser Serviceunterklasse seit dem letzten Zurücksetzen zugeordnet wurden. Die Zeit wird in Millisekunden gemessen. Wenn der Parameter COLLECT AGGREGATE REQUEST DATA der Serviceklasse auf NONE gesetzt ist, ist der Wert dieser Spalte NULL.</p> <p>Diese Summe wird anhand des Histogramms für die Ausführungszeit der Anforderung berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird -1 zurückgegeben.</p> <p>Anforderungen, die die Unterklasse passieren, deren Ausführung aber nicht in ihr abgeschlossen wird, beeinflussen die Summe der Ausführungszeiten von Anforderungen einer Serviceunterklasse nicht.</p>
ACT_REMAPPED_IN	BIGINT	Die Anzahl der Aktivitäten, die seit dem letzten Zurücksetzen in dieser Serviceunterklasse von einer Schwellenwertaktion REMAP ACTIVITY erneut zugeordnet wurden.
ACT_REMAPPED_OUT	BIGINT	Die Anzahl der Aktivitäten, die seit dem letzten Zurücksetzen aus dieser Serviceunterklasse von einer Schwellenwertaktion REMAP ACTIVITY erneut zugeordnet wurden.
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - Maximaler Wert für gleichzeitige Workloadvorkommen (Monitorelement)
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - Maximaler Wert für UOW-Gesamtzeit (Monitorelement)
UOW_THROUGHPUT	DOUBLE	<p>uow_throughput - UOW-Durchsatz (Monitorelement)</p> <p>Der UOW-Durchsatz seit dem letzten Zurücksetzen der Statistikdaten.</p>
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - Durchschnittliche UOW-Lebensdauer (Monitorelement)
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - Gesamtzahl beendeter UOWs (Monitorelement)
TOTAL_CPU_TIME	BIGINT	total_cpu_time - Gesamt-CPU-Zeit (Monitorelement)
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - Gesamtwartezeit in Ausführungswarteschlange des Dispatchers (Monitorelement)

Tabelle 74. Für WLM_GET_SERVICE_SUBCLASS_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
ACT_THROUGHPUT	DOUBLE	act_throughput - Aktivitätsdurchsatz (Monitorelement)
CPU_UTILIZATION	DOUBLE	cpu_utilization - CPU-Auslastung (Monitorelement)
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - Gesamtzahl erfolgreicher externer Koordinatoraktivitäten (Monitorelement)
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - Gesamtzahl fehlgeschlagener externer Koordinatoraktivitäten (Monitorelement)
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - Gesamtzahl zurückgewiesener externer Koordinatoraktivitäten (Monitorelement)
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

WLM_GET_SERVICE_SUPERCLASS_STATS - Statistikdaten zu Servicesuperklassen zurückgeben

Die Funktion WLM_GET_SERVICE_SUPERCLASS_STATS gibt Basisstatistikdaten für mindestens eine Servicesuperklasse zurück.

Syntax

```
►►—WLM_GET_SERVICE_SUPERCLASS_STATS—(—servicesuperklassenname—,—member—)——►►
```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

servicesuperklassenname

Ein Eingabeargument des Typs VARCHAR(128), das den Namen einer Servicesuperklasse in der Datenbank angibt, zu der zurzeit eine Verbindung besteht. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Daten für alle Superklassen in der Datenbank abgerufen.

member

Ein Eingabeargument des Typs INTEGER, das eine gültige Memberrnummer in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Datenbankmember oder -2 für alle Datenbankmember an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Die folgende Abfrage zeigt die Basisstatistikdaten für alle Servicesuperklassen in allen Datenbankmembern auf einem System an:

```
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
```

Die Beispielausgabe sieht wie folgt aus:

```
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
```

Hinweis zur Verwendung:

Die Tabellenfunktion WLM_GET_SERVICE_SUPERCLASS_STATS gibt eine Datenzeile pro Servicesuperklasse und pro Member zurück. Die Funktion fasst Daten nicht servicesuperklassenübergreifend (für ein Member) oder memberübergreifend (für mindestens eine Servicesuperklasse) zusammen. Sie können Daten jedoch wie im vorherigen Beispiel gezeigt mithilfe von SQL-Abfragen zusammenfassen.

Zurückgegebene Informationen

Tabelle 75. Für WLM_GET_SERVICE_SUPERCLASS_STATS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - Name der Servicesuperklasse (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
LAST_RESET	TIMESTAMP	last_reset - Zeitmarke für letzte Zurücksetzung (Monitorelement)
CONCURRENT_CONNECTION_TOP	INTEGER	concurrent_connection_top - Maximaler Wert für gleichzeitige Verbindungen (Monitorelement)
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

WLM_GET_WORK_ACTION_SET_STATS - Statistikdaten zu Arbeitsaktionssets zurückgeben

Die Funktion WLM_GET_WORK_ACTION_SET_STATS gibt die Statistikdaten zu einem Arbeitsaktionsset zurück.

Syntax

```
►►—WLM_GET_WORK_ACTION_SET_STATS—(—arbeitsaktionssetname—,—member—)————►►
```

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

arbeitsaktionssetname

Ein Eingabeargument des Typs VARCHAR(128), welches das Arbeitsaktionsset angibt, zu dem Statistikdaten zurückgegeben werden sollen. Wird das Argument weggelassen oder wird eine leere Zeichenfolge angegeben, werden Statistikdaten zu allen Arbeitsaktionssets zurückgegeben.

member

Ein Eingabeargument des Typs INTEGER, das eine gültige Memberrnummer in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Datenbankmember oder -2 für alle Datenbankmember an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Angenommen, es gibt drei Arbeitsklassen, ReadClass, WriteClass und LoadClass. Es gibt eine Arbeitsaktion, die ReadClass zugeordnet ist, sowie eine Arbeitsaktion, die LoadClass zugeordnet ist, jedoch keine Arbeitsaktion, die WriteClass zugeordnet ist. In Member 0 gibt es die folgende Anzahl Aktivitäten, die derzeit ausgeführt werden oder in der Warteschlange stehen:

- Klasse ReadClass: acht
- Klasse WriteClass: vier
- Klasse LoadClass: zwei
- Nicht zugeordnet: drei

```
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
```

Die Beispielausgabe sieht wie folgt aus. Da der Arbeitsklasse WriteClass keine Arbeitsaktion zugeordnet ist, werden die vier Aktivitäten, auf die sie angewendet

wird, zahlenmäßig in der künstlichen Klasse erfasst, die in der Ausgabe durch einen Stern (*) gekennzeichnet ist. Die drei Aktivitäten, die keiner Arbeitsklasse zugeordnet wurden, werden ebenfalls in der künstlichen Klasse erfasst.

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

Zurückgegebene Informationen

Tabelle 76. Für WLM_GET_WORK_ACTION_SET_STATS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
WORK_ACTION_SET_NAME	VARCHAR(128)	work_action_set_name - Name des Arbeitsaktionssets (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
LAST_RESET	TIMESTAMP	last_reset - Zeitmarke für letzte Zurücksetzung (Monitorelement)
WORK_CLASS_NAME	VARCHAR(128)	work_class_name - Name der Arbeitsklasse (Monitorelement)
ACT_TOTAL	BIGINT	act_total - Gesamtzahl der Aktivitäten (Monitorelement)
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES - Liste von Aktivitäten zurückgeben

Die Funktion WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES gibt die Liste aller Aktivitäten zurück, die von der angegebenen Anwendung im angegebenen Member übergeben und noch nicht beendet wurden.

Eine vollständige Liste der Informationen, die zurückgegeben werden können, finden Sie in Tabelle 77 auf Seite 430.

Syntax

►►—WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES—(—anwendungskennung—,—member—)►►

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

anwendungskennung

Ein Eingabeargument des Typs BIGINT, das eine Anwendungskennung angibt, für die eine Liste von Aktivitäten zurückgegeben werden soll. Wird das Argument weggelassen, werden Daten für alle Anwendungen in der Datenbank abgerufen.

member

Ein Eingabeargument des Typs INTEGER, das eine gültige Membernummer in

derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Member oder -2 für alle Member an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiele

Aktivitäten, die zurzeit mit einer bekannten Anwendungskennung ausgeführt werden

Nachdem Sie die Anwendungskennung ermittelt haben, können Sie alle Aktivitäten suchen, die zurzeit in dieser Anwendung ausgeführt werden. Angenommen, ein Administrator möchte die Aktivitäten einer Anwendung auflisten, deren Anwendungskennung mit dem Befehl **LIST APPLICATIONS** ermittelt wurde und 1 lautet. Der Administrator führt dazu die folgende Abfrage aus:

```
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
```

Die Beispielausgabe der Abfrage sieht wie folgt aus:

COORD	MEMB	UOWID	ACTID	PARUOWID	PARACTID	ACTTYPE	NESTING
0	0	2	3	-	-	CALL	0
0	0	2	5	2	3	READ_DML	1

Aktivitäten, die zurzeit auf dem System ausgeführt werden

Die folgende Abfrage verknüpft die Ausgabe von WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES mit der Ausgabe von MON_GET_PKG_CACHE_STMT für EXECUTABLE_ID, um einen Anweisungstext für alle Aktivitäten zu erhalten, die zurzeit auf dem System ausgeführt werden:

```
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
```

Die Beispielausgabe sieht wie folgt aus:

APPLICATION_HANDLE	UOW_ID	ACTIVITY_ID	STMT_TEXT
1	1	1	SELECT * FROM SYSCAT.TABLES
47	1	36	INSERT INTO T1 VALUES(123)

Zurückgegebene Informationen

Tabelle 77. Durch WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
APPLICATION_HANDLE	BIGINT	application_handle - Anwendungskennung (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
COORD_PARTITION_NUM	SMALLINT	coord_partition_num - Koordinatorpartitionsnummer (Monitorelement)
LOCAL_START_TIME	TIMESTAMP	local_start_time - Lokale Startzeit (Monitorelement)
UOW_ID	INTEGER	uow_id - UOW-ID (Monitorelement)
ACTIVITY_ID	INTEGER	activity_id - Aktivitäts-ID (Monitorelement)
PARENT_UOW_ID	INTEGER	parent_uow_id - UOW-ID der übergeordneten Aktivität (Monitorelement)
PARENT_ACTIVITY_ID	INTEGER	parent_activity_id - ID der übergeordneten Aktivität (Monitorelement)
ACTIVITY_STATE	VARCHAR(32)	activity_state - Aktivitätsstatus (Monitorelement)

Tabelle 77. Durch WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
ACTIVITY_STATE (Fortsetzung)	VARCHAR(32)	<p>Aktivitätszustand. Folgende Werte sind gültig:</p> <p><i>QUEUED</i> Die Aktivität wurde durch einen Warteschlangenschwellenwert des Workload-Managements in eine Warteschlange gestellt. In einer Umgebung mit partitionierten Datenbanken kann dieser Status darauf hindeuten, dass der Koordinatoragent einen Prozedurfernaufruf an das Katalogmember abgesetzt hat, um Schwellenwerttickets zu erhalten, und noch keine Antwort erhalten hat. Wird dieser Status angezeigt, wurde die Aktivität möglicherweise durch einen Warteschlangenschwellenwert des Workload-Managements in eine Warteschlange gestellt. Wird der Status nur kurzzeitig angezeigt, kann dies darauf hindeuten, dass die Aktivität gerade ihre Tickets abrufen. Wenn Sie sich ein genaueres Bild darüber machen wollen, ob die Aktivität tatsächlich in die Warteschlange gestellt wird, ermitteln Sie den Agenten, der die Aktivität verarbeitet, und stellen Sie fest, ob der Wert EVENT_OBJECT des Objekts im Katalogmember den Wert WLM_QUEUE hat.</p> <p><i>TERMINATING</i> Die Ausführung der Aktivität wurde beendet und die Aktivität wird aus dem System entfernt.</p>
ACTIVITY_TYPE	VARCHAR(32)	<p>Aktivitätstyp. Folgende Werte sind gültig:</p> <ul style="list-style-type: none"> • CALL • DDL • LOAD • OTHER • READ_DML • WRITE_DML
NESTING_LEVEL	INTEGER	nesting_level - Verschachtelungsebene (Monitorelement)
INVOCATION_ID	INTEGER	invocation_id - Aufruf-ID (Monitorelement)
ROUTINE_ID	INTEGER	routine_id - Routinen-ID (Monitorelement)
UTILITY_ID	INTEGER	utility_id - Dienstprogramm-ID (Monitorelement)
SERVICE_CLASS_ID	INTEGER	service_class_id - Serviceklassen-ID (Monitorelement)

Tabelle 77. Durch WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
DATABASE_WORK_ACTION_SET_ID	INTEGER	Einer der folgenden Werte: <ul style="list-style-type: none"> • Wenn diese Aktivität innerhalb einer Arbeitsklasse im Datenbankbereich kategorisiert wurde, ist der Wert die ID des Arbeitsklassensets, zu der diese Arbeitsklasse gehört. • Wenn die Aktivität nicht innerhalb einer Arbeitsklasse im Datenbankbereich kategorisiert wurde, ist der Wert null.
DATABASE_WORK_CLASS_ID	INTEGER	Einer der folgenden Werte: <ul style="list-style-type: none"> • Wenn die Aktivität innerhalb einer Arbeitsklasse im Datenbankbereich kategorisiert wurde, ist der Wert die ID der Arbeitsklasse. • Wenn die Aktivität nicht innerhalb einer Arbeitsklasse im Datenbankbereich kategorisiert wurde, ist der Wert null.
SERVICE_CLASS_WORK_ACTION_SET_ID	INTEGER	Einer der folgenden Werte: <ul style="list-style-type: none"> • Wenn die Aktivität innerhalb einer Arbeitsklasse im Serviceklassenbereich kategorisiert wurde, ist der Wert die ID des Arbeitsaktionssets, die dem Arbeitsklassenset, zu dem die Arbeitsklasse gehört, zugeordnet ist. • Wenn die Aktivität nicht innerhalb einer Arbeitsklasse im Serviceklassenbereich kategorisiert wurde, ist der Wert null.
SERVICE_CLASS_WORK_CLASS_ID	INTEGER	Einer der folgenden Werte: <ul style="list-style-type: none"> • Wurde die Aktivität innerhalb einer Arbeitsklasse im Serviceklassenbereich kategorisiert, ist der Wert die ID der Arbeitsklasse, die dieser Aktivität zugeordnet ist. • Wenn die Aktivität nicht innerhalb einer Arbeitsklasse im Serviceklassenbereich kategorisiert wurde, ist der Wert null.
EXECUTABLE_ID	VARCHAR(32) FOR BIT DATA	executable_id - Ausführbare ID (Monitorelement)
TOTAL_CPU_TIME	BIGINT	total_cpu_time - Gesamt-CPU-Zeit
ROWS_READ	BIGINT	rows_read - Gelesene Zeilen
ROWS_RETURNED	BIGINT	rows_returned - Zurückgegebene Zeilen
QUERY_COST_ESTIMATE	BIGINT	query_cost_estimate - Geschätzter Abfrageaufwand
DIRECT_READS	BIGINT	direct_reads - Direkte Lesevorgänge für Datenbank
DIRECT_WRITES	BIGINT	direct_writes - Direkte Schreibvorgänge für Datenbank
ENTRY_TIME	TIMESTAMP	entry_time - Eintrittszeit
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

Tabelle 77. Durch WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
COORD_MEMBER	SMALLINT	coord_member - Koordinatormember (Monitorelement)
PACKAGE_SCHEMA	VARCHAR(128)	package_schema - Paketschema.
PACKAGE_NAME	VARCHAR(128)	package_name - Paketname.
PACKAGE_VERSION_ID	VARCHAR (64)	package_version_id - Paketversion.
SECTION_NUMBER	BIGINT	section_number - Abschnittsnummer.
STMTNO	INTEGER	stmtno - Anweisungsnummer (Monitorelement)

WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Workloadstatistikdaten zurückgeben

Die Funktion WLM_GET_WORKLOAD_STATS gibt für jede Kombination aus Workloadname und Datenbankmemberrnummer eine Zeile mit Workloadstatistikdaten zurück.

Eine vollständige Liste der Informationen, die zurückgegeben werden können, finden Sie in Tabelle 78 auf Seite 434.

Syntax

►► WLM_GET_WORKLOAD_STATS (—workloadname—, —member—) ◀◀

Das Schema ist SYSPROC.

Tabellenfunktionsparameter

workloadname

Ein Eingabeargument des Typs VARCHAR(128), das eine Workload angibt, zu der die Statistikdaten zurückgegeben werden sollen. Wird für das Argument NULL oder eine leere Zeichenfolge angegeben, werden Statistikdaten zu allen Workloads zurückgegeben.

member

Ein Eingabeargument des Typs INTEGER, das die Nummer eines Members in derselben Instanz wie die Datenbank angibt, zu der zurzeit eine Verbindung besteht. Geben Sie -1 für das aktuelle Member oder -2 für alle Member an. Bei Angabe eines Nullwerts wird -1 implizit gesetzt.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiel

Die folgende Abfrage zeigt Statistikdaten zu Workloads an:

```
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
```

Die Beispielausgabe der Abfrage sieht wie folgt aus:

```
WL_DEF_NAME      MEMB COORD_ACT_LIFETIME_TOP ...  
-----  
SYSDEFAULTADMWORKL 0                -1 ...  
SYSDEFAULTUSERWORK 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
```

Hinweis zur Verwendung:

Die Funktion fasst Daten nicht workload-, member- oder serviceklassenübergreifend zusammen. Sie können Daten jedoch mithilfe von SQL-Abfragen zusammenfassen.

Zurückgegebene Informationen

Tabelle 78. Durch WLM_GET_WORKLOAD_STATS zurückgegebene Informationen

Spaltenname	Datentyp	Beschreibung
WORKLOAD_NAME	VARCHAR(128)	workload_name - Name der Workload (Monitorelement)
DBPARTITIONNUM	SMALLINT	dbpartitionnum - Datenbankpartitionsnummer (Monitorelement)
LAST_RESET	TIMESTAMP	last_reset - Zeitmarke für letzte Zurücksetzung (Monitorelement)
CONCURRENT_WLO_TOP	INTEGER	concurrent_wlo_top - Maximaler Wert für gleichzeitige Workloadvorkommen (Monitorelement)
CONCURRENT_WLO_ACT_TOP	INTEGER	concurrent_wlo_act_top - Maximaler Wert für gleichzeitige Workloadaktivitäten (Monitorelement)
COORD_ACT_COMPLETED_TOTAL	BIGINT	coord_act_completed_total - Gesamtzahl abgeschlossener Koordinatoraktivitäten (Monitorelement)
COORD_ACT_ABORTED_TOTAL	BIGINT	coord_act_aborted_total - Gesamtzahl abgebrochener Koordinatoraktivitäten (Monitorelement)
COORD_ACT_REJECTED_TOTAL	BIGINT	coord_act_rejected_total - Gesamtzahl zurückgewiesener Koordinatoraktivitäten (Monitorelement)

Tabelle 78. Durch WLM_GET_WORKLOAD_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
WLO_COMPLETED_TOTAL	BIGINT	wlo_completed_total - Gesamtzahl abgeschlossener Workloadvorkommen (Monitorelement)
COORD_ACT_LIFETIME_TOP	BIGINT	coord_act_lifetime_top - Maximaler Wert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)
COORD_ACT_LIFETIME_AVG	DOUBLE	coord_act_lifetime_avg - Durchschnittswert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)
COORD_ACT_LIFETIME_STDDEV	DOUBLE	Die Standardabweichung der Laufzeit von abgeschlossenen oder abgebrochenen Koordinatoraktivitäten auf der Verschachtelungsebene 0, die dieser Workload zugeordnet sind. Die Zeit wird in Millisekunden gemessen. Wenn der Parameter COLLECT AGGREGATE ACTIVITY DATA der Workload auf NONE gesetzt ist, ist der Wert der Spalte Null. Diese Standardabweichung wird anhand des Histogramms für die Laufzeit der Koordinatoraktivität berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird der Wert -1 zurückgegeben.
COORD_ACT_EXEC_TIME_AVG	DOUBLE	coord_act_exec_time_avg - Durchschnittswert für die Ausführungszeit von Koordinatoraktivitäten (Monitorelement)
COORD_ACT_EXEC_TIME_STDDEV	DOUBLE	Die Standardabweichung der Ausführungszeiten von abgeschlossenen oder abgebrochenen Koordinatoraktivitäten auf der Verschachtelungsebene 0, die dieser Workload zugeordnet sind. Die Zeit wird in Millisekunden gemessen. Diese Standardabweichung wird anhand des Histogramms für die Ausführungszeit der Koordinatoraktivität berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird der Wert -1 zurückgegeben. Wenn der Parameter COLLECT AGGREGATE ACTIVITY DATA der Workload auf NONE gesetzt ist, ist der Wert der Spalte Null.
COORD_ACT_QUEUE_TIME_AVG	DOUBLE	coord_act_queue_time_avg - Durchschnittliche Warteschlangenzeit für Koordinatoraktivitäten (Monitorelement)

Tabelle 78. Durch WLM_GET_WORKLOAD_STATS zurückgegebene Informationen (Forts.)

Spaltenname	Datentyp	Beschreibung
COORD_ACT_QUEUE_TIME_STDDEV	DOUBLE	Die Standardabweichung der Wartezeit von abgeschlossenen oder abgebrochenen Koordinatoraktivitäten auf der Verschachtelungsebene 0, die dieser Workload zugeordnet sind. Die Zeit wird in Millisekunden gemessen. Wenn der Parameter COLLECT AGGREGATE ACTIVITY DATA der Workload auf NONE gesetzt ist, ist der Wert der Spalte Null. Diese Standardabweichung wird anhand des Histogramms für die Wartezeit der Koordinatoraktivität berechnet und ist möglicherweise ungenau, wenn die für die Daten erforderliche Größe des Histogramms nicht korrekt angegeben wurde. Ist einer der Werte im letzten Histogrammbin enthalten, wird der Wert -1 zurückgegeben.
UOW_TOTAL_TIME_TOP	BIGINT	uow_total_time_top - Maximaler Wert für UOW-Gesamtzeit (Monitorelement)
UOW_THROUGHPUT	DOUBLE	uow_throughput - UOW-Durchsatz (Monitorelement)
UOW_LIFETIME_AVG	DOUBLE	uow_lifetime_avg - Durchschnittliche UOW-Lebensdauer (Monitorelement)
UOW_COMPLETED_TOTAL	BIGINT	uow_completed_total - Gesamtzahl beendeter UOWs (Monitorelement)
TOTAL_CPU_TIME	BIGINT	total_cpu_time - Gesamt-CPU-Zeit (Monitorelement)
TOTAL_DISP_RUN_QUEUE_TIME	BIGINT	total_disp_run_queue_time - Gesamtwartezeit in Ausführungswarteschlange des Dispatchers (Monitorelement)
ACT_THROUGHPUT	DOUBLE	act_throughput - Aktivitätsdurchsatz (Monitorelement)
CPU_UTILIZATION	DOUBLE	cpu_utilization - CPU-Auslastung (Monitorelement)
APP_ACT_COMPLETED_TOTAL	BIGINT	app_act_completed_total - Gesamtzahl erfolgreicher externer Koordinatoraktivitäten (Monitorelement)
APP_ACT_ABORTED_TOTAL	BIGINT	app_act_aborted_total - Gesamtzahl fehlgeschlagener externer Koordinatoraktivitäten (Monitorelement)
APP_ACT_REJECTED_TOTAL	BIGINT	app_act_rejected_total - Gesamtzahl zurückgewiesener externer Koordinatoraktivitäten (Monitorelement)
MEMBER	SMALLINT	member - Datenbankmember (Monitorelement)

WLM_SET_CLIENT_INFO (Prozedur) - Clientinformationen festlegen

Die Prozedur WLM_SET_CLIENT_INFO legt die Clientinformationen fest, die der aktuellen Verbindung auf dem DB2-Server zugeordnet sind.

Mit dieser Prozedur können Sie auf dem DB2-Server die Benutzer-ID, den Anwendungsnamen, den Workstationnamen, die Abrechnungsdaten oder die Workloadinformationen des Clients festlegen. Das Aufrufen dieser Prozedur bewirkt eine Än-

derung der gespeicherten Werte der entsprechenden Clientinformationsfelder für TP-Monitor sowie der Einstellungen des Sonderregisters für diese Verbindung.

Die Clientinformationsfelder werden auf dem DB2-Server zur Ermittlung der Identität der Anwendung oder des Benutzers verwendet, die bzw. der gerade die Verbindung verwendet. Die Clientinformationsfelder für eine Verbindung werden bei der DB2-Workloadbewertung berücksichtigt und werden auch in allen DB2-Prüfdatensätzen oder Anwendungsmomentaufnahmen angezeigt, die für diese Verbindung generiert wurden.

Anders als die API sqleseti legt diese Prozedur die Clientinformationen nicht auf dem Client fest, sondern legt stattdessen die entsprechenden Clientattribute auf dem DB2-Server fest. Daher können Sie die API sqleqry nicht zum Abrufen der Clientinformationen verwenden, die unter Verwendung dieser Prozedur auf dem DB2-Server festgelegt wurden. Wenn eine Anwendung die API sqleseti zum Ändern der Clientinformationen verwendet, wird die Einstellung auf dem DB2-Server durch diese neuen Werte geändert. Wird die API sqleseti zum Ändern der Benutzer-ID oder des Anwendungsnamens verwendet, ohne die Abrechnungsdaten zu ändern, werden die Abrechnungsdaten auf dem DB2-Server auf den Wert der Abrechnungsdaten auf dem Client zurückgesetzt.

Die mit der Prozedur bereitgestellten Datenwerte werden in die entsprechende Datenbankcodepage konvertiert, bevor sie in den zugehörigen Feldern für TP-Monitor oder für Sonderregister gespeichert werden. Datenwerte, die nach der Konvertierung in die Datenbankcodepage die maximal unterstützte Größe überschreiten, werden vor der Speicherung auf dem Server abgeschnitten. Die abgeschnittenen Werte werden bei einer Abfrage dieser gespeicherten Werte sowohl von den Feldern für TP-Monitor als auch von den Feldern für Sonderregister wiedergegeben.

Die Prozedur WLM_SET_CLIENT_INFO unterliegt nicht der Transaktionssteuerung und das Festschreiben von UOWs oder das Durchführen einer Rollback-Operation für UOWs hat auf die von der Prozedur vorgenommenen Änderungen der Clientinformationen keine Auswirkung. Da die erneute Workloadbewertung für jede Anwendung am Anfang der nächsten UOW erfolgt, müssen Sie eine Anweisung COMMIT oder eine Anweisung ROLLBACK absetzen, damit die Änderungen an den Clientinformationen wirksam werden.

Syntax

```
►—WLM_SET_CLIENT_INFO—(—client-benutzer-id—,—client-workstationname—,—  
►—clientanwendungsname—,—clientabrechnungszeichenfolge—,—client-workload—)—◄◄
```

Das Schema ist SYSPROC.

Prozedurparameter

client-benutzer-id

Ein Eingabeargument des Typs VARCHAR(255), das die Benutzer-ID für den Client angibt. Wenn Sie NULL angeben, bleibt der Wert unverändert. Wenn Sie eine leere Zeichenfolge eingeben (dies ist die Standardeinstellung), wird die Benutzer-ID für den Client auf den Standardwert (leer) zurückgesetzt.

clientworkstationname

Ein Eingabeargument des Typs VARCHAR(255), das den Workstationnamen für den Client angibt. Wenn Sie NULL angeben, bleibt der Wert unverändert.

Wenn Sie eine leere Zeichenfolge eingeben (dies ist die Standardeinstellung), wird der Workstationname für den Client auf den Standardwert (leer) zurückgesetzt.

clientanwendungsname

Ein Eingabeargument des Typs VARCHAR(255), das den Anwendungsnamen für den Client angibt. Wenn Sie NULL angeben, bleibt der Wert unverändert. Wenn Sie eine leere Zeichenfolge eingeben (dies ist die Standardeinstellung), wird der Anwendungsname für den Client auf den Standardwert (leer) zurückgesetzt.

clientabrechnungszeichenfolge

Ein Eingabeargument des Typs VARCHAR(200), das die Abrechnungszeichenfolge für den Client angibt. Wenn Sie NULL angeben, bleibt der Wert unverändert. Wenn Sie eine leere Zeichenfolge eingeben (dies ist die Standardeinstellung), wird die Abrechnungszeichenfolge für den Client auf den Standardwert (leer) zurückgesetzt.

client-workload

Ein Eingabeargument des Typs VARCHAR(255), das den Workloadzuordnungsmodus für den Client angibt. Wenn Sie NULL angeben, bleibt der Wert unverändert. Die Werte lauten wie folgt:

SYSDEFAULTADMWORKLOAD

Gibt an, dass die Datenbankverbindung der Workload SYSDEFAULTADMWORKLOAD zugeordnet wird, sodass Benutzer, die über die Berechtigung ACCESSCTRL, DATAACCESS, DBADM, SECADM oder WLMADM verfügen, die normale Workloadbewertung umgehen können.

AUTOMATIC

Gibt an, dass die Datenbankverbindung einer Workload zugeordnet wird, die von der automatisch durch den Server durchgeführten Workloadbewertung ausgewählt wird.

Anmerkung: Beim Argument *client-workload* muss die Groß-/Kleinschreibung beachtet werden.

Berechtigung

Zum Ausführen der Routine ist eine der folgenden Berechtigungen erforderlich:

- Zugriffsrecht EXECUTE für die Routine
- Berechtigung DATAACCESS
- Berechtigung DBADM
- Berechtigung SQLADM
- Berechtigung WLMADM

Standardzugriffsrecht für PUBLIC

Ohne

Beispiele

Der folgende Prozeduraufruf legt die Benutzer-ID, den Workstationnamen, den Anwendungsnamen, die Abrechnungszeichenfolge und den Workloadzuordnungsmodus für den Client fest:

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user', 'machine.torolab.ibm.com',  
'auditor', 'Accounting department', 'AUTOMATIC')
```

Der folgende Prozeduraufruf legt die Benutzer-ID für den Client auf 'db2user2' fest, ohne weitere Clientattribute festzulegen:

```
CALL SYSPROC.WLM_SET_CLIENT_INFO('db2user2', NULL, NULL, NULL, NULL)
```

Der folgende Prozeduraufruf setzt die Benutzer-ID für den Client auf ein Leerzeichen zurück, ohne die Werte der anderen Clientattribute zu ändern:

```
CALL SYSPROC.WLM_SET_CLIENT_INFO(' ', NULL, NULL, NULL, NULL)
```

Hinweise zur Verwendung

Wenn Ihre Eingabe für andere Prozedurparameter die angegebene Feldlänge überschreitet, wird das Eingabefeld abgeschnitten und die Prozedur wird unter Verwendung der abgeschnittenen Eingaben ausgeführt.

Eingabefelder, die einfache Anführungszeichen enthalten, werden nicht unterstützt und führen zur Ausgabe eines Fehlers.

Monitorelemente für Workload-Management

Die folgenden Monitorelemente stellen Informationen zu Aktivitäten, Schwellenwertverstößen und Statistikdaten zum Workload-Management bereit.

act_cpu_time_top - Maximaler Wert für CPU-Zeit von Aktivitäten (Monitorelement)

Der Höchstwert bei der Prozessorzeit, die von Aktivitäten auf allen Verschachtelungsebenen in einer Serviceklasse, Workload oder Arbeitsklasse verwendet wurde. Dieser Wert wird in Mikrosekunden angegeben.

Das Monitorelement gibt -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceklasse oder Workload, in der die Aktivität ausgeführt wird, auf NONE gesetzt ist. Aktivitäten werden bei diesem Höchstwert nur berücksichtigt, wenn Anforderungsmessdaten (Request Metrics) aktiviert sind. Wenn die Erfassung von Aktivitätsmessdaten nicht aktiviert ist, wird der Wert 0 zurückgegeben.

Bei Serviceklassen gilt: Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, wird nur der Höchstwert von 'act_cpu_time_top' der Serviceunterklasse aktualisiert, in der eine Aktivität abgeschlossen wird, sofern ein neuer Höchstwert erreicht wird. Die Höchstwerte für 'act_cpu_time_top' von anderen Serviceunterklassen, zu denen eine Aktivität zwar zugeordnet wird, in denen sie jedoch nicht abgeschlossen wird, bleiben unberührt.

Tabelle 79. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	Immer erfasst
Statistiken	event_wcstats	Immer erfasst
Statistiken	event_wlstats	Immer erfasst

Verwendung

Dieses Element gibt Aufschluss über die größte Menge verwendeter Prozessorzeit, die im erfassten Zeitraum von einer Aktivität für eine Serviceklasse, Workload oder Arbeitsklasse verwendet wurde.

act_exec_time - Ausführungszeit für Aktivitäten (Monitorelement)

Das Element 'act_exec_time' speichert die in diesem Member verbrachte Ausführungszeit in Mikrosekunden.

Bei Cursors ist die Ausführungszeit die kombinierte Zeit für das Öffnen, das Abrufen und das Schließen. Die Zeit, in der ein Cursor inaktiv ist, zählt nicht zur Ausführungszeit. Für Routinen umfasst die Ausführungszeit die Zeit vom Start bis zum Ende des Routinenaufrufs. Die Laufzeit von Cursors, die von einer Routine nach deren Beendigung offen gelassen werden (zur Rückgabe einer Ergebnismenge), zählt nicht zur Ausführungszeit der Routine. Bei allen anderen Aktivitäten ist die Ausführungszeit die Differenz zwischen Start- und Stoppzeit. Für alle diese Fälle gilt: Die Ausführungszeit umfasst weder die Zeit für die Initialisierung noch die in einer Warteschlange verbrachte Zeit.

Tabelle 80. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element kann allein verwendet werden, um die abgelaufene Zeit anzuzeigen, die von DB2 in jedem Member zur Ausführung der Aktivität verbraucht wurde. Dieses Element kann auch zusammen mit den Monitorelementen **time_started** und **time_completed** für das Koordinatormember verwendet werden, um die Leerlaufzeit für Cursoraktivitäten zu berechnen. Hierfür wird folgende Formel verwendet:

Cursor id+le time = (time_completed - time_started) - act_exec_time

act_remapped_in - Neu zugeordnete Aktivitäten in Service (Monitorelement)

Zähler für die Anzahl der Aktivitäten, die dieser Serviceunterklasse seit der letzten Zurücksetzung zugeordnet wurden.

Tabelle 81. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-

Verwendung

Mithilfe dieses Zähler kann ermittelt werden, ob die Neuordnung von Aktivitäten zur betreffenden Serviceunterklasse den Erwartungen entspricht.

act_remapped_out – Neu zugeordnete Aktivitäten aus Service (Monitorelement)

Zähler für die Anzahl der Aktivitäten, die aus dieser Serviceunterklasse seit der letzten Zurücksetzung zugeordnet wurden.

Tabelle 82. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-

Verwendung

Mithilfe dieses Zähler kann ermittelt werden, ob die Neuuzuordnung von Aktivitäten aus der betreffenden Serviceunterklasse den Erwartungen entspricht.

act_rows_read_top - Maximaler Wert für von Aktivitäten gelesene Zeilen (Monitorelement)

Der Höchstwert bei der Anzahl an Zeilen, die von Aktivitäten auf allen Verschachtelungsebenen in einer Serviceklasse, Workload oder Arbeitsklasse gelesen wurden.

Das Monitorelement gibt -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceklasse oder Workload, in der die Aktivität ausgeführt wird, auf NONE gesetzt ist. Aktivitäten werden bei diesem Höchstwert nur berücksichtigt, wenn Anforderungsmessdaten (Request Metrics) aktiviert sind. Wenn die Erfassung von Aktivitätsmessdaten nicht aktiviert ist, wird der Wert 0 zurückgegeben.

Bei Serviceklassen gilt: Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, wird nur der Höchstwert von 'act_rows_read_top' der Serviceunterklasse aktualisiert, in der eine Aktivität abgeschlossen wird, sofern ein neuer Höchstwert erreicht wird. Die Höchstwerte für 'act_rows_read_top' von Serviceunterklassen, zu denen eine Aktivität zwar zugeordnet wird, in denen sie jedoch nicht abgeschlossen wird, bleiben unberührt.

Tabelle 83. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	Immer erfasst
Statistiken	event_wcstats	Immer erfasst
Statistiken	event_wlstats	Immer erfasst

Verwendung

Dieses Element gibt Aufschluss über die größte Anzahl gelesener Zeilen, die im erfassten Zeitraum von einer Aktivität für eine Serviceklasse, Workload oder Arbeitsklasse gelesen wurden.

act_throughput - Aktivitätsdurchsatz (Monitorelement)

Die Rate, mit der Koordinatoraktivitäten auf einer beliebigen Verschachtelungsebene ausgeführt werden. Der Wert wird in Koordinatoraktivitäten pro Sekunde gemessen.

Tabelle 84. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	REQUEST METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 85. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst

Verwendung

Bei Rückgabe durch die Funktion WLM_GET_SERVICE_SUBCLASS_STATS oder WLM_GET_WORKLOAD_STATS stellt dieses Monitorelement den Aktivitätsdurchsatz seit der letzten Zurücksetzung der Statistiken dar.

Bei Rückgabe durch die Tabellenfunktion MON_SAMPLE_SERVICE_CLASS_METRICS oder MON_SAMPLE_WORKLOAD_METRICS stellt dieses Monitorelement den Aktivitätsdurchsatz seit der Ausführung der Funktion dar.

act_total - Gesamtzahl der Aktivitäten (Monitorelement)

Die Gesamtzahl der Aktivitäten auf allen Verschachtelungsebenen, auf die seit dem letzten Zurücksetzen Arbeitsaktionen angewendet wurden, die der angegebenen Arbeitsklasse entsprechen.

Tabelle 86. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_WORK_ACTION_SET_STATS (Tabellenfunktion) - Statistikdaten zu Arbeitsaktionssets zurückgeben	ACTIVITY METRICS BASE

Tabelle 87. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wcstats	-

Verwendung

Jedes Mal, wenn auf eine Aktivität mindestens eine Arbeitsaktion angewendet wird, die einer Arbeitsklasse zugeordnet ist, wird ein Zähler für die Arbeitsklasse aktualisiert. Dieser Zähler wird mithilfe des Monitorelements **act_total** dargestellt. Mit diesem Zähler kann die Effektivität des Arbeitsaktionssets bewertet werden (z. B. auf wie viele Aktivitäten Aktionen angewendet wurden). Darüber hinaus gibt dieser Zähler Aufschluss über die verschiedenen Aktivitätstypen auf dem System.

activate_timestamp - Zeitmarke für Aktivierung (Monitorelement)

Der Zeitpunkt, zu dem ein Ereignismonitor aktiviert wurde.

Tabelle 88. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivität	event_activity	-
Aktivität	event_activitystmt	-
Aktivität	event_activityvals	-
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Mithilfe dieses Elements können Informationen korreliert werden, die von den zuvor genannten Ereignistypen zurückgegeben werden.

activity_collected - Aktivitäten erfassen (Monitorelement)

Dieses Element gibt an, ob für einen Schwellenwert, gegen den verstoßen wurde, Datensätze des Aktivitätsereignismonitors erfasst werden sollen.

Tabelle 89. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Mit diesem Element lässt sich ermitteln, ob für eine Aktivität, die gegen den Schwellenwert verstoßen hat, ein an den Aktivitätsereignismonitor geschriebenes Aktivitätsereignis erwartet werden muss.

Wird eine Aktivität beendet oder abgebrochen und ist der Aktivitätsereignismonitor zu diesem Zeitpunkt aktiv, wird die Aktivität, die gegen den betreffenden Schwellenwert verstoßen hat, erfasst, sofern der Wert dieses Monitorelements 'Y' ist. Ist der Wert dieses Monitorelements 'N', wird die Aktivität nicht erfasst.

activity_id - Aktivitäts-ID (Monitorelement)

Ein Zähler, der eine Aktivität für eine Anwendung innerhalb einer bestimmten UOW eindeutig identifiziert.

Tabelle 90. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 91. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Sperren	-	Immer erfasst
Aktivitäten	event_activity	Immer erfasst
Aktivitäten	event_activitystmt	Immer erfasst
Aktivitäten	event_activityvals	Immer erfasst
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Schwellenwertverstöße	event_thresholdviolations	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse des Verhaltens einer Aktivität.

Um eine Aktivität außerhalb ihrer jeweiligen UOW eindeutig zu identifizieren, verwenden Sie die Werte von **activity_id** und **uow_id** plus dem Wert eines der folgenden Monitorelemente: **apl_id** oder **agent_id**.

activity_secondary_id - Sekundäre Aktivitäts-ID (Monitorelement)

Der Wert für dieses Element wird schrittweise erhöht, sobald ein weiterer Aktivitätsdatensatz für eine bestimmte Aktivität geschrieben wird.

Wird z. B. für eine Aktivität ein Aktivitätsdatensatz geschrieben, weil die Prozedur WLM_CAPTURE_ACTIVITY_IN_PROGRESS aufgerufen wurde, und ein weiterer Datensatz beim Beenden der Aktivität, weist das Element für den ersten Datensatz den Wert 0 und für den zweiten Datensatz den Wert 1 auf.

Tabelle 92. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	-
Aktivitäten	event_activitystmt	-
Aktivitäten	event_activityvals	-
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE

Verwendung

In Verbindung mit den Monitorelementen **activity_id**, **uow_id** und **appl_id** können Sie mit diesem Element Aktivitätsdatensätze eindeutig identifizieren, wenn Informationen zu einer Aktivität mehrmals in den Aktivitätsereignismonitor geschrieben wurden.

Beispiel: Informationen zu einer Aktivität werden in den folgenden Fällen zweimal an den Aktivitätsereignismonitor gesendet:

- Unter Verwendung der gespeicherten Prozedur `WLM_CAPTURE_ACTIVITY_IN_PROGRESS` wurden während der Ausführung der Aktivität Informationen zur Aktivität erfasst.
- Informationen zur Aktivität wurden nach Ausführung der Aktivität erfasst, da die Klausel `COLLECT ACTIVITY DATA` für die Serviceklasse angegeben wurde, der die Aktivität zugeordnet ist.

activity_type - Aktivitätstyp (Monitorelement)

Der Typ der Aktivität.

Tabelle 93. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 94. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Die möglichen Werte sind:

- LOAD
- READ_DML
- WRITE_DML
- DDL
- CALL
- OTHER

Der Wert OTHER wird für SET-Anweisungen zurückgegeben, die kein SQL ausführen (z. B. SET SPECIAL REGISTER oder SET EVENT MONITOR STATE), sowie für die Anweisung LOCK TABLE.

agg_temp_tablespace_top - Zusammengefasster maximaler Wert für Tabellenbereiche für temporäre Tabellen (Monitorelement)

Das Monitorelement 'agg_temp_tablespace_top' speichert den Höchstwert (in KB) für die zusammengefasste Belegung der Tabellenbereiche für temporäre Tabellen durch DML-Aktivitäten auf allen Verschachtelungsebenen innerhalb einer Serviceklasse.

Die Ergebnistabelle mit dem zusammengefassten Wert wird berechnet, indem die Werte für die Belegung der Tabellenbereiche für temporäre Tabellen durch alle Aktivitäten in der Serviceunterklasse addiert werden. Dieser Höchstwert stellt den maximalen Wert dar, der seit der letzten Zurücksetzung in dieser Ergebnistabelle erreicht wurde. Das Monitorelement gibt -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceklasse auf NONE gesetzt ist. Ein Schwellenwert für AGGSQLTEMPSPACE muss für mindestens eine Serviceunterklasse in derselben Superklasse definiert und aktiviert werden wie die der Unterklasse, zu der dieser Datensatz gehört. Andernfalls wird der Wert 0 zurückgegeben.

Tabelle 95. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	Immer erfasst

Verwendung

Mithilfe dieses Elements können Sie die Summe der höchsten Nutzung von Tabellenbereich für temporäre Systemtabellen für DML-Aktivitäten ermitteln, die im erfassten Zeitintervall in einem Member für eine Serviceunterklasse erreicht wurde.

arm_correlator - Korrelator zur Messung der Anwendungsantwortzeit (Monitorelement)

Transaktionskennung gemäß dem ARM-Standard (Application Response Measurement).

Tabelle 96. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	-

Verwendung

Dieses Element kann dazu verwendet werden, eine vom Ereignismonitor für Aktivitäten erfasste Aktivität mit den Anwendungen zu verknüpfen, die der Aktivität zugeordnet sind. Voraussetzung ist jedoch, dass die betreffenden Anwendungen den ARM-Standard (Application Response Measurement) unterstützen.

bin_id - ID des Histogrammbins (Monitorelement)

Die ID eines Histogrammbins. Der Wert für **bin_id** ist innerhalb des Histogramms eindeutig.

Tabelle 97. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_histogrambin	-

Verwendung

Dieses Element ermöglicht die Unterscheidung von Bins innerhalb eines Histogramms.

bottom - Untere Grenze von Histogrammbin (Monitorelement)

Die Untergrenze (nicht Teil des Bereichs) des für ein Histogrammbin festgelegten Bereichs. Der Wert dieses Monitorelements stellt gleichzeitig die Obergrenze (Teil des Bereichs) des Bereichs des vorangehenden Histogrammbins dar, soweit vorhanden.

Table 98. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_histogrambin	-

Verwendung

Mithilfe dieses Elements kann in Verbindung mit dem entsprechenden Element **top** der Bereich für ein Bin in einem Histogramm bestimmt werden.

concurrent_act_top - Maximaler Wert für gleichzeitige Aktivitäten (Monitorelement)

Der Höchstwert für die gleichzeitig ablaufenden Aktivitäten (auf allen Verschachtelungsebenen) in einer Serviceunterklasse nach dem letzten Zurücksetzen.

Anmerkung: Dieses Element überwacht den Höchstwert für die gleichzeitige Ausführung von Aktivitäten, einschließlich der Aktivitäten, die nicht am Schwellenwert CONCURRENTDBCOORDACTIVITIES teilnehmen. Obwohl beispielsweise CALL-Anweisungen nicht zu dem gemeinsamen Zugriff gerechnet werden, der vom Schwellenwert CONCURRENTDBCOORDACTIVITIES erzwungen wird, werden sie in die Messungen für den Höchstwert für gleichzeitig ablaufende Aktivitäten eingeschlossen.

Table 99. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE

Table 100. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-

Verwendung

Verwenden Sie dieses Element, um den höchsten gemeinsamen Zugriff von Aktivitäten (einschließlich verschachtelter Aktivitäten) zu ermitteln, der im erfassten Zeitintervall in einem Member für eine Serviceunterklasse erreicht wurde.

concurrent_connection_top - Maximaler Wert für gleichzeitige Verbindungen (Monitorelement)

Der Höchstwert für gleichzeitige Koordinatorverbindungen der Serviceklasse, der nach dem letzten Zurücksetzen erreicht wurde. Dieses Feld weist in den einzelnen Unterklassen einer Superklasse denselben Wert auf.

Tabelle 101. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUPERCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Servicesuperklassen zurückgeben	ACTIVITY METRICS BASE

Tabelle 102. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-

Verwendung

Dieses Element gibt durch die Angabe des aktuellen Höchstwerts Hinweise darauf, an welcher Stelle Schwellenwerte für die Gleichzeitigkeit von Verbindungen festgelegt werden sollten. Darüber hinaus kann mit diesem Element überprüft werden, ob ein Schwellenwert korrekt definiert ist und seinen Zweck erfüllt.

concurrent_wlo_act_top - Maximaler Wert für gleichzeitige Workloadaktivitäten (Monitorelement)

Der Höchstwert für gleichzeitig ablaufende Aktivitäten (auf allen Verschachtelungsebenen) für beliebige Vorkommen einer bestimmten Workload seit dem letzten Zurücksetzen.

Tabelle 103. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 104. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wlstats	-

Verwendung

Mithilfe dieses Elements können Sie die höchste Anzahl gleichzeitig ablaufender Aktivitäten ermitteln, die in einem Member für ein Vorkommen dieser Workload im erfassten Zeitintervall erreicht wurde.

concurrent_wlo_top - Maximaler Wert für gleichzeitige Workloadvorkommen (Monitorelement)

Der Höchstwert für die gleichzeitige Ausführung von Verarbeitungsprozessen seit dem letzten Zurücksetzen.

Table 105. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Table 106. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wlstats	-
Statistiken	event_scstats	-

Verwendung

Verwenden Sie dieses Element, um den höchsten gemeinsamen Zugriff von Workloadvorkommen zu ermitteln, der im erfassten Zeitintervall in einem Member für eine Workload erreicht wurde.

concurrentdbcoordactivities_db_threshold_id - Kennung des Datenbankschwellenwerts für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Die Kennung (ID) des Datenbankschwellenwerts für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde.

Table 107. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, welcher Datenbankschwellenwert für CONCURRENTDBCOORDACTIVITIES (sofern zutreffend) auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_db_threshold_queued - Durch Datenbankschwellenwert für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität durch den Datenbankschwellenwert für CONCURRENTDBCOORDACTIVITIES in die Warteschlange gestellt wurde. Der Wert 'No' gibt an, dass die Aktivität nicht in die Warteschlange gestellt wurde.

Tabelle 108. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität durch den Datenbankschwellenwert für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde, in die Warteschlange gestellt wurde.

concurrentdbcoordactivities_db_threshold_value - Wert des Datenbankschwellenwerts für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt die Obergrenze des Datenbankschwellenwerts für CONCURRENTDBCOORDACTIVITIES zurück, der auf die Aktivität angewendet wurde.

Tabelle 109. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie den Wert des Datenbankschwellenwerts für CONCURRENTDBCOORDACTIVITIES ermitteln, der auf die Aktivität angewendet wurde (sofern zutreffend).

concurrentdbcoordactivities_db_threshold_violated - Verstoß gegen den Datenbankschwellenwert für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität gegen den Datenbankschwellenwert für CONCURRENTDBCOORDACTIVITIES verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.

Tabelle 110. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität gegen den Datenbankschwellenwert für CONCURRENTDBCOORDACTIVITIES verstoßen hat, der auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_subclass_threshold_id - Kennung des Schwellenwerts für Serviceunterklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt die Kennung (ID) des Schwellenwerts für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES zurück, der auf die Aktivität angewendet wurde.

Tabelle 111. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, welcher Schwellenwert für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES (sofern zutreffend) auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_subclass_threshold_queued - Durch Schwellenwert für Serviceunterklasse für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität durch den Schwellenwert für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES in die Warteschlange gestellt wurde. Der Wert 'No' gibt an, dass die Aktivität nicht in die Warteschlange gestellt wurde.

Tabelle 112. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität durch den Schwellenwert für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde, in die Warteschlange gestellt wurde.

concurrentdbcoordactivities_subclass_threshold_value - Wert des Schwellenwerts für Serviceunterklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt die Obergrenze des Schwellenwerts für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES zurück, der auf die Aktivität angewendet wurde.

Tabelle 113. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
WLM_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Detaillierte Informationen zu einer bestimmten Aktivität zurückgeben	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie den Wert des Schwellenwerts für Serviceunterklasse CONCURRENTDBCOORDACTIVITIES ermitteln, der auf die Aktivität angewendet wurde (sofern zutreffend).

concurrentdbcoordactivities_subclass_threshold_violated - Verstoß gegen den Schwellenwert für Serviceunterklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität gegen den Schwellenwert für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.

Tabelle 114. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität gegen den Schwellenwert für Serviceunterklasse für CONCURRENTDBCOORDACTIVITIES verstoßen hat, der auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_superclass_threshold_id - Kennung des Schwellenwerts für Servicesuperklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Die Kennung (ID) des Schwellenwerts für CONCURRENTDBCOORDACTIVITIES_SUPERCLASS, der auf die Aktivität angewendet wurde.

Tabelle 115. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, welcher Schwellenwert für Servicesuperklasse für CONCURRENTDBCOORDACTIVITIES (sofern zutreffend) auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_superclass_threshold_queued - Durch Schwellenwert für Servicesuperklasse für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität durch den Schwellenwert für Servicesuperklasse für CONCURRENTDBCOORDACTIVITIES in die Warteschlange gestellt wurde. Der Wert 'No' gibt an, dass die Aktivität nicht in die Warteschlange gestellt wurde.

Tabelle 116. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität durch den Schwellenwert für Servicesuperklasse für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde, in die Warteschlange gestellt wurde.

concurrentdbcoordactivities_superclass_threshold_value - Wert des Schwellenwerts für Servicesuperklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Die Obergrenze des Schwellenwerts für Servicesuperklasse für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde.

Tabelle 117. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie den Wert des Schwellenwerts für Servicesuperklasse CONCURRENTDBCOORDACTIVITIES ermitteln, der auf die Aktivität angewendet wurde (sofern zutreffend).

concurrentdbcoordactivities_superclass_threshold_violated - Verstoß gegen den Schwellenwert für Servicesuperklasse für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität gegen den Schwellenwert für Servicesuperklasse für CONCURRENTDBCOORDACTIVITIES verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.

Tabelle 118. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität gegen den Schwellenwert für Servicesuperklasse für CONCURRENTDBCOORDACTIVITIES verstoßen hat, der auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_wl_was_threshold_id - Kennung des Schwellenwerts für Workload-Arbeitsaktionssets für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Die Kennung des Schwellenwerts für Workload-Arbeitsaktionssets für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde.

Tabelle 119. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, welcher Schwellenwert für Workload-Arbeitsaktionssets für CONCURRENTDBCOORDACTIVITIES (sofern zutreffend) auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_wl_was_threshold_queued - Durch Schwellenwert für Workload-Arbeitsaktionsset für gleichzeitige Datenbankkoordinatoraktivitäten in Warteschlange gestellt (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität durch den Schwellenwert für das Workload-Arbeitsaktionsset für CONCURRENTDBCOORDACTIVITIES in die Warteschlange gestellt wurde. Der Wert 'No' gibt an, dass die Aktivität nicht in die Warteschlange gestellt wurde.

Tabelle 120. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität durch den Schwellenwert für Workload-Arbeitsaktionssets für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde, in die Warteschlange gestellt wurde.

concurrentdbcoordactivities_wl_was_threshold_value - Wert des Schwellenwerts für Workload-Arbeitsaktionssets für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Die Obergrenze des Schwellenwerts für Workload-Arbeitsaktionssets für CONCURRENTDBCOORDACTIVITIES, der auf die Aktivität angewendet wurde.

Tabelle 121. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie den Wert des Schwellenwerts für Workload-Arbeitsaktionssets für CONCURRENTDBCOORDACTIVITIES ermitteln, der auf die Aktivität angewendet wurde.

concurrentdbcoordactivities_wl_was_threshold_violated - Verstoß gegen den Schwellenwert für Workload-Arbeitsaktionssets für gleichzeitige Datenbankkoordinatoraktivitäten (Monitorelement)

Dieses Monitorelement gibt den Wert 'Yes' zurück, um anzugeben, dass die Aktivität gegen den Schwellenwert für Workload-Arbeitsaktionssets für CONCURRENT-DBCOORDACTIVITIES verstoßen hat. Der Wert 'No' gibt an, dass die Aktivität noch nicht gegen den Schwellenwert verstoßen hat.

Tabelle 122. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können Sie ermitteln, ob die Aktivität gegen den Schwellenwert für Workload-Arbeitsaktionssets für CONCURRENTDBCOORDACTIVITIES verstoßen hat, der auf die Aktivität angewendet wurde.

coord_act_aborted_total - Gesamtzahl abgebrochener Koordinatoraktivitäten (Monitorelement)

Die Gesamtzahl von Koordinatoraktivitäten auf allen Verschachtelungsebenen, die seit dem letzten Zurücksetzen mit Fehler beendet wurden. Bei Serviceklassen wird dieser Wert beim Beenden von Aktivitäten aktualisiert. Bei Workloads wird der Wert bei Abschluss der zugehörigen UOW mit den einzelnen Workloadvorkommen aktualisiert.

Wenn eine Aktivität für eine Serviceklasse durch die Aktion REMAP ACTIVITY einer anderen Unterklasse neu zugeordnet wird, bevor sie abgebrochen wird, wird diese Aktivität nur für die Gesamtzahl der Unterklasse gezählt, in der sie abgebrochen wird.

Tabelle 123. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 124. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-

Tabelle 124. Informationen zur Ereignisüberwachung (Forts.)

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wlstats	-

Verwendung

Dieses Element gibt Aufschluss darüber, ob Aktivitäten auf dem System erfolgreich abgeschlossen werden. Aktivitäten können storniert oder aufgrund von Fehlern oder reaktiven Schwellenwerten vorzeitig abgebrochen werden.

coord_act_completed_total - Gesamtzahl abgeschlossener Koordinatoraktivitäten (Monitorelement)

Die Gesamtzahl von Koordinatoraktivitäten auf allen Verschachtelungsebenen, die seit dem letzten Zurücksetzen erfolgreich abgeschlossen wurden. Bei Serviceklassen wird dieser Wert beim Beenden von Aktivitäten aktualisiert. Bei Workloads wird der Wert bei Abschluss der zugehörigen UOW mit den einzelnen Workloadvorkommen aktualisiert.

Wenn eine Aktivität für eine Serviceklasse durch die Aktion REMAP ACTIVITY einer anderen Unterklasse neu zugeordnet wird, bevor sie abgeschlossen ist, wird diese Aktivität nur für die Gesamtzahl der Unterklasse gezählt, in der sie abgeschlossen wird.

Tabelle 125. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 126. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wlstats	-
Statistiken	event_scstats	-

Verwendung

Mithilfe dieses Elements kann der Durchsatz von Aktivitäten im System ermittelt oder die durchschnittliche Aktivitätslaufzeit für mehrere Member berechnet werden.

coord_act_est_cost_avg - Durchschnittswert für den geschätzten Aufwand für Koordinatoraktivitäten (Monitorelement)

Das arithmetische Mittel des geschätzten Aufwands für DML-Koordinatoraktivitäten auf der Verschachtelungsebene 0, die dieser Serviceunterklasse oder Arbeitsklasse seit dem letzten Zurücksetzen zugeordnet wurden.

Wenn für den intern aufgezeichneten Durchschnittswert ein Überlauf stattgefunden hat, wird der Wert -2 zurückgegeben. Bei Serviceunterklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceunterklasse auf NONE oder BASE gesetzt ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE oder BASE gesetzt ist. Der Wert wird in Timern gemessen.

Bei Serviceklassen wird die Kostenschätzung für eine Aktivität nur zu der Serviceunterklasse gezählt, in der die Aktivität in das System eintritt. Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, bleibt der Wert von 'coord_act_est_cost_avg' der Serviceunterklasse, zu der Sie eine Aktivität zuordnen, unberührt.

Table 127. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	Immer erfasst
Statistiken	event_wcstats	Immer erfasst
Statistiken	event_wlstats	Immer erfasst

Verwendung

Mit dieser Statistik können Sie das arithmetische Mittel des geschätzten Aufwands für DML-Koordinatoraktivitäten auf der Verschachtelungsebene 0 ermitteln, die einer abgeschlossenen oder abgebrochenen Serviceunterklasse, Workload oder Arbeitsklasse seit dem letzten Zurücksetzen zugeordnet wurden.

Ferner kann mithilfe dieses Durchschnittswerts festgestellt werden, ob die verwendete Histogrammschablone für das Histogramm des geschätzten Aufwands für Aktivitäten geeignet ist. Mit dem Histogramm des geschätzten Aufwands für Aktivitäten kann der Durchschnittswert für den geschätzten Aufwand für Aktivitäten berechnet werden. Vergleichen Sie den berechneten Durchschnittswert mit diesem Monitorelement. Wenn der berechnete Durchschnittswert vom tatsächlichen Durchschnittswert abweicht, der von diesem Monitorelement zurückgemeldet wird, sollten Sie eine Änderung der Histogrammschablone für das Histogramm des geschätzten Aufwands für Aktivitäten in Erwägung ziehen. Verwenden Sie dazu eine Gruppe von Binwerten, die für Ihre Daten besser geeignet sind.

coord_act_exec_time_avg - Durchschnittswert für die Ausführungszeit von Koordinatoraktivitäten (Monitorelement)

Das arithmetische Mittel der Ausführungszeiten von Koordinatoraktivitäten auf der Verschachtelungsebene 0, die dieser Serviceunterklasse oder Arbeitsklasse seit dem letzten Zurücksetzen zugeordnet wurden.

Wenn für den intern aufgezeichneten Durchschnittswert ein Überlauf stattgefunden hat, wird der Wert -2 zurückgegeben. Bei Serviceunterklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceunterklasse auf NONE gesetzt ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE gesetzt ist. Die Zeit wird in Millisekunden gemessen.

Bei Serviceklassen gilt: Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, bleibt der Durchschnittswert von 'coord_act_exec_time_avg' der Serviceunterklassen, zu denen eine Aktivität zugeordnet, in denen sie aber nicht abgeschlossen wird, unberührt.

Tabelle 128. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	COLLECT AGGREGATE ACTIVITY DATA
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	COLLECT AGGREGATE ACTIVITY DATA

Tabelle 129. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wcstats	-
Statistiken	event_wlstats	-

Verwendung

Mit dieser Statistik können Sie das arithmetische Mittel der Ausführungszeit für Koordinatoraktivitäten ermitteln, die einer abgeschlossenen oder abgebrochenen Serviceunterklasse, Workload oder Arbeitsklasse zugeordnet wurden.

Ferner kann mithilfe dieser Statistik festgestellt werden, ob die verwendete Histogrammschablone für das Histogramm der Ausführungszeit von Aktivitäten geeignet ist. Mit dem Histogramm der Ausführungszeit von Aktivitäten kann die durchschnittliche Ausführungszeit von Aktivitäten berechnet werden. Vergleichen Sie den berechneten Durchschnittswert mit diesem Monitorelement. Wenn der berechnete Durchschnittswert vom tatsächlichen Durchschnittswert abweicht, der von diesem Monitorelement zurückgemeldet wird, sollten Sie eine Änderung der Histogrammschablone für das Histogramm der Ausführungszeit von Aktivitäten in Erwägung ziehen. Verwenden Sie dazu eine Gruppe von Binwerten, die für Ihre Daten besser geeignet sind.

coord_act_interarrival_time_avg - Durchschnittswert für die Ankunftszeit von Koordinatoraktivitäten (Monitorelement)

Das arithmetische Mittel der Zeit zwischen der Ankunft von Koordinatoraktivitäten auf der Verschachtelungsebene 0, die dieser Serviceunterklasse oder Arbeitsklasse seit dem letzten Zurücksetzen zugeordnet wurden.

Wenn für den intern aufgezeichneten Durchschnittswert ein Überlauf stattgefunden hat, wird der Wert -2 zurückgegeben. Bei Serviceunterklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceunterklasse auf NONE oder BASE gesetzt ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE oder BASE gesetzt ist. Der Wert wird in Millisekunden gemessen.

Bei Serviceklassen gilt: Die durchschnittliche Zwischenankunftszeit (Inter-Arrival Time) wird für Serviceunterklassen berechnet, über die Aktivitäten in das System eintreten. Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion RE-MAP ACTIVITY neu zuordnen, bleibt der Wert von 'coord_act_interarrival_time_avg' der Serviceunterklasse, zu der Sie eine Aktivität zuordnen, unberührt.

Tabelle 130. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wcstats	-
Statistiken	event_wlstats	-

Verwendung

Mit dieser Statistik können Sie das arithmetische Mittel der Zeit zwischen der Ankunft von Koordinatoraktivitäten auf der Verschachtelungsebene 0 ermitteln, die der Serviceunterklasse, Workload oder Arbeitsklasse zugeordnet wurden.

Mit der Zeit zwischen der Ankunft von Aktivitäten kann die Ankunftsrate ermittelt werden, die den Kehrwert der Zeit zwischen der Ankunft von Aktivitäten darstellt. Ferner kann mithilfe dieses Durchschnittswerts festgestellt werden, ob die verwendete Histogrammschablone für das Histogramm der Zeit zwischen der Ankunft von Aktivitäten geeignet ist. Mit dem Histogramm der Zeit zwischen der Ankunft von Aktivitäten kann die durchschnittliche Zeit zwischen der Ankunft von Aktivitäten berechnet werden. Vergleichen Sie den berechneten Durchschnittswert mit diesem Monitorelement. Wenn der berechnete Durchschnittswert vom tatsächlichen Durchschnittswert abweicht, der von diesem Monitorelement zurückgegeben wird, sollten Sie eine Änderung der Histogrammschablone für das Histogramm der Zeit zwischen der Ankunft von Aktivitäten in Erwägung ziehen. Verwenden Sie dazu eine Gruppe von Binwerten, die für Ihre Daten besser geeignet sind.

coord_act_lifetime_avg - Durchschnittswert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)

Das arithmetische Mittel der Laufzeit von Koordinatoraktivitäten auf der Verschachtelungsebene 0, die der betreffenden Serviceunterklasse, Workload oder Arbeitsklasse seit dem letzten Zurücksetzen zugeordnet wurden.

Wenn für den intern aufgezeichneten Durchschnittswert ein Überlauf stattgefunden hat, wird der Wert -2 zurückgegeben. Bei Serviceunterklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceunterklasse auf NONE gesetzt ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement

den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE gesetzt ist. Der Wert wird in Millisekunden gemessen.

Bei Serviceklassen gilt: Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, ist nur der Durchschnittswert von 'coord_act_lifetime_avg' der letzten Serviceklasse betroffen, in der eine Aktivität abgeschlossen wird.

Tabelle 131. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	COLLECT AGGREGATE ACTIVITY DATA
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	COLLECT AGGREGATE ACTIVITY DATA

Tabelle 132. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wcstats	-
Statistiken	event_wlstats	-

Verwendung

Mit dieser Statistik können Sie das arithmetische Mittel der Laufzeit für Koordinatoraktivitäten ermitteln, die einer abgeschlossenen oder abgebrochenen Serviceunterklasse, Workload oder Arbeitsklasse zugeordnet wurden.

Ferner kann mithilfe dieser Statistik festgestellt werden, ob die verwendete Histogrammschablone für das Histogramm der Laufzeit von Aktivitäten geeignet ist. Mit dem Histogramm der Laufzeit von Aktivitäten kann die durchschnittliche Laufzeit von Aktivitäten berechnet werden. Vergleichen Sie den berechneten Durchschnittswert mit diesem Monitorelement. Wenn der berechnete Durchschnittswert vom tatsächlichen Durchschnittswert abweicht, der von diesem Monitorelement zurückgemeldet wird, sollten Sie eine Änderung der Histogrammschablone für das Histogramm der Laufzeit von Aktivitäten in Erwägung ziehen. Verwenden Sie dazu eine Gruppe von Binwerten, die für Ihre Daten besser geeignet sind.

coord_act_lifetime_top - Maximaler Wert für die Laufzeit von Koordinatoraktivitäten (Monitorelement)

Das Element 'coord_act_lifetime_top' ist der Höchstwert für die Laufzeit von Koordinatoraktivitäten. Der Wert wird für alle Verschachtelungsebenen berechnet. Die gespeicherten Informationen werden in Millisekunden angegeben.

Bei Serviceklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Serviceklasse auf NONE gesetzt ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Arbeitsklasse nicht de-

finiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE gesetzt ist.

Um diese Statistikdaten effektiv mit Serviceklassen verwenden zu können, wenn Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY auch neu zugeordnet werden, müssen Sie den Höchstwert des Elements 'coord_act_lifetime_top' einer bestimmten Serviceunterklasse mit dem Wert anderer Unterklassen zusammenfassen, die von demselben Schwellenwert bzw. denselben Schwellenwerten für Neuordnungen betroffen sind. Dies liegt daran, dass eine Aktivität abgeschlossen wird, nachdem sie durch einen Schwellenwert für Neuordnungen einer anderen Serviceunterklasse neu zugeordnet worden ist, und dass die Zeit, die die Aktivität vor der Neuordnung in anderen Serviceunterklassen vorhanden ist, nur für die Serviceklasse gezählt wird, in der sie abgeschlossen wird.

Tabelle 133. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	COLLECT AGGREGATE ACTIVITY DATA
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	COLLECT AGGREGATE ACTIVITY DATA

Tabelle 134. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wcstats	-
Statistiken	event_scstats	-
Statistiken	event_wlstats	-

Verwendung

Dieses Element ist beim Beurteilen der Effektivität von Schwellenwerten für die Laufzeit von Aktivitäten hilfreich und somit auch eine wertvolle Entscheidungshilfe für die Konfiguration derartiger Schwellenwerte.

coord_act_queue_time_avg - Durchschnittliche Warteschlangenzeit für Koordinatoraktivitäten (Monitorelement)

Das arithmetische Mittel der Warteschlangenzeit für Koordinatoraktivitäten auf der Verschachtelungsebene 0, die der Serviceunterklasse oder Arbeitsklasse seit dem letzten Zurücksetzen zugeordnet wurden.

Wenn für den intern aufgezeichneten Durchschnittswert ein Überlauf stattgefunden hat, wird der Wert -2 zurückgegeben. Bei Serviceunterklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die Serviceunterklasse auf NONE gesetzt ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE gesetzt ist. Der Wert wird in Millisekunden gemessen.

Bei Serviceklassen wird die Zeit in der Warteschlange nur zu der Serviceunterklasse gezählt, in der die Aktivität abgeschlossen oder abgebrochen wird. Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, bleibt der Durchschnittswert von 'coord_act_queue_time_avg' der Serviceunterklassen, zu denen eine Aktivität zugeordnet, in denen sie aber nicht abgeschlossen wird, unberührt.

Tabelle 135. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	COLLECT AGGREGATE ACTIVITY DATA
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	COLLECT AGGREGATE ACTIVITY DATA

Tabelle 136. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	Immer erfasst
Statistiken	event_wcstats	Immer erfasst
Statistiken	event_wlstats	Immer erfasst

Verwendung

Mit dieser Statistik können Sie das arithmetische Mittel der Warteschlangenzeit für Koordinatoraktivitäten ermitteln, die einer abgeschlossenen oder abgebrochenen Serviceunterklasse, Workload oder Arbeitsklasse zugeordnet wurden.

Ferner kann mithilfe dieser Statistik festgestellt werden, ob die verwendete Histogrammschablone für das Histogramm der Warteschlangenzeit von Aktivitäten geeignet ist. Mit dem Histogramm der Warteschlangenzeit von Aktivitäten kann die durchschnittliche Warteschlangenzeit von Aktivitäten berechnet werden. Vergleichen Sie den berechneten Durchschnittswert mit diesem Monitorelement. Wenn der berechnete Durchschnittswert vom tatsächlichen Durchschnittswert abweicht, der von diesem Monitorelement zurückgemeldet wird, sollten Sie eine Änderung der Histogrammschablone für das Histogramm der Warteschlangenzeit von Aktivitäten in Erwägung ziehen. Verwenden Sie dazu eine Gruppe von Binwerten, die für Ihre Daten besser geeignet sind.

coord_act_rejected_total - Gesamtzahl zurückgewiesener Koordinatoraktivitäten (Monitorelement)

Das Element 'coord_act_rejected_total' speichert die Gesamtzahl von Koordinatoraktivitäten auf allen Verschachtelungsebenen, die seit dem letzten Zurücksetzen nicht ausgeführt, sondern zurückgewiesen wurden.

Dieser Zähler wird aktualisiert, wenn eine Aktivität durch einen vorbeugenden Schwellenwert oder die Arbeitsaktion zum Verhindern der Ausführung an der Ausführung gehindert wird. Bei Serviceklassen wird dieser Wert beim Beenden von Aktivitäten aktualisiert. Bei Workloads wird der Wert bei Abschluss der zugehörigen UOW mit den einzelnen Workloadvorkommen aktualisiert.

Tabelle 137. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	COLLECT AGGREGATE ACTIVITY DATA
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 138. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wlstats	-

Verwendung

Mit diesem Element kann ermittelt werden, ob vorbeugende Schwellenwerte und Arbeitsaktionen, die die Ausführung verhindern, effektiv eingesetzt werden oder zu einschränkend wirken.

coord_partition_num - Nummer der Koordinatorpartition (Monitorelement)

Die Koordinatorpartition der UOW (Unit of Work) oder Aktivität. In einem Mehrpartitionssystem ist die Koordinatorpartition die Partition, durch die die Anwendung mit der Datenbank verbunden ist.

Tabelle 139. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 140. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity	Immer erfasst
Schwellenwertverstöße	event_thresholdviolations	Immer erfasst

Verwendung

Dieses Element ermöglicht es, bei Aktivitäten oder UOWs, die über Datensätze auf anderen Partitionen als der Koordinatorpartition verfügen, die Koordinatorpartition zu ermitteln.

cost_estimate_top - Maximaler Wert für Kostenschätzung (Monitorelement)

Das Monitorelement 'cost_estimate_top' ist der Höchstwert für die geschätzten Kosten von DML-Aktivitäten auf allen Verschachtelungsebenen innerhalb einer Serviceunterklasse oder Arbeitsklasse.

Bei Serviceunterklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Serviceunterklasse mit NONE definiert ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Arbeitsklasse nicht definiert ist.

Bei Serviceklassen wird die Kostenschätzung für DML-Aktivitäten nur zu der Serviceunterklasse gezählt, in der die Aktivität in das System eintritt. Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, bleibt der Wert von 'cost_estimate_top' der Serviceunterklasse, zu der Sie eine Aktivität zuordnen, unberührt.

Tabelle 141. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wcstats	-
Statistiken	event_wlstats	-

Verwendung

Dieses Element gibt Aufschluss über den höchsten geschätzten Aufwand für DML-Aktivitäten, der in einem Member für eine Serviceklasse, Workload oder Arbeitsklasse erreicht wurde.

cpu_limit - CPU-Begrenzung des WLM-Dispatchers (Monitorelement)

Die CPU-Begrenzung des WLM-Dispatchers, die für die Serviceklasse konfiguriert ist.

Tabelle 142. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_SAMPLE_SERVICE_CLASS_METRICS	ACTIVITY METRICS BASE
- Stichprobe für Messdaten zur Serviceklasse abrufen	

cpu_share_type - CPU-Anteilstyp des WLM-Dispatchers (Monitorelement)

Der Typ der CPU-Anteile des WLM-Dispatchers, die für die Serviceklasse konfiguriert sind. Mögliche Werte sind 'vorläufig' (soft) und 'fest' (hard).

Tabelle 143. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	ACTIVITY METRICS BASE

cpu_shares - CPU-Anteile des WLM-Dispatchers (Monitorelement)

Die Anzahl der CPU-Anteile des WLM-Dispatchers, die für die Serviceklasse konfiguriert sind.

Tabelle 144. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	ACTIVITY METRICS BASE

cpu_utilization - CPU-Auslastung (Monitorelement)

Die gesamte CPU-Zeit, die von der Serviceklasse oder der Auslastung (Workload) auf einer bestimmten logischen Partition verbraucht wurde, dividiert durch die CPU-Zeit, die auf dem Host oder der logischen Partition (LPAR) in einem gegebenen Zeitraum verfügbar war.

Tabelle 145. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	REQUEST METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	REQUEST METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	REQUEST METRICS BASE

Tabelle 146. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE

Verwendung

Bei Rückgabe durch die Funktion `WLM_GET_WORKLOAD_STATS` oder `WLM_GET_SERVICE_SUBCLASS_STATS` stellt dieses Monitorelement die CPU-Auslastung seit der letzten Zurücksetzung der Statistiken dar.

Bei Rückgabe durch die Tabellenfunktion `MON_SAMPLE_SERVICE_CLASS_METRICS` oder `MON_SAMPLE_WORKLOAD_METRICS` stellt dieses Monitorelement die CPU-Auslastung seit der Ausführung der Funktion dar.

`cpu_velocity` - CPU-Geschwindigkeit (Monitorelement)

Ein Wert für die Zugriffskonkurrenz für die CPU-Ressourcen auf einer Skala von 0 bis 1, wobei niedrigere Werte eine höhere Zugriffskonkurrenz bedeuten.

Die CPU-Geschwindigkeit berechnet sich aus dem Wert für die gemessene Zeitdauer, die Operationen in einer Serviceklasse Zugriff auf die CPU haben, dividiert durch die Gesamtzeit, die für den Zugriff auf die CPU oder für das Warten auf den CPU-Zugriff aufgewendet wurde. Es stellt ein Maß für die Effizienz, mit der Operationen, die ausgeführt werden, im Verhältnis zu der Effizienz dar, mit der sie ausgeführt werden könnten, wenn sie nie auf die CPU warten müssten. Es gilt die folgende Formel:

$$\text{cpu_velocity} = \text{total_cpu_time} / (\text{total_cpu_time} + \text{total_disp_run_queue_time})$$

Der Konfigurationsparameter `wlm_dispatcher` des Datenbankmanagers muss auf den Wert `ON` für `cpu_velocity` gesetzt werden, damit eine Erfassung ausgeführt wird.

Tabelle 147. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
<code>MON_SAMPLE_SERVICE_CLASS_METRICS</code> - Stichprobe für Messdaten zur Serviceklasse	REQUEST METRICS BASE
<code>MON_SAMPLE_WORKLOAD_METRICS</code> - Beispielauslastungsmessdaten abrufen	REQUEST METRICS BASE

Verwendung

Der Dispatcher wird bei der Priorisierung einer Serviceklasse oder Workload wirksam, wenn diese Serviceklasse oder Workload zu einem bestimmten Zeitpunkt mehr CPU-Ressourcen anfordern, als bereitgestellt werden können. In diesen Fällen werden die in der Serviceklasse oder der Workload ausgeführten Operationen in eine Warteschlange für den Zugriff auf CPU-Ressourcen eingereiht. Wenn dies auftritt, kann der Dispatcher einer solchen Serviceklasse oder Workload mehr CPU-Ressourcen zur Verfügung stellen, indem die Verteilung von CPU-Ressourcen an ein anderes Objekt verringert wird. Eine hohe CPU-Geschwindigkeit gibt an, dass der Dispatcher wenig Auswirkung auf die Verbesserung der Antwortzeiten oder des Durchsatzes für diese Serviceklasse zu deren aktuellen CPU-Anforderungsgrad haben kann, weil dieser Bedarf bereits befriedigt ist. Eine niedrige CPU-Geschwindigkeit gibt an, dass der Dispatcher möglicherweise eine bedeutende Auswirkung auf die Verbesserung von Antwortzeiten oder des Durchsatzes für diese Serviceklasse oder Workload zu deren aktuellen CPU-Anforderungsgrad haben kann.

Mit diesem Element können Sie ermitteln, ob die in einer Serviceklasse oder Workload ausgeführten Operationen einen relativ großen Teil ihrer Zeit mit dem Warten auf die Verwendung von CPU-Ressourcen verwenden. Wenn die CPU-Ge-

schwindigkeit für eine Serviceklasse niedrig ist und Sie diese erhöhen wollen, können Sie die Steuerung der CPU-Ressourcen durch den WLM-Dispatcher anpassen, indem Sie die Anteile der CPU-Anteil oder den CPU-Grenzwert erhöhen, der der Serviceklasse zugeordnet ist, die eine niedrige CPU-Geschwindigkeit zeigt.

db_work_action_set_id - Set-ID für Datenbankarbeitsaktionen (Monitorelement)

Wurde die Aktivität innerhalb einer Arbeitsklasse im Datenbankbereich kategorisiert, zeigt dieses Monitorelement die ID des Arbeitsaktionssets an, die dem Arbeitsklassenset, zu dem die Arbeitsklasse gehört, zugeordnet ist. Andernfalls zeigt dieses Monitorelement 0 an.

Tabelle 148. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Befehl und Ebene der Monitorelementerfassung
WLM_GET_ACTIVITY_DETAILS_COMPLETE (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen	ACTIVITY METRICS BASE

Tabelle 149. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element kann in Verbindung mit dem Element **db_work_class_id** verwendet werden, um die Datenbankarbeitsklasse der Aktivität (sofern vorhanden) eindeutig zu identifizieren.

db_work_class_id - Datenbankarbeitsklassen-ID (Monitorelement)

Wurde die Aktivität innerhalb einer Arbeitsklasse im Datenbankbereich kategorisiert, zeigt dieses Monitorelement die ID der Arbeitsklasse an. Andernfalls zeigt dieses Monitorelement 0 an.

Tabelle 150. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_ACTIVITY_DETAILS_COMPLETE (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 151. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element kann in Verbindung mit dem Element `db_work_action_set_id` verwendet werden, um die Datenbankarbeitsklasse der Aktivität, soweit vorhanden, eindeutig zu identifizieren.

`destination_service_class_id` - Zielserviceklassen-ID (Monitorelement)

Die ID der Serviceunterklasse, der eine Aktivität neu zugeordnet wurde, als der Datensatz für Schwellenwertverstoß, zu dem das betreffende Element gehört, generiert wurde. Dieses Element hat für alle Schwellenwertaktionen außer REMAP ACTIVITY den Wert null.

Tabelle 152. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Mithilfe dieses Elements kann der Pfad einer Aktivität durch die Serviceklassen, denen sie neu zugeordnet wurde, verfolgt werden. Darüber hinaus kann mit diesem Element zusammenfassend berechnet werden, wie viele Aktivitäten in eine bestimmten Serviceunterklasse hinein zugeordnet wurden.

`estimated_cpu_entitlement` - Geschätztes CPU-Nutzungsrecht (Monitorelement)

Der Prozentsatz von der gesamten CPU-Auslastung auf einem Host oder einer logischen Partition (LPAR), zu dessen Nutzung eine Serviceunterklasse auf der Basis der Konfiguration ihrer CPU-Anteile konfiguriert ist. Dies geht davon aus, dass sie nicht mehr und nicht weniger als die konfigurierte Größe nutzt.

Die Ermittlung, welche Serviceklassen an der zugehörigen Kalkulation beteiligt sind, basiert auf der tatsächlichen CPU-Auslastung, die im Stichprobenzeitraum gemessen wird, gegenüber der Konfigurationseinstellung `WLM_DISP_MIN_UTIL` des Datenbankmanagers. Die Auswirkung eines CPU-Grenzwerts für eine Serviceklasse selbst, für die Serviceklassen, mit denen diese konkurriert, oder für eine übergeordnete Serviceklasse (falls vorhanden) werden bei der Berechnung nicht berücksichtigt.

Tabelle 153. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
<code>MON_SAMPLE_SERVICE_CLASS_METRICS</code> - Stichprobe für Messdaten zur Serviceklasse abrufen	ACTIVITY METRICS BASE

`histogram_type` - Histogrammtyp (Monitorelement)

Der Typ des Histogramms im Zeichenfolgeformat.

Es gibt sieben verschiedene Typen von Histogrammen.

`CoordActQueueTime`

Ein Histogramm der im Koordinatormember gemessenen Zeit (in Millise-

kunden), die nicht verschachtelte Aktivitäten in einer Warteschlange (z. B. einer Schwellenwertwarteschlange) verbrachten.

CoordActExecTime

Ein Histogramm der Ausführungszeit (in Millisekunden) nicht verschachtelter Aktivitäten im Koordinatormember. Die Ausführungszeit umfasst nicht die für die Initialisierung benötigte bzw. die in Warteschlangen verbrachte Zeit. Bei Cursors umfasst die Ausführungszeit nur die Zeit für das Öffnen, Abrufen und Schließen von Anforderungen. Wenn eine Aktivität zwischen Serviceunterklassen neu zugeordnet wird, wird das Histogramm der Ausführungszeit nur für die Serviceunterklasse aktualisiert, in der die Ausführung der Aktivität abgeschlossen wird.

CoordActLifetime

Ein Histogramm der im Koordinatormember gemessenen abgelaufenen Zeit (in Millisekunden) zwischen dem Zeitpunkt, zu dem eine nicht verschachtelte Aktivität vom Datenbankmanager erkannt wird, bis zu dem Zeitpunkt, zu dem die Ausführung der Aktivität abgeschlossen wird. Wenn Aktivitäten zwischen Serviceunterklassen neu zugeordnet werden, wird das Histogramm der Laufzeit nur für die Serviceunterklasse aktualisiert, in der die Ausführung der Aktivität abgeschlossen wird.

CoordActInterArrivalTime

Ein Histogramm des Zeitintervalls (in Millisekunden) zwischen der Ankunft nicht verschachtelter Koordinatoraktivitäten. Die durchschnittliche Zwischenankunftszeit (Inter-Arrival Time) wird für Serviceunterklassen berechnet, über die Aktivitäten in das System eintreten. Wenn Sie Aktivitäten zwischen Serviceunterklassen neu zuordnen, bleibt das Histogramm der Zwischenankunftszeit der Serviceunterklasse, zu der Sie eine Aktivität zuordnen, unberührt.

CoordActEstCost

Ein Histogramm der geschätzten Kosten (in Timeron) nicht verschachtelter DML-Aktivitäten. Die Kostenschätzung einer Aktivität wird nur zu der Serviceunterklasse gezählt, in der die Aktivität in das System eintritt.

ReqExecTime

Ein Histogramm der Ausführungszeit von Anforderungen (in Millisekunden), einschließlich der Anforderungen im Koordinatormember, sowie aller Unteranforderungen sowohl in Koordinator- als auch in Nicht-Koordinatormitgliedern (wie RPC-Anforderungen oder Anforderungen von SMP-Subagenten). Den enthaltenen Anforderungen kann eine Aktivität zugeordnet sein: So sind beispielsweise sowohl PREPARE- als auch OPEN-Anforderungen in diesem Histogramm enthalten, doch während den OPEN-Anforderungen stets eine Cursoraktivität zugeordnet ist, sind PREPARE-Anforderungen keine Aktivitäten zugeordnet. Das Histogramm der Ausführungszeit einer an der Neuordnung beteiligten Serviceunterklasse zählt den Teil der Ausführungszeit, den die partielle Anforderung in der Serviceunterklasse verbringt.

UowLifetime

Ein Histogramm der verstrichenen Zeit (in Millisekunden) von dem Zeitpunkt an, zu dem eine UOW (Unit of Work) vom Datenbankmanager erkannt wird, bis zu dem Zeitpunkt, zu dem die UOW die Ausführung beendet (festgeschrieben oder rückgängig gemacht).

Tabelle 154. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_histogrambin	Immer erfasst

Verwendung

Mithilfe dieses Elements kann der Typ eines Histogramms bestimmt werden. Zu einem Statistikdatensatz können mehrere Histogramme gehören, jedoch jeweils nur ein Histogramm eines bestimmten Typs.

last_wlm_reset - Zeitpunkt des letzten Zurücksetzens (Monitorelement)

Dieses Element zeigt in Form einer lokalen Zeitmarke den Zeitpunkt an, zu dem der letzte Ereignisdatensatz für Statistikdaten diesen Typs erstellt wurde.

Tabelle 155. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wlstats	-
Statistiken	event_wcstats	-
Statistiken	event_qstats	-

Verwendung

Mit den Monitorelementen **wlm_last_reset** und **statistics_timestamp** lässt sich der Zeitraum ermitteln, in dem die Statistikdaten im Statistikdatensatz eines Ereignismonitors erfasst wurden. Das Erfassungsintervall beginnt mit der Zeit von **wlm_last_reset** und endet mit **statistics_timestamp**.

num_remaps - Anzahl der Neuordnungen (Monitorelement)

Ein Zähler für die Häufigkeit, mit der die betreffende Aktivität neu zugeordnet wurde. Wenn das Monitorelement 'num_remaps' einen Wert größer null hat, enthält das Monitorelement 'service_class_id' dieses Aktivitätsdatensatzes die Kennung (ID) der letzten Serviceklasse, der diese Aktivität zugeordnet wurde.

Tabelle 156. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Mithilfe dieser Information können Sie überprüfen, ob die Aktivität mit der erwarteten Häufigkeit neu zugeordnet wurde.

num_threshold_violations - Anzahl der Schwellenwertverstöße (Monitorelement)

Die Anzahl der Schwellenwertverstöße, die in der Datenbank seit der letzten Aktivierung aufgetreten sind.

Dieses Monitorelement ist ein Alias des Monitorelements „thresh_violations - Anzahl der Schwellenwertverstöße (Monitorelement)“ auf Seite 489, das von einigen Tabellenfunktionen (MON_*) zurückgegeben wird.

Tabelle 157. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Datenbank	dbase	Einfach

Bei der Momentaufnahmeüberwachung kann dieser Zähler zurückgesetzt werden.

Tabelle 158. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Datenbank	event_db	Immer erfasst

Verwendung

Mit diesem Element lässt sich feststellen, ob Schwellenwerte für eine bestimmte Anwendung sinnvoll definiert sind und ob zu viele Schwellenwertverstöße auftreten.

number_in_bin - Anzahl in Bin (Monitorelement)

Dieses Element enthält den Zähler für die Anzahl der Aktivitäten bzw. Anforderungen, die in den von dem Histogrammbin erfassten Zeitraum fallen.

Tabelle 159. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_histogrambin	-

Verwendung

Dieses Element gibt Aufschluss über die Höhe eines Bins in dem Histogramm.

parent_activity_id - ID der übergeordneten Aktivität (Monitorelement)

Die eindeutige Kennung (ID) der übergeordneten Aktivität der Aktivität innerhalb der UOW (Unit of Work). Ist keine übergeordnete Aktivität vorhanden, ist der Wert dieses Monitorelements 0.

Tabelle 160. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Befehl und Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 161. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Mit diesem Element lässt sich in Verbindung mit den Elementen **parent_uow_id** und **appl_id** die übergeordnete Aktivität der im Aktivitätsdatensatz beschriebenen Aktivität eindeutig identifizieren.

parent_uow_id - UOW-ID der übergeordneten Aktivität (Monitorelement)

Die innerhalb einer Anwendungskennung eindeutige ID der UOW (Unit of Work). Die ID der UOW, in der die übergeordnete Aktivität der Aktivität ihren Ursprung hat. Ist keine übergeordnete Aktivität vorhanden, ist der Wert 0.

Tabelle 162. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Befehl und Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 163. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Mit diesem Element lässt sich in Verbindung mit den Elementen **parent_activity_id** und **appl_id** die übergeordnete Aktivität der im Aktivitätsdatensatz beschriebenen Aktivität eindeutig identifizieren.

prep_time - Vorbereitungszeit (Monitorelement)

Die für die Vorbereitung einer SQL-Anweisung erforderliche Zeit (in Millisekunden); (falls es sich bei der Aktivität um eine SQL-Anweisung handelt, andernfalls ist der Wert 0).

Tabelle 164. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE

Tabelle 164. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_PKG_CACHE_STMT_DETAILS (Tabellenfunktion) - Detaillierte Messdaten für Paketcacheinträge abrufen	ACTIVITY METRICS BASE

Tabelle 165. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst
Paketcache	-	COLLECT BASE DATA

Verwendung

Das Monitorelement 'prep_time' gibt an, wie viel Zeit für die Vorbereitung einer SQL-Anweisung benötigt wurde (falls es sich bei dieser Aktivität um eine SQL-Anweisung handelte), als die Anweisung zum ersten Mal in den DB2-Paketcache eingefügt wurde. Diese Vorbereitungszeit ist nicht Teil der Aktivitätslaufzeit; sie stellt auch nicht den Zeitraum dar, der während eines bestimmten Aufrufs der Anweisung vergangen ist, falls sich die Anweisung vor diesem Aufruf bereits im Paketcache befand.

queue_assignments_total - Gesamtzahl der Warteschlangenzuordnungen (Monitorelement)

Gibt an, wie häufig eine Verbindung bzw. Aktivität der Schwellenwertwarteschlange seit dem letzten Zurücksetzen zugeordnet wurde.

Tabelle 166. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Tabelle 167. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	-

Verwendung

Dieses Element gibt Aufschluss darüber, wie häufig eine Verbindung bzw. Aktivität innerhalb eines bestimmten Zeitraums, der durch das Intervall für die Erfassung von Statistikdaten vorgegeben wird, in diese Warteschlange eingereiht wurde. Dieses Element ist hilfreich bei der Beurteilung der Effizienz von Schwellenwerten für die Warteschlangensteuerung.

queue_size_top - Maximaler Wert für Warteschlangenlänge (Monitorelement)

Der Höchstwert für die Warteschlangenlänge, der seit dem letzten Zurücksetzen erreicht wurde.

Table 168. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Table 169. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	-

Verwendung

Dieses Element gibt Aufschluss über die Effizienz von Schwellenwerten für die Warteschlangensteuerung und ermöglicht es zu erkennen, ob Warteschlangen zu umfangreich werden.

queue_time_total - Gesamtwarteschlangenzeit (Monitorelement)

Die Zeit, die alle Verbindungen bzw. Aktivitäten, die seit dem letzten Zurücksetzen in die Warteschlange gestellt wurden, insgesamt in der Warteschlange verbraucht haben. Die Zeit wird in Millisekunden gemessen.

Table 170. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Table 171. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	Immer erfasst

Dieses Element gibt Aufschluss über die Effizienz von Schwellenwerten für die Warteschlangensteuerung und ermöglicht es zu erkennen, ob Warteschlangen zu umfangreich werden.

Hinweise zur Verwendung

Das Monitorelement **queue_time_total** wird am Ende eines Statistikerfassungsintervalls nicht zurückgesetzt. Falls das Element **queue_time_total** über mehrere Intervalle hinweg verwendet wird, kann es sein, dass sein Wert größer als das Produkt aus **wlm_collect_int** und **queue_size_top** ist.

request_exec_time_avg - Durchschnittswert für die Ausführungszeit von Anforderungen (Monitorelement)

Das arithmetische Mittel der Ausführungszeiten von Anforderungen, die dieser Serviceunterklasse seit dem letzten Zurücksetzen zugeordnet wurden. Wenn für den intern aufgezeichneten Durchschnittswert ein Überlauf stattgefunden hat, wird der Wert -2 zurückgegeben. Dieses Monitorelement gibt -1 zurück, wenn COLL-

ECT AGGREGATE REQUEST DATA für die Serviceunterklasse auf NONE gesetzt ist. Die Zeit wird in Millisekunden gemessen.

Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, zählt der Durchschnittswert von 'request_act_exec_time_avg' die partielle Anforderung in jeder an der Neuordnung beteiligten Serviceunterklasse.

Tabelle 172. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	COLLECT AGGREGATE REQUEST DATA

Tabelle 173. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-

Verwendung

Mithilfe dieser Statistik können Sie schnell die durchschnittliche Zeitdauer für die Verarbeitung der einzelnen Anforderungen in einem Member in dieser Serviceunterklasse ermitteln.

Ferner kann mithilfe dieses Durchschnittswerts ermittelt werden, ob die verwendete Histogrammschablone für das Histogramm der Ausführungszeit von Anforderungen geeignet ist. Mit dem Histogramm der Ausführungszeit von Anforderungen kann die durchschnittliche Ausführungszeit von Anforderungen berechnet werden. Vergleichen Sie den berechneten Durchschnittswert mit diesem Monitorelement. Wenn der berechnete Durchschnittswert vom tatsächlichen Durchschnittswert abweicht, der von diesem Monitorelement zurückgemeldet wird, sollten Sie eine Änderung der Histogrammschablone für das Histogramm der Ausführungszeit von Anforderungen in Erwägung ziehen. Verwenden Sie dazu eine Gruppe von Binwerten, die für Ihre Daten besser geeignet sind.

routine_id - Routinen-ID (Monitorelement)

Eine eindeutige Routinenkennung. Dieses Monitorelement gibt 0 zurück, wenn die Aktivität nicht zu einer Routine gehört.

Tabelle 174. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT (Tabellenfunktion)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS - Detaillierte Messdaten für Paketcacheinträge abrufen	ACTIVITY METRICS BASE

Tabelle 174. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ROUTINE (Tabellenfunktion) Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusammengefasste Ausführungsmessdaten abrufen	Immer erfasst
MON_GET_ROUTINE_EXEC_LIST (Tabellenfunktion) - Liste der von der Routine ausgeführten Anweisungen abrufen	Immer erfasst
MON_GET_SECTION_ROUTINE (Tabellenfunktion) - Liste von Routinen für Eingabeabschnitt abrufen	Immer erfasst
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 175. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activitystmt	Immer erfasst
UOW (Unit of Work)	uow_package_list	Immer erfasst
Paketcache	pkgcache_metrics	ACTIVITY METRICS BASE

Verwendung

Der Wert dieses Elements entspricht einem Wert in der Spalte ROUTINEID der Sicht SYSCAT.ROUTINES. Wenn die Aktivität Teil einer SQL-PL-Routine ist, die Sie in einer anderen SQL-PL-Routine deklarieren, entspricht der Wert dieses Elements der ROUTINEID der äußeren Routine.

rows_fetched - Abgerufene Zeilen (Monitorelement)

Die Anzahl der aus der Tabelle gelesenen Zeilen.

Dieses Monitorelement ist ein Alias des Monitorelements **rows_read**.

Anmerkung: Dieses Monitorelement listet nur die Werte für das Member auf, für das diese Informationen aufgezeichnet werden. In Datenbankumgebungen mit mehreren Members geben diese Werte möglicherweise nicht die korrekten Summen für die gesamte Aktivität wieder.

Tabelle 176. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Anweisung

Verwendung

Nähere Informationen können Sie dem Abschnitt zum Monitorelement **rows_read** entnehmen.

rows_modified - Modifizierte Zeilen (Monitorelement)

Die Anzahl der eingefügten, aktualisierten und gelöschten Zeilen.

Dieses Monitorelement ist ein Alias des Monitorelements **rows_written**.

Tabelle 177. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung(Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_XML_METRICS_BY_ROW - Formatierte zeilenbasierte Ausgabe für alle Messwerte abrufen	Nicht zutreffend; meldet alle Elemente zu- rück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_CONNECTION (Tabellenfunktion) - Verbindungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS (Tabellenfunktion) - Detaillierte Verbindungsmessdaten abrufen (im XML- Dokument in der Spalte DETAILS zurückge- meldet)	REQUEST METRICS BASE
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS (Tabellenfunktion) - Detaillierte Messdaten für Paketcacheeinträge abrufen	ACTIVITY METRICS BASE
MON_GET_ROUTINE (Tabellenfunktion) - Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusamme- gefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zu- rückgemeldet)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	REQUEST METRICS BASE

Table 177. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE

Table 178. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE
Aktivitäten	event_activity	Anweisung
Paketcache	Im Dokument 'activity_metrics' zurückgemeldet.	ACTIVITY METRICS BASE

Verwendung

Nähere Informationen können Sie dem Abschnitt zum Monitorelement **rows_written** entnehmen.

rows_returned - Zurückgegebene Zeilen (Monitorelement)

Das Monitorelement 'rows_returned' gibt die Anzahl der Zeilen an, die ausgewählt und an die Anwendung zurückgegeben wurden.

Dieses Element weist für partielle Aktivitätsdatensätze den Wert 0 auf (z. B. wenn eine Aktivität noch während der Ausführung erfasst wird oder ein vollständiger Aktivitätsdatensatz aufgrund von Speicherbegrenzungen nicht in den Ereignismonitor geschrieben werden konnte).

Dieses Monitorelement ist ein Alias des Monitorelements **fetch_count**.

Table 179. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung(Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_XML_METRICS_BY_ROW - Formatierte zeilenbasierte Ausgabe für alle Messwerte abrufen	Nicht zutreffend; meldet alle Elemente zu- rück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_CONNECTION (Tabellenfunktion) - Verbindungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS (Tabellenfunktion) - Detaillierte Verbindungsmessdaten abrufen (im XML- Dokument in der Spalte DETAILS zurückge- meldet)	REQUEST METRICS BASE
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS (Tabellenfunktion) - Detaillierte Messdaten für Paketcacheeinträge abrufen	ACTIVITY METRICS BASE
MON_GET_ROUTINE (Tabellenfunktion) - Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusamme- gefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zu- rückgemeldet)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Mess- daten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	REQUEST METRICS BASE

Tabelle 179. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	REQUEST METRICS BASE

Tabelle 180. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE
Aktivitäten	event_activity	Immer erfasst
Paketcache	Im Dokument 'activity_metrics' zurückgemeldet.	ACTIVITY METRICS BASE

Verwendung

Mit diesem Element können die Schwellenwerte für Zeilen ermittelt werden, die an die Anwendung zurückgegeben werden. Darüber hinaus gibt dieses Element Aufschluss darüber, ob ein jeweiliger Schwellenwert korrekt konfiguriert ist und seinen Zweck erfüllt.

rows_returned_top - Maximaler Wert für zurückgegebene Zeilen (Monitorelement)

Das Monitorelement 'rows_returned_top' ist der Höchstwert für die für DML-Aktivitäten auf allen Verschachtelungsebenen innerhalb einer Service- oder Arbeitsklasse zurückgegebenen Zeilen.

Bei Serviceklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Serviceklasse mit NONE definiert ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE gesetzt ist.

Bei Serviceklassen gilt: Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, wird nur der Höchstwert von 'rows_returned_top' der Serviceunterklasse aktualisiert, in der eine Aktivität abgeschlossen wird. Die Höchstwerte von Serviceunterklassen, zu denen eine Aktivität zwar zugeordnet wird, in denen sie jedoch nicht abgeschlossen wird, bleiben unberührt.

Tabelle 181. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wcstats	-
Statistiken	event_wlstats	-

Verwendung

Mithilfe dieses Elements können Sie festlegen, wie viele Zeilen maximal für eine Aktivität zurückgegeben werden, die in einem Member für ein bestimmte Serviceklasse ausgeführt wird.

sc_work_action_set_id - Set-ID für Serviceklassenarbeitsaktionen (Monitorelement)

Wurde die Aktivität innerhalb einer Arbeitsklasse im Serviceklassenbereich kategorisiert, zeigt dieses Monitorelement die ID des Arbeitsaktionssets an, die dem Arbeitsklassenset, zu dem die Arbeitsklasse gehört, zugeordnet ist. Andernfalls zeigt dieses Monitorelement 0 an.

Tabelle 182. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Befehl und Ebene der Monitorelementerfassung
WLM_GET_ACTIVITY_DETAILS_COMPLETE (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 183. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element kann in Verbindung mit dem Element **sc_work_class_id** verwendet werden, um die Serviceklassenarbeitsklasse der Aktivität (sofern vorhanden) eindeutig zu identifizieren.

sc_work_class_id - Arbeitsklassen-ID für Serviceklassen (Monitorelement)

Wurde die Aktivität innerhalb einer Arbeitsklasse im Serviceklassenbereich kategorisiert, zeigt dieses Monitorelement die ID der Arbeitsklasse an, die dieser Aktivität zugeordnet ist. Andernfalls zeigt dieses Monitorelement 0 an.

Tabelle 184. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Befehl und Ebene der Monitorelementerfassung
WLM_GET_ACTIVITY_DETAILS_COMPLETE (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 185. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element kann in Verbindung mit dem Element `sc_work_action_set_id` verwendet werden, um die Serviceklassenarbeitsklasse der Aktivität (sofern vorhanden), eindeutig zu identifizieren.

section_env - Abschnittsumgebung (Monitorelement)

Ein BLOB-Objekt, das den Abschnitt für eine SQL-Anweisung enthält. Dabei handelt es sich um den Inhalt des tatsächlichen Abschnitts, der die ausführbare Form eines Abfrageplans darstellt.

Tabelle 186. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activitystmt	Immer erfasst
Paketcache	pkgcache	COLLECT DETAILED DATA

Verwendung

Verwenden Sie dieses Element zusammen mit den EXPLAIN-Prozeduren für Abschnitte, um eine Anweisung mit EXPLAIN zu bearbeiten und den Zugriffsplan für die Anweisung anzuzeigen.

service_class_id - Serviceklassen-ID (Monitorelement)

Die Eindeutige Kennung (ID) einer Serviceunterklasse. Für eine UOW (Unit of Work) stellt diese Kennung die Serviceunterklassen-ID der Workload dar, der die Verbindung, die die UOW aufruft, zugeordnet ist.

Tabelle 187. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung(Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	ACTIVITY METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen	ACTIVITY METRICS BASE

Tabelle 187. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung(Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 188. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Sperren	-	Immer erfasst
UOW (Unit of Work)	-	Immer erfasst
Statistiken	event_histogrambin	Immer erfasst
Statistiken	event_scstats	Immer erfasst

Verwendung

Der Wert dieses Elements entspricht einem Wert in der Spalte SERVICECLASSID der Sicht SYSCAT.SERVICECLASSES. Mithilfe dieses Elements können Sie den Namen einer Serviceunterklasse ermitteln oder Informationen zu einer Serviceunterklasse aus verschiedenen Quellen verbinden. Zum Beispiel können Sie Statistikdaten von Serviceklassen mit Histogrammbinsätzen kombinieren.

Der Wert dieses Elements ist 0, wenn die folgenden Bedingungen zutreffen:

- Das Element wird in einer logischen Datengruppe 'event_histogrambin' zurückgegeben.
- Die Histogrammdata werden für ein Objekt erfasst, das keine Serviceklasse ist.

service_subclass_name - Name der Serviceunterklasse (Monitorelement)

Der Name einer Serviceunterklasse.

Tabelle 189. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	ACTIVITY METRICS BASE

Tabelle 189. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	ACTIVITY METRICS BASE
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE

Tabelle 190. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'details_xml' zurückgemeldet)	REQUEST METRICS BASE
Sperren	-	Immer erfasst
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity	Immer erfasst
Statistiken	event_scstats	Immer erfasst
Statistiken	event_qstats	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Aktivitätselementen eine Analyse des Verhaltens einer Aktivität und in Verbindung mit anderen Statistikelementen eine Analyse einer Serviceklasse oder Schwellenwertwarteschlange.

service_superclass_name - Name der Servicesuperklasse (Monitorelement)

Der Name einer Servicesuperklasse.

Tabelle 191. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	ACTIVITY METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zu- rückgemeldet)	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Mess- daten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	ACTIVITY METRICS BASE
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_WORKLOAD_ OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	ACTIVITY METRICS BASE
WLM_GET_SERVICE_SUPERCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Servicesuperklassen zurückgeben	ACTIVITY METRICS BASE

Tabelle 192. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'details_xml' zurückgemeldet)	REQUEST METRICS BASE
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity	Immer erfasst
Statistiken	event_scstats	Immer erfasst
Statistiken	event_qstats	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Aktivitätselementen eine Analyse des Verhaltens einer Aktivität und in Verbindung mit anderen Statistikelementen eine Analyse einer Serviceklasse oder Schwellenwertwarteschlange.

source_service_class_id - Quellenserviceklassen-ID (Monitorelement)

Die ID der Serviceunterklasse, aus der eine Aktivität neu zugeordnet wurde, als der Datensatz für Schwellenwertverstoß, zu dem das betreffende Element gehört, generiert wurde. Dieses Element hat für alle Schwellenwertaktionen außer REMAP ACTIVITY den Wert null.

Tabelle 193. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Mithilfe dieses Elements kann der Pfad einer Aktivität durch die Serviceklassen, denen sie neu zugeordnet wurde, verfolgt werden. Darüber hinaus kann mit diesem Element zusammenfassend berechnet werden, wie viele Aktivitäten aus einer bestimmten Serviceunterklasse heraus zugeordnet wurden.

statistics_timestamp - Zeitmarke für Statistikdatensatz (Monitorelement)

Der Zeitpunkt, zu dem der Statistikdatensatz generiert wurde.

Tabelle 194. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wlstats	-
Statistiken	event_wcstats	-
Statistiken	event_qstats	-
Statistiken	event_histogrambin	-

Verwendung

Mithilfe dieses Elements kann ermittelt werden, wann der Statistikdatensatz erstellt wurde.

Verwenden Sie dieses Element in Verbindung mit dem Element **last_wlm_reset**, um das Zeitintervall zu bestimmen, in dem die Statistikdaten in diesem Statistikdatensatz generiert wurden.

Mithilfe dieses Monitorelements können darüber hinaus auch alle Statistikdatensätze zusammengefasst werden, die in demselben Erfassungsintervall generiert wurden.

stmt_invocation_id - Aufruf-ID der Anweisung (Monitorelement)

Eine Kennung (ID), die einen Aufruf einer Routine von anderen auf derselben Verschachtelungsebene innerhalb einer UOW (Unit of Work) unterscheidet. Sie ist innerhalb einer UOW für eine bestimmte Verschachtelungsebene eindeutig.

Tabelle 195. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 196. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activitystmt	-
Sperren	-	-
Deadlocks mit Detailprotokollwerten ¹	event_stmt_history	-
Deadlocks mit Detailprotokoll ¹	event_stmt_history	-
UOW (Unit of Work)	In der Paketliste zurückgemeldet.	-

- 1 Diese Option wird nicht weiter unterstützt. Von einer Verwendung dieser Option, die in zukünftigen Releases möglicherweise nicht mehr enthalten sein wird, wird deshalb abgeraten. Verwenden Sie die Anweisung CREATE EVENT MONITOR FOR LOCKING, um Ereignisse zu überwachen, die im Zusammenhang mit Sperren stehen, wie beispielsweise Überschreitungen von Sperrzeiten, Wartestatus für Sperren und Deadlocks.

Verwendung

Mithilfe dieses Elements kann der Aufruf eindeutig ermittelt werden, in dem eine bestimmte SQL-Anweisung ausgeführt wurde. Darüber hinaus lässt sich mit diesem Element in Verbindung mit anderen Einträgen des Anweisungsprotokolls die Reihenfolge von SQL-Anweisungen ermitteln, die zu dem Deadlock geführt haben.

temp_tablespace_top - Maximaler Wert für Tabellenbereiche für temporäre Tabellen (Monitorelement)

Das Monitorelement 'temp_tablespace_top' ist der Höchstwert (in KB) für die Belegung der Tabellenbereiche für temporäre Tabellen durch DML-Aktivitäten auf allen Verschachtelungsebenen innerhalb einer Service- oder Arbeitsklasse.

Bei Serviceklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Serviceklasse mit NONE definiert ist. Bei Arbeitsklassen gibt dieses Monitorelement -1 zurück, wenn die Arbeitsaktion COLLECT AGGREGATE ACTIVITY DATA für die Arbeitsklasse nicht definiert ist. Bei Workloads gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Workload auf NONE gesetzt ist.

Bei Serviceklassen gilt: Wenn Sie Aktivitäten zwischen Serviceunterklassen mit einer Aktion REMAP ACTIVITY neu zuordnen, wird nur der Höchstwert von 'temp_tablespace_top' der Serviceunterklasse geändert, in der eine Aktivität abgeschlossen wird. Die Höchstwerte von Serviceunterklassen, zu denen eine Aktivität zwar zugeordnet wird, in denen sie jedoch nicht abgeschlossen wird, bleiben unberührt.

Tabelle 197. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	-
Statistiken	event_wcstats	-
Statistiken	event_wlstats	-

Verwendung

Dieses Element gibt Aufschluss über die höchste Nutzung von Tabellenbereich für temporäre Systemtabellen für DML-Aktivitäten, die in einem Member für eine Serviceklasse, Workload oder Arbeitsklasse erreicht wurde.

Dieses Element wird nur von Aktivitäten aktualisiert, auf die ein Schwellenwert für den Speicherplatz von Tabellenbereichen für temporäre Tabellen angewendet wird. Wird kein Schwellenwert für den Speicherplatz von Tabellenbereichen für temporäre Tabellen auf eine Aktivität angewendet, wird der Wert 0 zurückgegeben.

thresh_violations - Anzahl der Schwellenwertverstöße (Monitorelement)

Die Anzahl, wie häufig ein Schwellenwert nicht eingehalten wurde.

Dieses Monitorelement ist ein Alias des Monitorelements „num_threshold_violations - Anzahl der Schwellenwertverstöße (Monitorelement)“ auf Seite 471, das von Routinen der Momentaufnahmeüberwachung und dem Datenbankereignismonitor zurückgegeben wird.

Tabelle 198. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_XML_METRICS_BY_ROW - Formatierte zeilenbasierte Ausgabe für alle Messwerte abrufen	Nicht zutreffend; meldet alle Elemente zurück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_CONNECTION (Tabellenfunktion) - Verbindungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS (Tabellenfunktion) - Detaillierte Verbindungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE

Tabelle 198. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS (Tabellenfunktion) - Detaillierte Messdaten für Paketcacheinträge abrufen	ACTIVITY METRICS BASE
MON_GET_ROUTINE (Tabellenfunktion) - Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE

Tabelle 199. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'details_xml' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Paketcache	Im Dokument 'activity_metrics' zurückgemeldet.	ACTIVITY METRICS BASE

Tabelle 199. Informationen zur Ereignisüberwachung (Forts.)

Ereignistyp	Logische Datengruppierung	Monitorschalter
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE

Verwendung

Mithilfe dieses Elements können Sie rasch ermitteln, ob Verstöße gegen WLM-Schwellenwerte aufgetreten sind. Wenn es Verstöße gegen Schwellenwerte gegeben hat, können Sie mithilfe des Ereignismonitors für Schwellenwertverstöße (sofern erstellt und aktiv) Details zu den Schwellenwertverstößen abrufen.

Sie können zum Beispiel feststellen, gegen welchen Schwellenwert verstoßen wurde.

threshold_action - Schwellenwertaktion (Monitorelement)

Die Aktion für den Schwellenwert, auf den sich dieser Datensatz für Schwellenwertverstöße bezieht. Folgende Werte sind möglich: Stoppen (Stop), Fortsetzen (Continue) oder Neu Zuordnen (Remap).

Tabelle 200. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Mit diesem Element kann festgestellt werden, ob eine Aktivität, die gegen den Schwellenwert verstoßen hat, bei Auftreten des Verstoßes gestoppt wurde, oder ob die Ausführung der Aktivität fortgesetzt werden konnte, oder ob die Aktivität einer anderen Serviceunterklasse zugeordnet wurde. Wenn die Aktivität gestoppt wurde, erhält die Anwendung, die die Aktivität übergeben hat, einen SQL-Fehler (SQL4712N). Wenn die Aktivität einer anderen Serviceunterklasse zugeordnet wurde, werden Agenten, die für die Aktivität im Member arbeiten, in die Zielserviceunterklasse des Schwellenwerts verschoben.

threshold_domain - Schwellenwertdomäne (Monitorelement)

Die Domäne des für diese Warteschlange verantwortlichen Schwellenwerts.

Mögliche Werte:

- Datenbank
- Arbeitsaktionsset
- Servicesuperklasse
- Serviceunterklasse
- Auslastung

Tabelle 201. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Tabelle 202. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	-

Verwendung

Mit diesem Element lassen sich die Warteschlangenstatistiken für Schwellenwerte voneinander abgrenzen, die dasselbe Vergleichselement, aber unterschiedliche Domänen aufweisen.

threshold_maxvalue - Maximaler Wert für Schwellenwert (Monitorelement)

Bei Schwellenwerten, die sich nicht auf Warteschlangen beziehen, stellt dieses Monitorelement den Wert dar, dessen Überschreitung den Schwellenwertverstoß verursacht hat. Bei Schwellenwerten für die Warteschlangensteuerung gibt dieses Monitorelement die Ebene des gemeinsamen Zugriffs an, die den Schwellenwertverstoß verursacht hat.

Die Ebene des gemeinsamen Zugriffs, die den Verstoß gegen den Schwellenwert für die Warteschlangensteuerung verursacht hat, ergibt sich aus der Summe der Monitorelemente **threshold_maxvalue** und **threshold_queuesize**.

Tabelle 203. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	Immer erfasst

Verwendung

Bei Aktivitätsschwellenwerten liefert dieses Element einen Protokolldatensatz, der den maximalen Wert für den Schwellenwert zum Zeitpunkt des Schwellenwertverstoßes angibt. Dies ist hilfreich, wenn sich der maximale Wert für den Schwellenwert seit dem Schwellenwertverstoß geändert hat und der alte Wert in der Sicht SYSCAT.THRESHOLDS nicht mehr angezeigt wird. Für die Schwellenwerte DATATAGINSC IN und DATATAGINSC NOT IN enthält dieses Element den Wert des Datentags, das den Schwellenwert nicht eingehalten hat.

threshold_name - Name des Schwellenwerts (Monitorelement)

Der eindeutige Name des für diese Warteschlange verantwortlichen Schwellenwerts.

Tabelle 204. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Tabelle 205. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	-

Verwendung

Mit diesem Element lässt sich der Schwellenwert für Warteschlangensteuerung, auf den sich die Statistikdaten dieses Datensatzes beziehen, eindeutig ermitteln.

threshold_predicate - Vergleichselement für Schwellenwert (Monitorelement)

Gibt den Typ des Schwellenwerts an, gegen den verstoßen bzw. für den Statistikdaten erfasst wurden.

Tabelle 206. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Tabelle 207. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	Immer erfasst
Statistiken	event_qstats	Immer erfasst

Verwendung

Dieses Monitorelement ermöglicht in Verbindung mit anderen Monitorelementen für Statistiken oder Schwellenwertverstöße eine Analyse des Schwellenwertverstoßes.

Die folgenden Werte sind für dieses Monitorelement gültig, wenn es in der logischen Datengruppe 'event_thresholdviolations' zurückgegeben wird:

- AggSQLTempSpace
- SQLTempSpace
- SQLRowsReturned
- ActivityTotalTime
- EstimatedSQLCost
- TotalMemberConnections
- ConnectionIdleTime
- ConcurrentWorkloadOccurrences
- ConcurrentWorkloadActivities
- ConcurrentDBCoordActivities
- TotalSCMemberConnections
- SQLRowsRead
- SQLRowsReadInSC
- CPUTime

CPUTimeInSC
 UowTotalTime
 DataTagInSC
 DataTagNotInSC

Die folgenden Werte sind für dieses Monitorelement gültig, wenn es in der logischen Datengruppe 'event_qstats' zurückgegeben wird:

TotalMemberConnections
 ConcurrentDBCoordActivities
 TotalSCMemberConnections

threshold_queuesize - Größe der Schwellenwertwarteschlange (Monitorelement)

Die Größe der Warteschlange für einen Schwellenwert für die Warteschlangensteuerung. Ein Versuch, diese Größe zu überschreiten, führt zu einem Schwellenwertverstoß. Für Schwellenwerte, die nicht auf Warteschlangen bezogen sind, hat dieses Monitorelement den Wert 0.

Tabelle 208. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Mit diesem Element lässt sich die Anzahl der Aktivitäten bzw. Verbindungen in der Warteschlange für diesen Schwellenwert bestimmen, die sich zum Zeitpunkt des Schwellenwertverstoßes in der Warteschlange befanden.

thresholdid - Schwellenwert-ID (Monitorelement)

Gibt den Schwellenwert an, auf den sich ein Datensatz für Schwellenwertverstöße bezieht oder für den Warteschlangenstatistikdaten erfasst wurden.

Tabelle 209. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE

Tabelle 210. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-
Statistiken	event_qstats	-

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse einer Warteschlange mit Schwellenwertsteuerung bzw. der Aktivität, die gegen einen Schwellenwert verstoßen hat.

time_completed - Abschlusszeitpunkt (Monitorelement)

Der Zeitpunkt, zu dem die Ausführung der von dem Aktivitätsdatensatz beschriebenen Aktivität abgeschlossen wurde. Bei diesem Element handelt es sich um eine lokale Zeitmarke.

Tabelle 211. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse des Verhaltens einer Aktivität.

Dieses Feld weist den Wert 0000-00-00-00.00.00.000000 auf, wenn aufgrund von Speicherbegrenzungen kein vollständiger Aktivitätsdatensatz in einen Tabellenergebnismonitor geschrieben werden konnte. Wenn die Aktivität erfasst wurde, während sie sich in Bearbeitung befand, stellt dieses Feld den Zeitpunkt dar, an dem die Aktivität erfasst wurde.

time_created - Erstellungszeitpunkt (Monitorelement)

Der Zeitpunkt, zu dem die von dem Aktivitätsdatensatz beschriebene Aktivität von einem Benutzer übergeben wurde. Bei diesem Element handelt es sich um eine lokale Zeitmarke.

Tabelle 212. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	-

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse des Verhaltens einer Aktivität.

time_of_violation - Zeitpunkt des Verstoßes (Monitorelement)

Der Zeitpunkt, zu dem der in diesem Datensatz beschriebene Schwellenwertverstoß aufgetreten ist. Bei diesem Element handelt es sich um eine lokale Zeitmarke.

Tabelle 213. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Schwellenwertverstöße	event_thresholdviolations	-

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Monitorelementen für Schwellenwertverstöße die Analyse eines Schwellenwertverstoßes.

time_started - Startzeitpunkt (Monitorelement)

Der Zeitpunkt, zu dem die Ausführung der von dem Aktivitätsdatensatz beschriebenen Aktivität gestartet wurde. Bei diesem Element handelt es sich um eine lokale Zeitmarke.

Tabelle 214. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse des Verhaltens einer Aktivität.

Falls die Aktivität zurückgewiesen wurde, lautet der Wert des Monitorelements **act_exec_time** 0. In diesem Fall entspricht der Wert des Monitorelements **time_started** dem Wert des Monitorelements **time_completed**.

bottom - Obere Grenze von Histogrammbin (Monitorelement)

Die Obergrenze (Teil des Bereichs) des für ein Histogrammbin festgelegten Bereichs. Der Wert dieses Monitorelements stellt gleichzeitig die Untergrenze (nicht Teil des Bereichs) des Bereichs des nachfolgenden Histogrammbins dar, soweit vorhanden.

Tabelle 215. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_histogrambin	-

Verwendung

Mit diesem Element kann in Verbindung mit dem entsprechenden Element **bottom** der Bereich für ein Bin in einem Histogramm bestimmt werden.

total_disp_run_queue_time - Gesamtwartezeit in Ausführungswarteschlange des Dispatchers (Monitorelement)

Die Gesamtzeit, die Anforderungen, die in dieser Serviceklasse ausgeführt wurden, für das Warten auf den Zugriff auf die CPU aufgewendet haben. Dieser Wert wird in Mikrosekunden angegeben.

Tabelle 216. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_XML_METRICS_BY_ROW - Formatierte zeilenbasierte Ausgabe für alle Messwerte abrufen	Nicht zutreffend; meldet alle Elemente zurück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_CONNECTION (Tabellenfunktion) - Verbindungsmessdaten abrufen	REQUEST METRICS BASE

Tabelle 216. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_CONNECTION_DETAILS (Tabellenfunktion) - Detaillierte Verbindungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS - Detaillierte Messdaten für Paketcacheinträge abrufen	ACTIVITY METRICS BASE
MON_GET_ROUTINE (Tabellenfunktion) - Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	REQUEST METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	REQUEST METRICS BASE

Tabelle 216. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	REQUEST METRICS BASE

Tabelle 217. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Paketcache	Im Dokument 'activity_metrics' zurückgemeldet.	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE

Verwendung

Zusammen mit dem Monitorelement **total_cpu_time** kann das Monitorelement **total_disp_run_queue_time** dazu verwendet werden, ein Maß für die Nutzungskonkurrenz der CPU-Ressource zu berechnen. Diese Maß wird auf einer Skala von 0 bis 1 ermittelt, wobei niedrigere Werte einen höheren Konkurrenzgrad bei der Nutzung der CPU-Ressource bedeuten. Dieses Maß, das als CPU-Geschwindigkeit bezeichnet wird, berechnet sich aus dem Wert für die gemessene Zeitdauer, die Operationen in einer Serviceklasse Zugriff auf die CPU haben, dividiert durch die Gesamtzeit, die für den Zugriff auf die CPU oder für das Warten auf den CPU-Zugriff aufgewendet wurde. Es stellt ein Maß für die Effizienz, mit der Operationen, die ausgeführt werden, im Verhältnis zu der Effizienz dar, mit der sie ausgeführt werden könnten, wenn sie nie auf die CPU warten müssten. Es gilt die folgende Formel:

$$\text{CPU-Geschwindigkeit} = \text{total_cpu_time} / (\text{total_cpu_time} + \text{total_disp_run_queue_time})$$

Bei Rückgabe durch die Funktion WLM_GET_SERVICE_SUBCLASS_STATS oder WLM_GET_WORKLOAD_STATS stellt dieses Monitorelement die Gesamtwartezeit in der Ausführungswarteschlange des Dispatchers seit der letzten Zurücksetzung der Statistiken dar.

Bei Rückgabe durch die Tabellenfunktion MON_SAMPLE_SERVICE_CLASS_METRICS oder MON_SAMPLE_WORKLOAD_METRICS stellt dieses Monitorelement die Gesamtwartezeit in der Ausführungswarteschlange des Dispatchers seit der Ausführung der Funktion dar.

uow_completed_total - Gesamtzahl beendeter UOWs (Monitorelement)

Die Gesamtzahl der UOWs (Units of Work, Arbeitseinheiten), die entweder durch COMMIT oder durch ROLLBACK beendet wurden.

Tabelle 218. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	REQUEST METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	REQUEST METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	REQUEST METRICS BASE

Tabelle 219. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst

Verwendung

Bei Rückgabe durch die Funktion WLM_GET_SERVICE_SUBCLASS_STATS oder WLM_GET_WORKLOAD_STATS stellt dieses Monitorelement die Gesamtzahl der beendeten UOWs seit der letzten Zurücksetzung der Statistiken dar.

Bei Rückgabe durch die Tabellenfunktion MON_SAMPLE_SERVICE_CLASS_METRICS oder MON_SAMPLE_WORKLOAD_METRICS stellt dieses Monitorelement die Gesamtzahl beendeter UOWs seit der Ausführung der Funktion dar.

uow_comp_status - Fertigstellungsstatus der UOW

Der Status der UOW (Unit of Work, Arbeitseinheit) und wie sie gestoppt wurde.

Elementkennung

uow_comp_status

Elementtyp

Information

Tabelle 220. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl	UOW (Unit of Work)
DCS-Anwendung	dcs_appl	Einfach

Tabelle 221. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Transaktionen	event_xact	Immer erfasst

Verwendung

Mithilfe dieses Elements kann ermittelt werden, ob die UOW (Unit of Work, Arbeitseinheit) aufgrund eines Deadlocks oder einer abnormalen Beendigung endete. Folgendes kann auf die UOW zutreffen:

- Sie wurde aufgrund einer COMMIT-Anweisung festgeschrieben.
- Sie wurde aufgrund einer ROLLBACK-Anweisung rückgängig gemacht.
- Sie wurde aufgrund eines Deadlocks rückgängig gemacht.
- Sie wurde aufgrund einer abnormalen Beendigung rückgängig gemacht.
- Sie wurde bei einer normalen Beendigung der Anwendung festgeschrieben.
- Sie war unbekannt aufgrund eines Befehls FLUSH EVENT MONITOR, für den UOWs in Bearbeitung waren.

Anmerkung: API-Benutzer sollten die Kopfdatendatei (*sqlmon.h*) heranziehen, die Definitionen der Konstanten des Datenbanksystemmonitors enthält.

uow_elapsed_time - Abgelaufene Zeit der letzten UOW

Die abgelaufene Ausführungszeit der zuletzt abgeschlossenen UOW (Unit of Work).

Elementkennung

uow_elapsed_time

Elementtyp

Zeit

Tabelle 222. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl	UOW (Unit of Work), Zeitmarke
DCS-Anwendung	dcs_appl	UOW (Unit of Work), Zeitmarke

Verwendung

Dieses Element gibt Aufschluss darüber, wie viel Zeit UOWs benötigen, um vollständig ausgeführt zu werden.

Dieses Element besteht aus zwei Unterelementen, die die abgelaufene Zeit in Sekunden und Mikrosekunden (Millionstel einer Sekunde) zurückmelden. Die Namen der Unterelemente können durch Hinzufügen von "_s" und "_ms" zum Namen dieses Monitorelements abgeleitet werden. Zum Abrufen der Gesamtzeit, die für dieses Monitorelement abgelaufen ist, müssen die Werte der beiden Unterelemente addiert werden. Wenn beispielsweise das Unterelement "_s" den Wert 3 und das Unterelement "_ms" den Wert 20 aufweist, beträgt die für das Monitorelement insgesamt abgelaufene Zeit 3,00002 Sekunden.

uow_id - UOW-ID (Monitorelement)

Die Kennung der UOW (Unit of Work). Die UOW-ID ist innerhalb einer Anwendungskennung eindeutig.

Tabelle 223. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion) - Liste von Aktivitäten zurückgeben	ACTIVITY METRICS BASE

Tabelle 224. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Sperren	-	Immer erfasst
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity	Immer erfasst
Aktivitäten	event_activitystmt	Immer erfasst
Aktivitäten	event_activityvals	Immer erfasst
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Schwellenwertverstöße	event_thresholdviolations	Immer erfasst
Änderungsprotokoll	ddlstmtexec txncompletion	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse des Verhaltens einer Aktivität.

Sie können dieses Element auch in Verbindung mit den Monitorelementen **activity_id** und **appl_id** verwenden, um eine Aktivität eindeutig zu identifizieren.

uow_lifetime_avg - Durchschnittliche UOW-Lebensdauer (Monitorelement)

Die durchschnittliche Lebensdauer (Laufzeit) einer UOW (Unit of Work, Arbeitseinheit). Der Wert wird in Millisekunden gemessen.

Tabelle 225. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	REQUEST METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	REQUEST METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	REQUEST METRICS BASE

Tabelle 226. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst

Verwendung

Bei Rückgabe durch die Funktion WLM_GET_SERVICE_SUBCLASS_STATS oder WLM_GET_WORKLOAD_STATS stellt dieses Monitorelement die durchschnittliche UOW-Laufzeit seit der letzten Zurücksetzung der Statistiken dar.

Bei Rückgabe durch die Tabellenfunktion MON_SAMPLE_SERVICE_CLASS_METRICS oder MON_SAMPLE_WORKLOAD_METRICS stellt dieses Monitorelement die durchschnittliche UOW-Laufzeit seit der Ausführung der Funktion dar.

uow_lock_wait_time - Gesamtwartezeit der UOW auf Sperren (Monitorelement)

Die insgesamt abgelaufene Zeit, die die betreffende UOW (Unit of Work, Arbeits-einheit) auf Sperren gewartet hat. Dieser Wert wird in Millisekunden angegeben.

Elementkennung

uow_lock_wait_time

Elementtyp

Zähler

Tabelle 227. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl	UOW (Unit of Work)

Verwendung

Dieses Element kann dabei helfen, die Wertigkeit des Ressourcenkonflikts zu ermitteln.

uow_log_space_used - Verwendeter UOW-Protokollspeicher (Monitorelement)

Die Menge (in Byte) des Protokollspeicherbereichs, der in der aktuellen UOW (Unit of Work, Arbeitseinheit) der überwachten Anwendung verwendet wird.

Tabelle 228. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 229. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl	UOW (Unit of Work)

Tabelle 230. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Transaktionen	event_xact	Immer erfasst
UOW (Unit of Work)	-	Immer erfasst

Verwendung

Dieses Element liefert Informationen, die einen Einblick in die Protokollanforderungen auf UOW (Unit of Work, Arbeitseinheit)-Ebene geben.

uow_start_time - Zeitmarke beim Start der UOW (Monitorelement)

Der Zeitpunkt (Datum und Uhrzeit), zu dem die UOW (Unit of Work, Arbeitseinheit) erstmals Datenbankressourcen benötigte.

Tabelle 231. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 232. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl	UOW (Unit of Work), Zeitmarke
DCS-Anwendung	dcs_appl	UOW (Unit of Work), Zeitmarke

Tabelle 233. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
UOW (Unit of Work)	-	-
Transaktion	event_xact	-

Verwendung

Dieser Ressourcenbedarf tritt bei der ersten Ausführung einer SQL-Anweisung der betreffenden UOW (Unit of Work, Arbeitseinheit) wie folgt ein:

- Bei der ersten UOW (Unit of Work, Arbeitseinheit) zum Zeitpunkt der ersten Datenbankankforderung (Ausführung der SQL-Anweisung) nach **conn_complete_time**.
- Bei nachfolgenden UOWs (Units of Work, Arbeitseinheiten) zum Zeitpunkt der ersten Datenbankankforderung (Ausführung der SQL-Anweisung) nach der vorherigen COMMIT- oder ROLLBACK-Operation

Anmerkung: Im Handbuch *SQL Reference* werden die Grenzen einer UOW (Unit of Work, Arbeitseinheit) als COMMIT- oder ROLLBACK-Punkte definiert.

Der Datenbanksystemmonitor schließt die zwischen einer COMMIT-/ROLLBACK-Operation und der nächsten SQL-Anweisung liegende Zeit aus seiner Definition einer UOW (Unit of Work, Arbeitseinheit) aus. Diese Messmethode gibt die Zeit, die der Datenbankmanager mit der Verarbeitung von Datenbankankforderungen verbringt, getrennt von der Zeit wieder, die in der Anwendungslogik vor der ersten SQL-Anweisung der betreffenden UOW (Unit of Work, Arbeitseinheit) verbracht wird. Die verstrichene UOW (Unit of Work, Arbeitseinheit)-Zeit umfasst jedoch die Zeit, in der Anwendungslogik zwischen SQL-Anweisungen innerhalb der UOW (Unit of Work, Arbeitseinheit) ausgeführt wurde.

Dieses Element kann zusammen mit dem Monitorelement **uow_stop_time** verwendet werden, um die insgesamt verstrichene Zeit der UOW (Unit of Work, Arbeitseinheit) zu berechnen, und zusammen mit dem Monitorelement **prev_uow_stop_time**, um die Zeit zu berechnen, die zwischen UOWs (Units of Work, Arbeitseinheiten) in der Anwendung verbracht wurde.

Mithilfe der Monitorelemente **uow_stop_time** und **prev_uow_stop_time** kann die verstrichene Zeit einer UOW (Unit of Work, Arbeitseinheit) gemäß Definition im Handbuch *SQL Reference* berechnet werden.

uow_status - UOW-Status

Der Status der UOW (Unit of Work, Arbeitseinheit).

Elementkennung

uow_status

Elementtyp

Information

Tabelle 234. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Transaktionen	event_xact	Immer erfasst

Verwendung

Mithilfe dieses Elements kann der Status einer UOW (Unit of Work, Arbeitseinheit) ermittelt werden. API-Benutzer sollten die Kopfdatei 'sqlmon.h' heranziehen, die Definitionen der Konstanten des Datenbanksystems `monitors` enthält.

uow_stop_time - Zeitmarke beim Stopp der UOW (Monitorelement)

Der Zeitpunkt (Datum und Uhrzeit), zu dem die letzte UOW (Unit of Work, Arbeitseinheit) beendet wurde. Dies geschieht, wenn Datenbankänderungen festgeschrieben oder zurückgesetzt werden.

Tabelle 235. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl	UOW (Unit of Work), Zeitmarke
DCS-Anwendung	dcs_appl	UOW (Unit of Work), Zeitmarke

Tabelle 236. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
UOW (Unit of Work)	-	-

Verwendung

Verwenden Sie dieses Element zusammen mit dem Monitorelement `prev_uow_stop_time`, um die zwischen COMMIT-/ROLLBACK-Punkten insgesamt abgelaufene Zeit zu berechnen, und zusammen mit dem Monitorelement `uow_start_time`, um die abgelaufene Zeit der letzten UOW (Unit of Work, Arbeitseinheit) zu berechnen.

Der Inhalt der Zeitmarke wird wie folgt festgelegt:

- Wenn die Anwendung eine UOW (Unit of Work, Arbeitseinheit) abgeschlossen und noch keine neue gestartet hat (gemäß Definition im Monitorelement `uow_start_time`), ist dieses Element eine gültige Zeitmarke ungleich Null.
- Wenn die Anwendung momentan eine UOW (Unit of Work, Arbeitseinheit) ausführt, enthält dieses Element Nullen.
- Wenn die Anwendung erstmals eine Verbindung zur Datenbank herstellt, wird dieses Element auf den Wert des Monitorelements `conn_complete_time` gesetzt.

Beim Start einer neuen UOW (Unit of Work, Arbeitseinheit) wird der Inhalt dieses Elements in das Monitorelement `prev_uow_stop_time` versetzt.

uow_throughput - UOW-Durchsatz (Monitorelement)

Die Ausführungsrate von UOWs (Units of Work, Arbeitseinheiten), gemessen in UOWs pro Sekunde.

Tabelle 237. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_SAMPLE_SERVICE_CLASS_METRICS - Stichprobe für Messdaten zur Serviceklasse abrufen	REQUEST METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	REQUEST METRICS BASE
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	REQUEST METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	REQUEST METRICS BASE

Tabelle 238. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	Immer erfasst

Verwendung

Bei Rückgabe durch die Funktion WLM_GET_SERVICE_SUBCLASS_STATS oder WLM_GET_WORKLOAD_STATS stellt dieses Monitorelement den UOW-Durchsatz seit der letzten Zurücksetzung der Statistiken dar.

Bei Rückgabe durch die Tabellenfunktion MON_SAMPLE_SERVICE_CLASS_METRICS oder MON_SAMPLE_WORKLOAD_METRICS stellt dieses Monitorelement den UOW-Durchsatz seit der Ausführung der Funktion dar.

uow_total_time_top - Maximaler Wert für UOW-Gesamtzeit (Monitorelement)

Der Höchstwert für UOW-Lebensdauer in Millisekunden.

Tabelle 239. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion) - Statistikdaten zu Serviceunterklassen zurückgeben	REQUEST METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	REQUEST METRICS BASE

Tabelle 240. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wlstats	Immer erfasst

Tabelle 240. Informationen zur Ereignisüberwachung (Forts.)

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_scstats	Immer erfasst

Verwendung

Dieses Element kann bei der Beurteilung der Effektivität des Schwellenwerts für UOWTOTALTIME hilfreich und somit auch eine wertvolle Entscheidungshilfe für die Konfiguration eines solchen Schwellenwerts sein.

Bei Serviceklassen gibt dieses Monitorelement -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Serviceklasse auf NONE gesetzt ist.

Bei Auslastungen (Workloads) gibt dieses Monitorelement den Wert -1 zurück, wenn COLLECT AGGREGATE ACTIVITY DATA für die jeweilige Auslastung auf NONE gesetzt ist.

Für eine Serviceklasse werden Messungen, die für diesen Höchstwert erfasst werden, für die Serviceklasse berechnet, die durch die Auslastung zugeordnet wurde. Eine Zuordnung durch ein Arbeitsaktionsset zum Ändern der Serviceklasse einer Aktivität wirkt sich auf diesen Höchstwert nicht aus.

wl_work_action_set_id - Arbeitsaktionsset-ID der Auslastung (Monitorelement)

Wenn die Aktivität innerhalb einer Arbeitsklasse im Auslastungsbereich (Workload) kategorisiert wurde, zeigt dieses Monitorelement die ID des Arbeitsaktionssets an, die dem Arbeitsklassenset, zu dem die Arbeitsklasse gehört, zugeordnet ist. Andernfalls zeigt dieses Monitorelement den Wert 0 an.

Tabelle 241. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 242. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Mit diesem Monitorelement können Sie in Verbindung mit dem Monitorelement **wl_work_class_id** die Auslastungsarbeitsklasse der Aktivität (sofern vorhanden) eindeutig zu identifizieren.

wl_work_class_id - Auslastungsarbeitsklassen-ID (Monitorelement)

Wenn die Aktivität innerhalb einer Arbeitsklasse im Auslastungsbereich kategorisiert wurde, zeigt dieses Monitorelement die Kennung (ID) der Arbeitsklasse an. Andernfalls zeigt dieses Monitorelement 0 an.

Tabelle 243. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE

Tabelle 244. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity	Immer erfasst

Verwendung

Mit diesem Monitorelement können Sie in Verbindung mit dem Monitorelement **wl_work_action_set_id** die Auslastungsarbeitsklasse der Aktivität (sofern vorhanden) eindeutig zu identifizieren.

wlm_queue_assignments_total - Gesamtzahl der Warteschlangenzuweisungen durch den Workload-Manager (Monitorelement)

Die Häufigkeit, mit der Aktivitäten oder Verbindungen durch einen WLM-Schwellenwert (WLM - Workload-Manager) in eine Warteschlange gestellt wurden.

Tabelle 245. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_XML_METRICS_BY_ROW - Formatierte zeilenbasierte Ausgabe für alle Messwerte abrufen	Nicht zutreffend; meldet alle Elemente zurück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_CONNECTION (Tabellenfunktion) - Verbindungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS (Tabellenfunktion) - Detaillierte Verbindungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS (Tabellenfunktion) - Detaillierte Messdaten für Paketcacheinträge abrufen	ACTIVITY METRICS BASE

Tabelle 245. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_ROUTINE (Tabellenfunktion) - Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE

Tabelle 246. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Paketcache	Im Dokument 'activity_metrics' zurückgemeldet.	ACTIVITY METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE

wlm_queue_time_total - Gesamtwartezeit für Workload-Manager-Warteschlange (Monitorelement)

Die Zeit, die zum Warten auf einen WLM-Schwellenwert (WLM, Workload-Manager) zum Einreihen in eine Warteschlange aufgewendet wurde. Dieser Wert wird in Millisekunden angegeben.

Tabelle 247. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung(Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_XML_METRICS_BY_ROW - Formatierte zeilenbasierte Ausgabe für alle Messwerte abrufen	Nicht zutreffend; meldet alle Elemente zurück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_FORMAT_XML_TIMES_BY_ROW - Formatierte zeilenbasierte Hierarchie mit kombinierten Warte- und Verarbeitungszeiten abrufen	Nicht zutreffend; meldet alle Elemente zurück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_FORMAT_XML_WAIT_TIMES_BY_ROW - Formatierte zeilenbasierte Ausgabe für Wartezeiten abrufen	Nicht zutreffend; meldet alle Elemente zurück, die in dem der Formatierungsfunktion als Eingabe bereitgestellten XML-Dokument enthalten sind.
MON_GET_ACTIVITY_DETAILS (Tabellenfunktion) - Vollständige Aktivitätsdetails abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
MON_GET_CONNECTION (Tabellenfunktion) - Verbindungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS (Tabellenfunktion) - Detaillierte Verbindungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_PKG_CACHE_STMT (Tabellenfunktion) - Aktivitätsmessdaten für SQL-Anweisungen im Paketcache abrufen	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT_DETAILS (Tabellenfunktion) - Detaillierte Messdaten für Paketcacheinträge abrufen	ACTIVITY METRICS BASE
MON_GET_ROUTINE (Tabellenfunktion) - Zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_ROUTINE_DETAILS (Tabellenfunktion) - Detaillierte zusammengefasste Ausführungsmessdaten für Routinen abrufen	Immer erfasst
MON_GET_SERVICE_SUBCLASS (Tabellenfunktion) - Serviceunterklassenmessdaten abrufen	REQUEST METRICS BASE

Tabelle 247. Informationen zur Überwachung mit Tabellenfunktionen (Forts.)

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_SERVICE_SUBCLASS_DETAILS (Tabellenfunktion) - Detaillierte Serviceunterklassenmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	REQUEST METRICS BASE

Tabelle 248. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Aktivitäten	event_activitymetrics	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Paketcache	Im Dokument 'activity_metrics' zurückgemeldet.	ACTIVITY METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	REQUEST METRICS BASE

wlo_completed_total - Gesamtzahl abgeschlossener Workloadvorkommen (Monitorelement)

Die Anzahl von abgeschlossenen Verarbeitungsprozessen, die seit dem letzten Zurücksetzen ausgeführt wurden.

Tabelle 249. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 250. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wlstats	-

Verwendung

Mit diesem Element kann festgestellt werden, wie viele Ausführungen eines bestimmten Verarbeitungsprozesses zur Auslastung des Systems beitragen.

work_action_set_id - ID des Arbeitsaktionssets (Monitorelement)

Die ID des Arbeitsaktionssets, auf das sich dieser Statistikdatensatz bezieht.

Tabelle 251. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_histogrambin	Immer erfasst
Statistiken	event_wcstats	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Elementen für die Protokollierung von Aktivitäten eine Analyse des Verhaltens einer Aktivität und in Verbindung mit anderen Statistikelementen eine Analyse einer Arbeitsklasse.

Der Wert dieses Elements ist 0, wenn die folgenden Bedingungen zutreffen:

- Das Element wird in einer logischen Datengruppe 'event_histogrambin' zurückgegeben.
- Die Histogrammdata werden für ein Objekt erfasst, das keine Arbeitsklasse ist.

work_action_set_name - Name des Arbeitsaktionssets (Monitorelement)

Der Name des Arbeitsaktionssets, auf den sich die als Teil dieses Ereignisses angezeigten Statistikdaten beziehen.

Tabelle 252. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE
WLM_GET_WORK_ACTION_SET_STATS (Tabellenfunktion) - Statistikdaten zu Arbeitsaktionssets zurückgeben	ACTIVITY METRICS BASE

Tabelle 253. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	-
Statistiken	event_wcstats	-

Verwendung

Mit diesem Element lässt sich in Verbindung mit dem Element **work_class_name** die Arbeitsklasse eindeutig identifizieren, deren Statistikdaten in diesem Datensatz angezeigt werden. Ferner lässt sich die Arbeitsklasse eindeutig identifizieren, die Domäne der Schwellenwertwarteschlange ist, deren Statistikdaten in diesem Datensatz angezeigt werden.

work_class_id - Arbeitsklassen-ID (Monitorelement)

Die ID der Arbeitsklasse, auf die sich der Statistikdatensatz bezieht.

Tabelle 254. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_wcstats	Immer erfasst
Statistiken	event_histogrambin	Immer erfasst

Verwendung

Dieses Element ermöglicht in Verbindung mit anderen Statistikelementen eine Analyse einer Arbeitsklasse.

Der Wert dieses Elements ist 0, wenn die folgenden Bedingungen zutreffen:

- Das Element wird in einer logischen Datengruppe 'event_histogrambin' zurückgegeben.
- Die Histogrammdata werden für ein Objekt erfasst, das keine Arbeitsklasse ist.

work_class_name - Name der Arbeitsklasse (Monitorelement)

Der Name der Arbeitsklasse, auf die sich die als Teil dieses Ereignisses angezeigten Statistikdaten beziehen.

Tabelle 255. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE
WLM_GET_WORK_ACTION_SET_STATS (Tabellenfunktion) - Statistikdaten zu Arbeitsaktionssets zurückgeben	ACTIVITY METRICS BASE

Tabelle 256. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Statistiken	event_qstats	-
Statistiken	event_wcstats	-

Verwendung

Mit diesem Element lässt sich in Verbindung mit dem Element **work_action_set_name** die Arbeitsklasse eindeutig identifizieren, deren Statistikdaten in diesem Datensatz angezeigt werden. Ferner lässt sich die Arbeitsklasse eindeutig identifizieren, die Domäne der Schwellenwertwarteschlange ist, deren Statis-

tikdaten in diesem Datensatz angezeigt werden.

workload_id - Workload-ID (Monitorelement)

Eine ganze Zahl (Integer), die eine Auslastung (Workload) eindeutig identifiziert.

Tabelle 257. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	ACTIVITY METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen	ACTIVITY METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	ACTIVITY METRICS BASE

Tabelle 258. Informationen zur Momentaufnahmeüberwachung

Ebene der Momentaufnahme	Logische Datengruppierung	Monitorschalter
Anwendung	appl_info	Einfach

Tabelle 259. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Sperren	-	Immer erfasst
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	Immer erfasst
Statistiken	event_wlstats	Immer erfasst
Statistiken	event_histogrambin	Immer erfasst
Aktivitäten	event_activity	Immer erfasst
Schwellenwertverstöße	event_thresholdviolations	Immer erfasst

Verwendung

Mithilfe dieser ID können Sie die Auslastung eindeutig identifizieren, zu der diese Aktivität, diese Anwendung, dieser Histogrammbin oder dieser Auslastungsstatistikatz gehört.

Der Wert dieses Elements ist 0, wenn die folgenden Bedingungen zutreffen:

- Das Element wird in einer logischen Datengruppe 'event_histogrambin' zurückgegeben.

- Die Histogramm Daten werden für ein Objekt erfasst, das keine Auslastung (Workload) ist.

workload_name - Name der Workload (Monitorelement)

Der Name der Workload.

Tabelle 260. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung (Weitere Informationen zu Ebenen der Monitorelementerfassung finden Sie in .)
MON_FORMAT_LOCK_NAME (Tabellenfunktion) - Internen Sperrennamen formatieren und Details zurückgeben	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_WORKLOAD (Tabellenfunktion) - Auslastungsmessdaten abrufen	ACTIVITY METRICS BASE
MON_GET_WORKLOAD_DETAILS (Tabellenfunktion) - Detaillierte Auslastungsmessdaten abrufen	ACTIVITY METRICS BASE
MON_SAMPLE_WORKLOAD_METRICS - Stichprobe für Auslastungsmessdaten abrufen	ACTIVITY METRICS BASE
WLM_GET_QUEUE_STATS (Tabellenfunktion) - Statistikdaten zur Schwellenwertwarteschlange zurückgeben	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE
WLM_GET_WORKLOAD_STATS (Tabellenfunktion) - Auslastungsstatistikdaten zurückgeben	ACTIVITY METRICS BASE

Tabelle 261. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
Sperren	-	Immer erfasst
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity (im Dokument 'details_xml' zurückgemeldet)	ACTIVITY METRICS BASE
Statistiken	event_scstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE
Statistiken	event_wlstats (im Dokument 'metrics' zurückgemeldet)	REQUEST METRICS BASE

Tabelle 261. Informationen zur Ereignisüberwachung (Forts.)

Ereignistyp	Logische Datengruppierung	Monitorschalter
UOW (Unit of Work)	Im Dokument 'system_metrics' zurückgemeldet.	Immer erfasst
Statistiken	event_wlstats	Immer erfasst

Verwendung

Beim Statistikereignismonitor und bei den Workloadtabellenfunktionen gibt der Workloadname die Workload an, für die Statistiken oder Messdaten erfasst und zurückgemeldet werden. Beim UOW-Ereignismonitor und bei den UOW-Tabellenfunktionen gibt der Workloadname die Workload an, der die UOW zugeordnet war.

Mithilfe des Workloadnamens werden UOWs oder Informationsmengen identifiziert, die für eine bestimmte relevante Workload gelten.

workload_occurrence_id - ID des Workloadvorkommens (Monitorelement)

Die Kennung (ID) des Workloadvorkommens, zu dem diese Aktivität gehört.

Tabelle 262. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion) - In einer Serviceklasse ausgeführte Agenten auflisten	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE

Tabelle 263. Informationen zur Ereignisüberwachung

Ereignistyp	Logische Datengruppierung	Monitorschalter
UOW (Unit of Work)	-	Immer erfasst
Aktivitäten	event_activity	Immer erfasst

Verwendung

Mit diesem Element kann das Workloadvorkommen identifiziert werden, das die Aktivität übergeben hat.

workload_occurrence_state - Status des Workloadvorkommens (Monitorelement)

Der Status des Workloadvorkommens (Auslastungsvorkommens).

Tabelle 264. Informationen zur Überwachung mit Tabellenfunktionen

Tabellenfunktion	Ebene der Monitorelementerfassung
MON_GET_UNIT_OF_WORK (Tabellenfunktion) - UOW-Messdaten abrufen	ACTIVITY METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS (Tabellenfunktion) - Detaillierte UOW-Messdaten abrufen (im XML-Dokument in der Spalte DETAILS zurückgemeldet)	ACTIVITY METRICS BASE
WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion) - Workloadvorkommen auflisten	ACTIVITY METRICS BASE

Verwendung

Mögliche Werte:

DECOUPLED

Das Workloadvorkommen hat keinen zugeordneten Koordinatoragenten (Konzentratorfall).

DISCONNECTPEND

Das Workloadvorkommen trennt die Verbindung zur Datenbank.

FORCED

Das Workloadvorkommen wurde zwangsweise beendet und aus dem System entfernt.

INTERRUPTED

Das Workloadvorkommen wurde unterbrochen.

QUEUED

Der Koordinatoragent des Workloadvorkommens wurde durch einen Schwellenwert zum Einreihen in eine Warteschlange des Workload-Managements in eine Warteschlange gestellt. In einer Umgebung mit partitionierten Datenbanken kann dieser Status darauf hindeuten, dass der Koordinatoragent einen Prozedurfernaufruf an ein anderes Member abgesetzt hat, um Schwellenwerttickets zu erhalten, und noch keine Antwort erhalten hat.

TRANSIENT

Das Workloadvorkommen wurde noch keiner Servicesuperklasse zugeordnet.

UOWEXEC

Das Workloadvorkommen verarbeitet zurzeit eine Anforderung.

UOWWAIT

Das Workloadvorkommen wartet zurzeit auf eine Anforderung vom Client.

Befehle

SET WORKLOAD

Gibt die Workload an, der die Datenbankverbindung zugeordnet werden soll. Dieser Befehl kann vor dem Herstellen einer Datenbankverbindung abgesetzt werden oder er kann nach dem Herstellen der Verbindung verwendet werden, um die aktuelle Verbindung neu zuzuordnen. Wurde die Verbindung bereits hergestellt, wird die erneute Workloadzuordnung am Anfang der nächsten UOW durchgeführt.

Berechtigung

Keine, aber siehe Hinweise

Erforderliche Verbindung

Keine

Befehlssyntax

```
➤ SET WORKLOAD TO {AUTOMATIC | SYSDEFAULTADMWORKLOAD} ➤
```

Befehlsparameter

AUTOMATIC

Gibt an, dass die Datenbankverbindung einer Workload zugeordnet wird, die von der automatisch durch den Server durchgeführten Workloadbewertung ausgewählt wird.

SYSDEFAULTADMWORKLOAD

Gibt an, dass die Datenbankverbindung der Workload SYSDEFAULTADMWORKLOAD zugeordnet wird, sodass Benutzer, die über die Berechtigung *accessctrl*, *dataaccess*, *wlmadm*, *secadm* oder *dbadm* verfügen, die normale Workloadbewertung übergehen können.

Beispiele

Gehen Sie wie folgt vor, um die Verbindung der Workload **SYSDEFAULTADMWORKLOAD** zuzuordnen:

```
SET WORKLOAD TO SYSDEFAULTADMWORKLOAD
```

Gehen Sie wie folgt vor, um die Workloadzuordnung zurückzusetzen, sodass die Workload verwendet wird, die von der durch den Server durchgeführten Workloadbewertung ausgewählt wird:

```
SET WORKLOAD TO AUTOMATIC
```

Hinweise zur Verwendung

Wenn die Sitzungsberechtigungs-ID der Datenbankverbindung nicht über die Berechtigung *accessctrl*, *dataaccess*, *wlmadm*, *secadm* oder *dbadm* verfügt, kann die Verbindung nicht der Workload **SYSDEFAULTADMWORKLOAD** zugeordnet werden und es wird der SQL-Fehler SQL0552N zurückgegeben. Wird der Befehl **SET WORKLOAD TO SYSDEFAULTADMWORKLOAD** vor dem Herstellen einer Datenbankverbindung abgesetzt,

wird der SQL-Fehler SQL0552N nach dem Herstellen der Datenbankverbindung am Anfang der ersten UOW ausgegeben. Wird der Befehl nach dem Herstellen der Datenbankverbindung abgesetzt, wird der SQL-Fehler SQL0552N am Anfang der nächsten UOW zu dem Zeitpunkt ausgegeben, an dem eigentlich die Neuordnung der Workload erfolgen sollte.

Konfigurationsparameter

wlm_collect_int - Workload-Management-Erfassungsintervall

Dieser Parameter gibt ein Erfassungs- und Zurücksetzungsintervall (in Minuten) für die Workload-Management-Statistik an.

Alle x Minuten (wobei x der Wert des Parameters **wlm_collect_int** ist) werden alle Workload-Management-Statistikdaten gesammelt und an sämtliche aktiven Ereignismonitore für Statistiken gesendet. Anschließend wird die Statistik zurückgesetzt. Wenn ein aktiver Statistikereignismonitor vorhanden ist, wird die Statistik in Abhängigkeit davon, wie er erstellt wurde, in eine Datei, in eine Pipe oder in eine Tabelle geschrieben. Wenn kein aktiver Ereignismotor vorhanden ist, wird die Statistik nur zurückgesetzt, aber nicht erfasst.

Erfassungen erfolgen in den angegebenen Intervallzeiten relativ zum Zeitpunkt Sonntag 00:00:00 Uhr. Nach der Aktivierung des Katalogmembers erfolgt die nächste Erfassung zu Beginn des nächsten nach der Aktivierung terminierten Intervalls. Das terminierte Intervall ist nicht relativ zur Aktivierungszeit des Katalogmembers. Ist ein Member zum Zeitpunkt der Erfassung nicht aktiv, werden für dieses Member keine Statistikdaten zusammengestellt. Wurde der Intervallwert beispielsweise auf 60 gesetzt und das Katalogmember wurde am Sonntag um 9:24 Uhr aktiviert, erfolgen die Erfassungen stündlich zur vollen Stunde. Das heißt, die nächste Erfassung erfolgt um 10:00 Uhr. Wenn das Member um 10:00 Uhr nicht aktiv ist, werden für dieses Member keine Statistikdaten zusammengestellt.

Der Prozess der Erfassung und Zurücksetzung wird vom Katalogmember ausgelöst. Für das Member muss der Parameter **wlm_collect_int** angegeben sein. Er wird für keine anderen Member verwendet.

Konfigurationstyp

Datenbank

Parametertyp

- Online konfigurierbar

Standardwert [Bereich]

0 [0 (keine Erfassung), 5 - 32 767]

Die vom Ereignismonitor für Statistiken erfasste Workload-Management-Statistik kann zum Überwachen des Kurz- und Langzeitsystemverhaltens verwendet werden. Ein kleines Intervall kann zur Ermittlung des Kurz- und Langzeitsystemverhaltens verwendet werden, da die Ergebnisse zusammengefasst werden können, um ein Langzeitverhalten zu erhalten. Die erforderliche manuelle Zusammenfassung der Ergebnisse aus verschiedenen Intervallen verkompliziert allerdings die Analyse. Wenn es nicht erforderlich ist, führt ein kleines Intervall zu einer unnötigen Erhöhung der Verarbeitungszeit. Aus diesem Grund sollten Sie das Intervall verkleinern, um das Kurzzeitverhalten zu erfassen, und das Intervall vergrößern, um die Verarbeitungszeit zu reduzieren, wenn eine Analyse des Langzeitverhaltens bereits ausreicht.

Das Intervall muss jeweils für eine Datenbank, nicht für die einzelnen SQL-Anforderungen, Befehlsaufrufe oder Anwendungen, angepasst werden. Andere Konfigurationsparameter müssen nicht berücksichtigt werden.

Anmerkung: Alle WLM-Statistiktabellenfunktionen geben Statistikdaten zurück, die seit der letzten Zurücksetzung der Statistik gesammelt wurden. Die Statistik wird regelmäßig auf der Basis des durch diesen Konfigurationsparameter angegebenen Intervalls zurückgesetzt.

wlm_dispatcher - Workload-Management-Dispatcher

Dieser Parameter aktiviert (YES) oder inaktiviert (NO) den DB2 Workload Management-Dispatcher. Ein aktivierter Dispatcher ermöglicht standardmäßig die Einstellung der CPU-Begrenzungen.

Konfigurationstyp

Datenbankmanager

Gilt für

- Datenbankserver mit lokalen und fernen Clients
- Datenbankserver mit lokalen Clients
- Partitionierten Datenbankserver mit lokalen und fernen Clients

Parametertyp

Online konfigurierbar

Weitergabeklasse

Sofort

Standardwert [Bereich]

NO [NO; YES]

Beim Aktualisieren des DB2-Datenbankmanagers wird als Wert für den Konfigurationsparameter **wlm_dispatcher** des Datenbankmanagers der Wert NO eingestellt.

Der Dispatcher stellt Funktionen für die CPU-Zeitplanung auf Serviceklassenebene im DB2-Datenbankmanager durch Zuteilen von CPU-Ressourcen in Form von CPU-Anteilen und/oder CPU-Begrenzungen zur Verfügung.

Wenn der Dispatcher aktiviert ist, werden alle Arbeitsoperationen in Benutzer- und Wartungsserviceklassen der Steuerung durch den Dispatcher unterstellt. Ist der Dispatcher aktiviert, werden die Einstellungen für CPU-Begrenzungen standardmäßig durch den Dispatcher erzwungen. Damit die Zuteilung von CPU-Ressourcen in Form von CPU-Anteilen verwendet werden kann, muss der Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers aktiviert sein.

Wenn der Konfigurationsparameter **wlm_dispatcher** auf YES gesetzt ist, gelten die folgenden Bedingungen:

- Wenn eine Serviceklasse über eine Prioritätseinstellung verfügt, die nicht der Standardeinstellung entspricht, wird im Zeitpunkt der Datenbankaktivierung eine Warnung in das DB2-Diagnoseprotokoll (db2diag) und in das Protokoll mit Benachrichtigungen für die Systemverwaltung geschrieben.
- Bei jedem Versuch, für die Agentenpriorität in einer Serviceklasse einen anderen als den Standardwert festzulegen, wird eine Warnung an die Anwendung zurückgegeben, von der die Anweisung zum Erstellen oder Ändern der Serviceklasse ausgeführt wurde.

wlm_disp_concur - WLM-Dispatcher-Thread-Parallelität

Dieser Parameter gibt an, wie der DB2 Workload Manager-Dispatcher (WLM-Dispatcher) die Einstellung für den gemeinsamen Threadzugriff festlegt. Sie können auch manuell einen festen Wert für den gemeinsamen Threadzugriff festlegen.

Konfigurationstyp

Datenbankmanager

Gilt für

- Datenbankserver mit lokalen und fernen Clients
- Datenbankserver mit lokalen Clients
- Partitionierten Datenbankserver mit lokalen und fernen Clients

Parametertyp

Online konfigurierbar

Weitergabeklasse

Sofort

Standardwert [Bereich]

COMPUTED [COMPUTED; *manuell_festgelegter_wert*]

Beim Aktualisieren des DB2-Datenbankmanagers hat der Konfigurationsparameter **wlm_disp_concur** den Wert COMPUTED.

COMPUTED

Der DB2-Datenbankmanager berechnet einen festen Wert für den gemeinsamen Threadzugriff auf der Basis des Vierfachen der Anzahl logischer CPUs, die für den DB2-Datenbankmanager verfügbar sind.

manuell_festgelegter_wert

Sie können manuell einen festen Wert für den gemeinsamen Threadzugriff im Bereich von 1 bis 32767 festlegen. Der optimale Wert ist abhängig von der verwendeten Hardware und Betriebssystemversion und liegt im Allgemeinen im Bereich zwischen dem Zwei- und Vierfachen der Anzahl logischer CPUs auf dem Host oder in der logischen Partition (LPAR).

Maßeinheit

Anzahl der gleichzeitig ausgeführten Threads

Die Einstellung für diesen Konfigurationsparameter des Datenbankmanagers steuert, für wie viele Threads der WLM-Dispatcher die gleichzeitige (parallele) Zuteilung zu den Ausführungswarteschlangen des Betriebssystems zulässt. Als Wert wird ein kleines Vielfaches der Anzahl logischer CPUs angegeben, die für den DB2-Datenbankmanager verfügbar sind. In der Regel kann für diesen Wert das Vierfache der Anzahl verfügbarer logischer CPUs angegeben werden, um mögliche Latenzzeiten der Zeitplanung zu berücksichtigen, die beim Umschalten von Threads zwischen dem aktiven und inaktiven Status entstehen können. Ein optimaler Wert ist gerade groß genug, um sicherzustellen, dass genügend Threads für den DB2-Datenbankmanager verfügbar sind, um die CPUs auf dem Host oder in der LPAR in vollem Umfang zu nutzen. Dieser optimale Wert sorgt für maximale Effizienz und gibt dem DB2-WLM-Dispatcher maximale Kontrolle über die CPU-Zuordnung.

wlm_disp_cpu_shares - CPU-Anteil des WLM-Dispatchers

Dieser Parameter aktiviert (Einstellung YES) oder inaktiviert (Einstellung NO) die Steuerung der CPU-Anteile durch den DB2 Workload Manager-Dispatcher (DB2 WLM-Dispatcher). Standardmäßig steuert ein aktivierter WLM-Dispatcher nur CPU-Begrenzungen.

Konfigurationstyp

Datenbankmanager

Gilt für

- Datenbankserver mit lokalen und fernen Clients
- Datenbankserver mit lokalen Clients
- Partitionierten Datenbankserver mit lokalen und fernen Clients

Parametertyp

Online konfigurierbar

Weitergabeklasse

Sofort

Standardwert [Bereich]

NO [NO; YES]

Beim Aktualisieren des DB2-Datenbankmanagers hat der Konfigurationsparameter **wlm_disp_cpu_shares** des Datenbankmanagers den Wert NO.

Wenn der Wert des Konfigurationsparameters **wlm_dispatcher** des Datenbankmanagers auf YES und der Wert des Konfigurationsparameters **wlm_disp_cpu_shares** des Datenbankmanagers auf NO gesetzt ist, kann der WLM-Dispatcher bei der Verwaltung von Serviceklassen nur CPU-Begrenzungen anwenden.

Ist der Wert des Konfigurationsparameters **wlm_dispatcher** des Datenbankmanagers auf YES gesetzt und der Wert des Konfigurationsparameters **wlm_disp_cpu_shares** des Datenbankmanagers auf YES, kann der WLM-Dispatcher bei der Verwaltung von Serviceklassen sowohl CPU-Begrenzungen als auch CPU-Anteile anwenden. Standardmäßig werden jeder Serviceklasse 1.000 feste CPU-Anteile zugeordnet, um eine gleichmäßige Aufteilung der CPU-Ressourcen zu gewährleisten.

Tabelle 265. Zusammenfassung der erforderlichen Einstellungen der Datenbankmanager-Konfigurationsparameter für die Serviceklassenverwaltung durch den DB2-WLM-Dispatcher

Serviceklassenverwaltung	Einstellung für wlm_dispatcher	Einstellung für wlm_disp_cpu_shares
Keine	NO	NO
CPU-Begrenzungen	YES	NO
CPU-Begrenzungen + CPU-Anteile	YES	YES

wlm_disp_min_util - Minimale CPU-Auslastung des Workload-Manager-Dispatchers

Dieser Parameter gibt die minimale CPU-Auslastung an, die erforderlich ist, um eine Serviceklasse in die vom DB2-WLM verwaltete gemeinsame Nutzung von CPU-Ressourcen einzubinden.

Konfigurationstyp

Datenbankmanager

Gilt für

- Datenbankserver mit lokalen und fernen Clients
- Datenbankserver mit lokalen Clients
- Datenbankserver mit mehreren Mitgliedern mit lokalen und fernen Clients

Parametertyp

Online konfigurierbar

Weitergabeklasse

Sofort

Standardwert [Bereich]

5 [0 bis 100]

Bei einem Upgrade des DB2-Datenbankmanagers weist der Konfigurationsparameter **wlm_disp_min_util** des Datenbankmanagers den Wert 5 auf.

Maßeinheit

Prozent

Ein Beispiel für die Verwendung dieses Datenbankmanager-Konfigurationsparameters: Angenommen es gibt drei Serviceklassen (A, B und C), die jeweils über 1.000 Anteile der CPU-Ressourcen verfügen. In diesem Beispiel wird stets das gleiche Ergebnis erzielt, unabhängig davon, ob es sich um feste oder veränderbare CPU-Anteile handelt. Die CPU-Auslastung der Serviceklassen A und B ist jeweils größer-gleich dem Wert (8 %), der für den Konfigurationsparameter **wlm_disp_min_util** angegeben ist. Die CPU-Auslastung der Serviceklasse C beträgt 3 %, d. h., sie ist kleiner als der Wert (8 %), der für den Konfigurationsparameter **wlm_disp_min_util** angegeben ist. In den Berechnungen für die CPU-Anteile wird davon ausgegangen, dass in der Klasse C keine Arbeitseinheiten zur Ausführung vorliegen. Demzufolge werden die CPU-Ressourcen gleichmäßig auf die Serviceklassen A und B verteilt, d. h., jeder Klasse werden 50 % der CPU-Ressourcen zugeteilt. Sobald in der Serviceklasse Arbeitselemente verarbeitet werden, die eine CPU-Auslastung größer-gleich dem im Konfigurationsparameter **wlm_disp_min_util** angegebenen Wert von 8 % verursachen, werden alle drei Serviceklassen als gleichrangige Kandidaten für die Nutzung der CPU-Ressourcen angesehen, d. h., jeder Klasse werden 33,3 % der CPU-Ressourcen zugeteilt.

In Datenbankumgebungen mit mehreren Mitgliedern wird die Summe der CPU-Auslastung aller Mitglieder auf einem Host oder einer logischen Partition mit dem Wert des Konfigurationsparameters **wlm_disp_min_util** verglichen, um zu ermitteln, ob der Host oder die logische Partition in der vom WLM verwalteten gemeinsamen Nutzung der CPU-Ressourcen mit eingeschlossen ist.

Katalogsichten

SYSCAT.HISTOGRAMTEMPLATEBINS

Jede Zeile stellt einen Histogrammschablonenbin dar.

Tabelle 266. SYSCAT.HISTOGRAMTEMPLATEBINS (Katalogsicht)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
TEMPLATENAME	VARCHAR(128)	Y	Name der Histogrammschablone.
TEMPLATEID	INTEGER		ID der Histogrammschablone.

Tabelle 266. SYSCAT.HISTOGRAMTEMPLATEBINS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
BINID	INTEGER		ID des Histogrammschablonenbins.
BINUPPERVALUE	BIGINT		Der obere Wert eines einzelnen Bins in der Histogrammschablone.

SYSCAT.HISTOGRAMTEMPLATES

Jede Zeile stellt eine Histogrammschablone dar.

Tabelle 267. SYSCAT.HISTOGRAMTEMPLATES (Katalogsicht)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
TEMPLATEID	INTEGER		ID der Histogrammschablone.
TEMPLATENAME	VARCHAR(128)		Name der Histogrammschablone.
CREATE_TIME	TIMESTAMP		Zeit, zu der die Histogrammschablone erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der die Histogrammschablone zuletzt geändert wurde.
NUMBINS	INTEGER		Anzahl der Bins in der Histogrammschablone, einschließlich des letzten Bins, das einen unbegrenzten oberen Wert hat.
REMARKS	VARCHAR (254)	Y	Vom Benutzer bereitgestellte Kommentare oder der Nullwert.

SYSCAT.HISTOGRAMTEMPLATEUSE

Jede Zeile stellt eine Beziehung zwischen einem Workload-Management-Objekt, das Histogrammschablonen verwenden kann, und einer Histogrammschablone dar.

Tabelle 268. SYSCAT.HISTOGRAMTEMPLATEUSE(Katalogsicht)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
TEMPLATENAME	VARCHAR(128)	Y	Name der Histogrammschablone.
TEMPLATEID	INTEGER		ID der Histogrammschablone.

Tabelle 268. SYSCAT.HISTOGRAMTEMPLATEUSE(Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
HISTOGRAMTYPE	CHAR (1)		Der Typ der Informationen, die von Histogrammen erfasst werden, die auf dieser Schablone basieren. <ul style="list-style-type: none"> • C = Histogramm des geschätzten Aufwands für Aktivitäten • E = Histogramm der Aktivitätsausführungszeit • I = Histogramm der Zwischenankunftszeit von Aktivitäten • L = Histogramm der Aktivitätslaufzeit • Q = Histogramm der Aktivitätswartezeit • R = Histogramm der Anforderungsausführungszeit • U = Histogramm der UOW-Lebensdauer
OBJECTTYPE	CHAR (1)		Der Typ des WLM-Objekts. <ul style="list-style-type: none"> • b = Serviceklasse • k = Arbeitsaktion • w = Workload
OBJECTID	INTEGER		ID des WLM-Objekts.
SERVICECLASSNAME	VARCHAR(128)	Y	Name der Serviceklasse.
PARENTSERVICECLASSNAME	VARCHAR(128)	Y	Der Name der übergeordneten Serviceklasse der Serviceunterklasse, welche die Histogrammschablone verwendet.
WORKACTIONNAME	VARCHAR(128)	Y	Der Name der Arbeitsaktion, welche die Histogrammschablone verwendet.
WORKACTIONSETNAME	VARCHAR(128)	Y	Der Name des Arbeitsaktionssets, welches die Histogrammschablone verwendet.
WORKLOADNAME	VARCHAR(128)	Y	Der Name der Workload, welche die Histogrammschablone verwendet.

SYSCAT.SERVICECLASSES

Jede Zeile stellt eine Serviceklasse dar.

Tabelle 269. SYSCAT.SERVICECLASSES (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
SERVICECLASSNAME	VARCHAR(128)		Name der Serviceklasse.
PARENTSERVICECLASSNAME	VARCHAR(128)	Y	Serviceklassenname der übergeordneten Servicesuperklasse.
SERVICECLASSID	SMALLINT		Kennung für die Serviceklasse.
PARENTID	SMALLINT		Kennung der übergeordneten Serviceklasse für diese Serviceklasse. 0, wenn diese Serviceklasse eine Servicesuperklasse ist.
CREATE_TIME	TIMESTAMP		Zeit, zu der die Serviceklasse erstellt wurde.

Tabelle 269. SYSCAT.SERVICECLASSES (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
ALTER_TIME	TIMESTAMP		Zeit, zu der die Serviceklasse zuletzt geändert wurde.
ENABLED	CHAR (1)		Status der Serviceklasse. <ul style="list-style-type: none"> • N = Inaktiviert • Y = Aktiviert
AGENTPRIORITY	SMALLINT		Threadpriorität der Agenten in der Serviceklasse relativ zur normalen Priorität der DB2-Threads. <ul style="list-style-type: none"> • -20 bis 20 (Linux und UNIX) • -6 bis 6 (Windows) • -32768 = nicht festgelegt
PREFETCHPRIORITY	CHAR (1)		Vorablesepriorität der Agenten in der Serviceklasse. <ul style="list-style-type: none"> • H = Hoch • L = Niedrig • M = Mittel • Leer = nicht festgelegt
MAXDEGREE	SMALLINT	Y	Zur zukünftigen Verwendung reserviert.
BUFFERPOOLPRIORITY	CHAR (1)		Pufferpoolpriorität der Agenten in der Serviceklasse. <ul style="list-style-type: none"> • H = Hoch • L = Niedrig • M = Mittel • Leer = nicht festgelegt
INBOUNDCORRELATOR	VARCHAR(128)	Y	Zur zukünftigen Verwendung.
OUTBOUNDCORRELATOR	VARCHAR(128)	Y	Zeichenfolge, mit der die Serviceklasse einer Workload-Manager-Serviceklasse des Betriebssystems zugeordnet wird.
COLLECTAGGACTDATA	CHAR (1)		Gibt an, welche zusammengefassten Aktivitätsdaten der entsprechende Ereignismonitor für die Serviceklasse erfassen soll. <ul style="list-style-type: none"> • B = Zusammengefasste Basisaktivitätsdaten erfassen • E = Erweiterte zusammengefasste Aktivitätsdaten erfassen • N = Keine
COLLECTAGGREQDATA	CHAR (1)		Gibt an, welche zusammengefassten Anforderungsdaten der entsprechende Ereignismonitor für die Serviceklasse erfassen soll. <ul style="list-style-type: none"> • B = Zusammengefasste Basisanforderungsdaten erfassen • N = Keine

Tabelle 269. SYSCAT.SERVICECLASSES (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
COLLECTACTDATA	CHAR (1)		Gibt an, welche Aktivitätsdaten der entsprechende Ereignismonitor erfassen soll. <ul style="list-style-type: none"> • D = Aktivitätsdaten mit Details • N = Keine • S = Aktivitätsdaten mit Details und Abschnittsumgebung • V = Aktivitätsdaten mit Details und Werten • W = Aktivitätsdaten ohne Details • X = Aktivitätsdaten mit Details, Abschnittsumgebung und Werten
COLLECTACTPARTITION	CHAR (1)		Gibt an, wo Aktivitätsdaten erfasst werden. <ul style="list-style-type: none"> • C = Koordinatormember der Aktivität • D = Alle Member
COLLECTREQMETRICS	CHAR (1)		Gibt die Überwachungsstufe für Anforderungen an, die von einer der Servicesuperklasse zugeordneten Verbindung übergeben wurden. <ul style="list-style-type: none"> • B = Basisanforderungsmessdaten erfassen • E = Erweiterte Anforderungsmessdaten erfassen • N = Keine
CPUSHARES	INTEGER		Die Anzahl der CPU-Anteile, die dieser Serviceklasse zugeordnet sind.
CPUSHARETYPE	CHAR (1)		Gibt den Typ der CPU-Anteile an. <ul style="list-style-type: none"> • S = Veränderbare Anteile • H = Feste Anteile
CPULIMIT	SMALLINT		Der maximale Prozentsatz der CPU-Ressource, der der Serviceklasse zugeordnet werden kann; -1, wenn kein CPU-Grenzwert vorhanden ist.
SORTMEMORYPRIORITY	CHAR (1)		Zur zukünftigen Verwendung reserviert.
SECTIONACTUALSOPTIONS	VARCHAR (32)		Gibt an, welche Abschnitts-Ist-Daten während der Ausführung eines Abschnitts erfasst werden. Die erste Position in der Zeichenfolge gibt an, ob die Erfassung der Abschnitts-Ist-Daten aktiviert ist. <ul style="list-style-type: none"> • B = Aktiviert und erfasst Bedienerkardinalitätszähler und Statistikdaten für jedes Objekt, auf das der Abschnitt verweist (nur DML-Anweisungen). • N = Nicht aktiviert. Die zweite Position ist immer 'N' und ist für eine zukünftige Verwendung reserviert.

Tabelle 269. SYSCAT.SERVICECLASSES (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
COLLECTAGGUOWDATA	CHAR (1)		Gibt an, welche zusammengefassten UOW-Daten der entsprechende Ereignismonitor für die Serviceklasse erfassen soll. <ul style="list-style-type: none"> • B = Zusammengefasste UOW-Basisdaten erfassen • N = Keine
REMARKS	VARCHAR (254)	Y	Vom Benutzer bereitgestellte Kommentare oder der Nullwert.

SYSCAT.THRESHOLDS

Jede Zeile stellt einen Schwellenwert dar.

Tabelle 270. SYSCAT.THRESHOLDS (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
THRESHOLDNAME	VARCHAR(128)		Name des Schwellenwerts.
THRESHOLDID	INTEGER		Kennung für den Schwellenwert.
ORIGIN	CHAR (1)		Ursprung des Schwellenwerts. <ul style="list-style-type: none"> • U = Der Schwellenwert wurde von einem Benutzer erstellt • W = Der Schwellenwert wurde während eines Arbeitsaktionssets erstellt
THRESHOLDCLASS	CHAR (1)		Klassifikation des Schwellenwerts. <ul style="list-style-type: none"> • A = Kumulativer Schwellenwert • C = Aktivitätsschwellenwert

Tabelle 270. SYSCAT.THRESHOLDS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
THRESHOLDPREDICATE	VARCHAR (15)		Typ des Schwellenwerts. Mögliche Werte: <ul style="list-style-type: none"> • AGGTEMPSPACE • CONCDBC • CONCWCN • CONCWOC • CONNIDLETIME • CPUTIME • CPUTIMEINSC • DATATAGINSC • DATATAGNOTINSC • DBCONN • ESTSQLCOST • ROWSREAD • ROWSREADINSC • ROWSRET • SCCONN • TEMPSPACE • TOTALTIME • UOWTOTALTIME
THRESHOLDPREDICATEID	SMALLINT		Kennung für das Schwellenwertvergleichselement.
DOMAIN	CHAR(2)		Domäne des Schwellenwerts. <ul style="list-style-type: none"> • DB = Datenbank • SB = Serviceunterklasse • SP = Servicesuperklasse • WA = Arbeitsaktionsset • WD = Workloaddefinition • SQ = SQL-Anweisung
DOMAINID	INTEGER		Die Kennung des Objekts, dem der Schwellenwert zugeordnet ist. Dies kann eine Serviceklasse, Arbeitsaktion, eindeutige Workload-ID oder eine SQL-Anweisung sein. Ist dies ein Datenbankschwellenwert, ist der Wert 0.
ENFORCEMENT	CHAR (1)		Umsetzungsbereich des Schwellenwerts. <ul style="list-style-type: none"> • D = Datenbank • P = Member • W = Workloadvorkommen
QUEUING	CHAR (1)		<ul style="list-style-type: none"> • N = Der Grenzwert wird nicht in die Warteschlange gestellt • Y = Der Grenzwert wird in die Warteschlange gestellt

Tabelle 270. SYSCAT.THRESHOLDS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
MAXVALUE	BIGINT		Vom Schwellenwert angegebene Obergrenze. Wenn THRESHOLDPREDICATE auf 'DATATAGINSC' oder 'DATATAGNOTINSC' gesetzt ist, codiert dieser Wert mindestens eubeb Datentag.
DATATAGLIST	VARCHAR(256)	Y	Wenn THRESHOLDPREDICATE auf 'DATATAGINSC' oder 'DATATAGNOTINSC' gesetzt ist, stellt dieser Wert mindestens einen Wert als durch Kommas getrennte Liste dar. Ansonsten ist der Wert ein Nullwert.
QUEUE SIZE	INTEGER		Wenn QUEUEING auf 'Y' gesetzt ist, ist dies die Größe der Warteschlange. Ansonsten -1.
OVERFLOWPERCENT	SMALLINT		Zur zukünftigen Verwendung reserviert.
COLLECTACTDATA	CHAR (1)		Gibt an, welche Aktivitätsdaten der entsprechende Ereignismonitor erfassen soll. <ul style="list-style-type: none"> • D = Aktivitätsdaten mit Details • N = Keine • S = Aktivitätsdaten mit Details und Abschnittsumgebung • V = Aktivitätsdaten mit Details und Werten • W = Aktivitätsdaten ohne Details • X = Aktivitätsdaten mit Details, Abschnittsumgebung und Werten
COLLECTACTPARTITION	CHAR (1)		Gibt an, wo Aktivitätsdaten erfasst werden. <ul style="list-style-type: none"> • C = Koordinatormember der Aktivität • D = Alle Member
EXECUTION	CHAR (1)		Gibt die Aktion an, die nach der Überschreitung eines Schwellenwerts ausgeführt wird. <ul style="list-style-type: none"> • C = Ausführung wird fortgesetzt • F = Anwendung wird zwangsweise vom System abgemeldet • R = Ausführung wird einer anderen Serviceunterklasse neu zugeordnet • S = Ausführung wird gestoppt
REMAPSCID	SMALLINT		Zielserviceunterklassen-ID der Aktion REMAP ACTIVITY.
VIOLATIONRECORDLOGGED	CHAR (1)		Gibt an, ob bei einem Schwellenwertverstoß ein Datensatz in den Ereignismonitor geschrieben wird. <ul style="list-style-type: none"> • N = Nein • Y = Ja

Tabelle 270. SYSCAT.THRESHOLDS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
CHECKINTERVAL	INTEGER		Das Intervall in Sekunden, in dem die Schwellenwertbedingung überprüft wird, wenn THRESHOLDPREDICATE auf einen der folgenden Werte gesetzt ist: <ul style="list-style-type: none"> • 'CPUTIME' • 'CPUTIMEINSC' • 'ROWSREAD' • 'ROWSREADINSC' Ansonsten -1.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> • N = Dieser Schwellenwert ist inaktiviert. • Y = Dieser Schwellenwert ist aktiviert.
CREATE_TIME	TIMESTAMP		Zeit, zu der der Schwellenwert erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der der Schwellenwert zuletzt geändert wurde.
REMARKS	VARCHAR (254)	Y	Vom Benutzer bereitgestellte Kommentare oder der Nullwert.

SYSCAT.WORKACTIONS

Jede Zeile stellt eine Arbeitsaktion dar, die für ein Arbeitsaktionsset definiert ist.

Tabelle 271. SYSCAT.WORKACTIONS (Katalogsicht)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
ACTIONNAME	VARCHAR(128)		Name der Arbeitsaktion.
ACTIONID	INTEGER		Kennung für die Arbeitsaktion.
ACTIONSETNAME	VARCHAR(128)	Y	Name des Arbeitsaktionssets.
ACTIONSETID	INTEGER		Die ID des Arbeitsaktionssets, zu dem diese Arbeitsaktion gehört. Diese Spalte bezieht sich auf die Spalte ACTIONSETID in der Sicht SYSCAT.WORKACTIONSETS.
WORKCLASSNAME	VARCHAR(128)	Y	Name der Arbeitsklasse.
WORKCLASSID	INTEGER		Kennung der Arbeitsklasse. Diese Spalte bezieht sich auf die Spalte WORKCLASSID in der Sicht SYSCAT.WORKCLASSES.
CREATE_TIME	TIMESTAMP		Zeit, zu der die Arbeitsaktion erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der die Arbeitsaktion zuletzt geändert wurde.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> • N = Diese Arbeitsaktion ist inaktiviert. • Y = Diese Arbeitsaktion ist aktiviert.

Tabelle 271. SYSCAT.WORKACTIONS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Daten- eingabe optional	Beschreibung
ACTIONTYPE	CHAR (1)		<p>Der Typ der Aktion, die für jede DB2-Aktivität ausgeführt wird, die mit den Attributen in der Arbeitsklasse innerhalb des Bereichs übereinstimmt.</p> <ul style="list-style-type: none"> • B = Zusammengefasste Basisaktivitätsdaten erfassen; kann nur für Arbeitsaktionssets angegeben werden, die auf Serviceklassen oder Workloads angewendet werden. • C = Jeder DB2-Aktivität, die unter der zugehörigen Arbeitsklasse ausgeführt wird, die Ausführung erlauben und den Arbeitsklassenzähler erhöhen. • D = Im Koordinationsmember der Aktivität Aktivitätsdaten mit Details erfassen. • E = Erweiterte zusammengefasste Aktivitätsdaten erfassen; kann nur für Arbeitsaktionssets angegeben werden, die auf Serviceklassen oder Workloads angewendet werden. • F = Im Koordinationsmember der Aktivität Aktivitätsdaten mit Details, Abschnitt und Werten erfassen. • G = Im Koordinationsmember der Aktivität Aktivitätsdetails und Abschnitt erfassen und in allen Mitgliedern Aktivitätsdaten erfassen. • H = Im Koordinationsmember der Aktivität Aktivitätsdetails, Abschnitt und Werte erfassen und in allen Mitgliedern Aktivitätsdaten erfassen. • M = Einer Serviceunterklasse zuordnen; kann nur für Arbeitsaktionssets angegeben werden, die auf Serviceklassen angewendet werden. • P = Die Ausführung aller DB2-Aktivitäten unter der Arbeitsklasse verhindern, der diese Arbeitsaktion zugeordnet ist. • S = Im Koordinationsmember der Aktivität Aktivitätsdaten mit Details und Abschnitt erfassen. • T = Die Aktion stellt einen Schwellenwert dar; kann nur für Arbeitsaktionssets angegeben werden, die einer Datenbank oder einer Workload zugeordnet sind. • U = Alle Aktivitäten mit einer Verschachtelungsebene Null und alle unter diesen Aktivitäten verschachtelten Aktivitäten einer Serviceunterklasse zuordnen; kann nur für Arbeitsaktionssets angegeben werden, die auf Serviceklassen angewendet werden. • V = Im Koordinationsmember Aktivitätsdaten mit Details und Werten erfassen. • W = Im Koordinationsmember Aktivitätsdaten ohne Details erfassen. • X = Im Koordinationsmember Aktivitätsdaten mit Details erfassen und in allen Mitgliedern Aktivitätsdaten erfassen. • Y = Im Koordinationsmember Aktivitätsdaten mit Details und Werten erfassen und in allen Mitgliedern Aktivitätsdaten erfassen. • Z = In allen Mitgliedern Aktivitätsdaten ohne Details erfassen.

Tabelle 271. SYSCAT.WORKACTIONS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
REFOBJECTID	INTEGER	Y	Wenn ACTIONTYPE auf 'M' (zuordnen) oder 'N' (verschachtelte zuordnen) gesetzt ist, wird dieser Wert auf die ID der Serviceunterklasse gesetzt, der die DB2-Aktivität zugeordnet ist. Wenn ACTIONTYPE auf 'T' (Schwellenwert) gesetzt ist, wird dieser Wert auf die ID des zu verwendenden Schwellenwerts gesetzt. Für alle anderen Aktionen ist dieser Wert NULL.
REFOBJECTTYPE	VARCHAR(30)		Wenn der ACTIONTYPE 'M' oder 'N' ist, wird dieser Wert auf 'SERVICE CLASS' gesetzt; wenn ACTIONTYPE auf 'T' gesetzt ist, lautet dieser Wert 'THRESHOLD'; ansonsten ist dieser Wert NULL.
SECTIONACTUALSOPTIONS	VARCHAR (32)		Gibt an, welche Abschnitts-Ist-Daten während der Ausführung eines Abschnitts erfasst werden. Die erste Position in der Zeichenfolge gibt an, ob die Erfassung der Abschnitts-Ist-Daten aktiviert ist. <ul style="list-style-type: none"> • B = Aktiviert und erfasst Bedienerkardinalitätszähler und Statistikdaten für jedes Objekt, auf das der Abschnitt verweist (nur DML-Anweisungen). • N = Nicht aktiviert. Die zweite Position ist immer 'N' und ist für eine zukünftige Verwendung reserviert.

SYSCAT.WORKACTIONSETS

Jede Zeile stellt ein Arbeitsaktionsset dar.

Tabelle 272. SYSCAT.WORKACTIONSETS (Katalogsicht)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
ACTIONSETNAME	VARCHAR(128)		Name des Arbeitsaktionssets.
ACTIONSETID	INTEGER		Kennung für das Arbeitsaktionsset.
WORKCLASSETNAME	VARCHAR(128)	Y	Name des Arbeitsklassensets.
WORKCLASSETID	INTEGER		Die ID des Arbeitsklassensets, das dem von OBJECTID angegebenen Objekt zugeordnet werden soll. Diese Spalte bezieht sich auf die Spalte WORKCLASSETID in der Sicht SYSCAT.WORKCLASSETS.
CREATE_TIME	TIMESTAMP		Zeit, zu der das Arbeitsaktionsset erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der das Arbeitsaktionsset zuletzt geändert wurde.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> • N = Dieses Arbeitsaktionsset ist inaktiviert. • Y = Dieses Arbeitsaktionsset ist aktiviert.

Tabelle 272. SYSCAT.WORKACTIONSETS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
OBJECTTYPE	CHAR (1)		<ul style="list-style-type: none"> • b = Servicesuperklasse • w = Workload • Leer = Datenbank
OBJECTNAME	VARCHAR(128)	Y	Name der Serviceklasse oder der Workload.
OBJECTID	INTEGER		Die ID des Objekts, dem das (von WORKCLASSESETID angegebene) Arbeitsklassenset zugeordnet wird. Wenn OBJECTTYPE auf 'b' gesetzt ist, ist OBJECTID die ID der Servicesuperklasse. Wenn OBJECTTYPE auf 'w' gesetzt ist, ist OBJECTID die ID der Workload. Wenn OBJECTTYPE leer ist, ist OBJECTID -1.
REMARKS	VARCHAR (254)	Y	Vom Benutzer bereitgestellte Kommentare oder der Nullwert.

SYSCAT.WORKCLASSES

Jede Zeile stellt eine Arbeitsklasse dar, die für ein Arbeitsklassenset definiert wurde.

Tabelle 273. SYSCAT.WORKCLASSES (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
WORKCLASSNAME	VARCHAR(128)		Name der Arbeitsklasse.
WORKCLASSESETNAME	VARCHAR(128)	Y	Name des Arbeitsklassensets.
WORKCLASSID	INTEGER		Kennung für die Arbeitsklasse.
WORKCLASSESETID	INTEGER		ID des Arbeitsklassensets, zu dem diese Arbeitsklasse gehört. Diese Spalte bezieht sich auf die Spalte WORKCLASSESETID in der Sicht SYSCAT.WORKCLASSESETS.
CREATE_TIME	TIMESTAMP		Zeit, zu der die Arbeitsklasse erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der die Arbeitsklasse zuletzt geändert wurde.
EVALUATIONORDER	SMALLINT		Eindeutige ID der Bewertungsreihenfolge, die zum Auswählen einer Arbeitsklasse in einem Arbeitsklassenset verwendet wird.

SYSCAT.WORKCLASSESETS

Jede Zeile stellt ein Arbeitsklassenset dar.

Tabelle 274. SYSCAT.WORKCLASSESETS (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
WORKCLASSESETNAME	VARCHAR(128)		Name des Arbeitsklassensets.

Tabelle 274. SYSCAT.WORKCLASSESETS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
WORKCLASSETID	INTEGER		Kennung für das Arbeitsklassenset.
CREATE_TIME	TIMESTAMP		Zeit, zu der das Arbeitsklassenset erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der das Arbeitsklassenset zuletzt geändert wurde.
REMARKS	VARCHAR (254)	Y	Vom Benutzer bereitgestellte Kommentare oder der Nullwert.

SYSCAT.WORKLOADAUTH

Jede Zeile stellt einen Benutzer, eine Gruppe oder eine Rolle dar, dem bzw. der das Zugriffsrecht USAGE für eine Workload erteilt wurde.

Tabelle 275. SYSCAT.WORKLOADAUTH (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
WORKLOADID	INTEGER		Kennung für die Workload.
WORKLOADNAME	VARCHAR(128)		Der Name der Workload.
GRANTOR	VARCHAR(128)		Berechtigungsgeber des Zugriffsrechts.
GRANTORTYPE	CHAR (1)		<ul style="list-style-type: none"> • U = Berechtigter ist ein einzelner Benutzer
GRANTEE	VARCHAR(128)		Inhaber des Zugriffsrechts.
GRANTEETYPE	CHAR (1)		<ul style="list-style-type: none"> • G = Berechtigter ist eine Gruppe • R = Berechtigter ist eine Rolle • U = Berechtigter ist ein einzelner Benutzer
USAGEAUTH	CHAR (1)		<p>Gibt an, ob der Berechtigte über das Zugriffsrecht USAGE für die Workload verfügt.</p> <ul style="list-style-type: none"> • N = Verfügt nicht darüber • Y = Verfügt darüber

SYSCAT.WORKLOADCONNATTR

Jede Zeile stellt ein Verbindungsattribut in der Definition einer Workload dar.

Tabelle 276. SYSCAT.WORKLOADCONNATTR (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
WORKLOADID	INTEGER		Kennung für die Workload.
WORKLOADNAME	VARCHAR(128)		Der Name der Workload.

Tabelle 276. SYSCAT.WORKLOADCONNATTR (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
CONNATTRTYPE	VARCHAR(30)		Typ des Verbindungsattributs. <ul style="list-style-type: none"> • 1 = APPLNAME • 2 = SYSTEM_USER • 3 = SESSION_USER • 4 = SESSION_USER GROUP • 5 = SESSION_USER ROLE • 6 = CURRENT CLIENT_USERID • 7 = CURRENT CLIENT_APPLNAME • 8 = CURRENT CLIENT_WRKSTNNAME • 9 = CURRENT CLIENT_ACCTNG • 10 = ADDRESS
CONNATTRVALUE	VARCHAR (1000)		Wert des Verbindungsattributs.

SYSCAT.WORKLOADS

Jede Zeile stellt eine Workload dar.

Tabelle 277. SYSCAT.WORKLOADS (Katalogsicht)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
WORKLOADID	INTEGER		Kennung für die Workload.
WORKLOADNAME	VARCHAR(128)		Der Name der Workload.
EVALUATIONORDER	SMALLINT		Zur Auswahl einer Workload verwendete Bewertungsreihenfolge.
CREATE_TIME	TIMESTAMP		Zeit, zu der die Workload erstellt wurde.
ALTER_TIME	TIMESTAMP		Zeit, zu der die Workload zuletzt geändert wurde.
ENABLED	CHAR (1)		<ul style="list-style-type: none"> • N = Diese Workload ist inaktiviert. • Y = Diese Workload ist aktiviert.
ALLOWACCESS	CHAR (1)		<ul style="list-style-type: none"> • N = Eine dieser Workload zugeordnete UOW wird zurückgewiesen. • Y = Eine dieser Workload zugeordnete UOW (Unit of Work) kann auf die Datenbank zugreifen.
MAXDEGREE	SMALLINT		Maximaler Grad der Parallelität für die Workload. Gültige Werte: 1 bis 32767 und -1. Wenn MAXIMUM DEGREE auf DEFAULT gesetzt ist, ist der Wert -1.
SERVICECLASSNAME	VARCHAR(128)		Name der Serviceunterklasse, der eine (dieser Workload zugeordnete) UOW zugewiesen wird.
PARENTSERVICECLASSNAME	VARCHAR(128)	Y	Name der Servicesuperklasse, der eine (dieser Workload zugeordnete) UOW zugewiesen wird.

Tabelle 277. SYSCAT.WORKLOADS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
COLLECTAGGACTDATA	CHAR (1)		<p>Gibt an, welche zusammengefassten Aktivitätsdaten der entsprechende Ereignismonitor für die Workload erfassen soll.</p> <ul style="list-style-type: none"> • B = Zusammengefasste Basisaktivitätsdaten erfassen • E = Erweiterte zusammengefasste Aktivitätsdaten erfassen • N = Keine
COLLECTACTDATA	CHAR (1)		<p>Gibt an, welche Aktivitätsdaten der entsprechende Ereignismonitor erfassen soll.</p> <ul style="list-style-type: none"> • D = Aktivitätsdaten mit Details • N = Keine • S = Aktivitätsdaten mit Details und Abschnittsumgebung • V = Aktivitätsdaten mit Details und Werten. Gültig, wenn die Spalte COLLECT auf 'C' gesetzt ist. • W = Aktivitätsdaten ohne Details • X = Aktivitätsdaten mit Details, Abschnittsumgebung und Werten
COLLECTACTPARTITION	CHAR (1)		<p>Gibt an, wo Aktivitätsdaten erfasst werden.</p> <ul style="list-style-type: none"> • C = Koordinator member der Aktivität • D = Alle Member
COLLECTDEADLOCK	CHAR (1)		<p>Gibt an, dass der entsprechende Ereignismonitor Deadlock-Ereignisse erfassen soll.</p> <ul style="list-style-type: none"> • H = Nur Deadlock-Daten mit zurückliegenden Aktivitäten erfassen • V = Deadlock-Daten mit zurückliegenden Aktivitäten und Werten erfassen • W = Deadlock-Daten ohne zurückliegende Aktivitäten und Werte erfassen
COLLECTLOCKTIMEOUT	CHAR (1)		<p>Gibt an, dass der entsprechende Ereignismonitor Ereignisse mit Zeitlimitüberschreitung für Sperren erfassen soll.</p> <ul style="list-style-type: none"> • H = Nur Daten zur Überschreitung der Sperrzeit mit zurückliegenden Aktivitäten erfassen • N = Daten zur Überschreitung der Sperrzeit nicht erfassen • V = Daten zur Überschreitung der Sperrzeit mit zurückliegenden Aktivitäten und Werten erfassen • W = Daten zur Überschreitung der Sperrzeit ohne zurückliegende Aktivitäten und Werte erfassen

Tabelle 277. SYSCAT.WORKLOADS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Dateneingabe optional	Beschreibung
COLLECTLOCKWAIT	CHAR (1)		<p>Gibt an, dass der entsprechende Ereignismonitor Ereignisse mit Wartestatus für Sperren erfassen soll.</p> <ul style="list-style-type: none"> • H = Nur Daten zu Wartestatus für Sperren mit zurückliegenden Aktivitäten erfassen • N = Daten zu Wartestatus für Sperren nicht erfassen • V = Daten zu Wartestatus für Sperren mit zurückliegenden Aktivitäten und Werten erfassen • W = Daten zu Wartestatus für Sperren ohne zurückliegende Aktivitäten und Werte erfassen
LOCKWAITVALUE	INTEGER		<p>Gibt die Wartezeit in Millisekunden für eine Sperre an, bevor der entsprechende Ereignismonitor ein Sperrereignis erfasst; 0, wenn COLLECTLOCKWAIT = 'N'</p>
COLLECTACTMETRICS	CHAR (1)		<p>Gibt die Überwachungsstufe für Aktivitäten an, die von einem Vorkommen der Workload übergeben wurden.</p> <ul style="list-style-type: none"> • B = Basisaktivitätsmessdaten erfassen • E = Erweiterte Aktivitätsmessdaten erfassen • N = Keine
COLLECTUOWDATAOPTIONS	VARCHAR (32)		<p>Gibt an, welche UOW-Daten vom entsprechenden Ereignismonitor erfasst werden. Die erste Position in der Zeichenfolge gibt an, ob die Erfassung der UOW-Daten aktiviert ist.</p> <ul style="list-style-type: none"> • B = Aktiviert und erfasst UOW-Basisdaten • N = Nicht aktiviert <p>Ab der zweiten Position in der Zeichenfolge gibt jede Position jeweils eine bestimmte erweiterte Option an:</p> <ul style="list-style-type: none"> • 2 = Paketreferenzliste • 3 = Liste der ausführbaren IDs <p>Jede Position, die eine erweiterte Option darstellt, wird dann auf einen der folgenden Werte gesetzt:</p> <ul style="list-style-type: none"> • Y = Erweiterte Option ist eingeschlossen • N = Erweiterte Option ist nicht eingeschlossen

Tabelle 277. SYSCAT.WORKLOADS (Katalogsicht) (Forts.)

Spaltenname	Datentyp	Datenein- gabe opti- onal	Beschreibung
COLLECTUOWDATA	CHAR (1)		<p>Gibt an, welche UOW-Daten vom entsprechenden Ereignismonitor erfasst werden sollen.</p> <ul style="list-style-type: none"> • B = UOW-Basisdaten erfassen • N = Keine • P = UOW-Basisdaten und die Paketliste erfassen <p>Diese Spalte wird nicht weiter unterstützt. Informationen zu der Spalte sind in COLLECTUOWDATAOPTIONS verfügbar.</p>
EXTERNALNAME	VARCHAR(128)	Y	Zur zukünftigen Verwendung reserviert.
SECTIONACTUALSOPTIONS	VARCHAR (32)		<p>Gibt an, welche Abschnitts-Ist-Daten während der Ausführung eines Abschnitts erfasst werden.</p> <p>Die erste Position in der Zeichenfolge gibt an, ob die Erfassung der Abschnitts-Ist-Daten aktiviert ist.</p> <ul style="list-style-type: none"> • B = Aktiviert und erfasst Bedienerkardinalitätszähler und Statistikdaten für jedes Objekt, auf das der Abschnitt verweist (nur DML-Anweisungen). • N = Nicht aktiviert. <p>Die zweite Position ist immer 'N' und ist für eine zukünftige Verwendung reserviert.</p>
COLLECTAGGUOWDATA	CHAR (1)		<p>Gibt an, welche zusammengefassten UOW-Daten der entsprechende Ereignismonitor für die Workload erfassen soll.</p> <ul style="list-style-type: none"> • B = Zusammengefasste UOW-Basisdaten erfassen • N = Keine
REMARKS	VARCHAR (254)	Y	Vom Benutzer bereitgestellte Kommentare oder der Nullwert.

Anhang A. Allgemeine Namenskonventionen

Für die Benennung aller Datenbankobjekte, Benutzernamen, Kennwörter, Gruppen, Dateien und Pfade gelten Regeln. Einige dieser Regeln sind für die Plattform spezifisch, auf der Sie arbeiten.

Die folgenden Punkte sind zum Beispiel in Bezug auf die Verwendung von Groß- und Kleinbuchstaben in den Namen von Objekten zu beachten, die im Dateisystem (Datenbanken, Instanzen usw.) sichtbar sind:

- Auf UNIX-Plattformen wird bei Namen die Groß-/Kleinschreibung unterschieden. Zum Beispiel ist ein Verzeichnis `/data1` nicht mit den Verzeichnissen `/DATA1` oder `/Data1` identisch.
- Auf Windows-Plattformen wird die Groß-/Kleinschreibung in Namen nicht unterschieden. Zum Beispiel bezeichnet `\data1` denselben Namen wie `\DATA1` und `\Data1`.

Wenn nichts anderes angegeben ist, dürfen alle Namen die folgenden Zeichen enthalten:

- Die Buchstaben A - Z und a bis z, wie sie im ASCII-Grundzeichensatz (7-Bit-Zeichensatz) definiert sind. Wenn sie in Kennungen für Objekte verwendet werden, die durch SQL-Anweisungen erstellt werden, werden die Kleinbuchstaben "a" bis "z" in die entsprechenden Großbuchstaben umgewandelt, wenn sie nicht in Anführungszeichen (") gesetzt werden.
- 0 bis 9.
- ! % () { } . - ^ ~ _ (Unterstreichungszeichen), @, #, \$ und Leerzeichen.
- \ (umgekehrter Schrägstrich oder Backslash).

Einschränkungen

- Namen dürfen nicht mit einer Ziffer oder dem Unterstreichungszeichen beginnen.
- Für SQL reservierte Wörter dürfen nicht als Namen von Tabellen, Sichten, Spalten, Indizes oder Berechtigungs-IDs verwendet werden.
- In Namen für Verzeichnis- oder Dateinamen sollten nur die Buchstaben verwendet werden, die im ASCII-Grundzeichensatz definiert sind. Das Betriebssystem Ihres Computers unterstützt vielleicht verschiedene Codepages, jedoch ist es möglich, dass andere Zeichen als ASCII-Zeichen nicht zuverlässig funktionieren. Die Verwendung von Nicht-ASCII-Zeichen kann insbesondere in einer verteilten Umgebung, in der verschiedene Computer unterschiedliche Codepages verwenden, zu einem Problem werden.
- In Abhängigkeit vom verwendeten Betriebssystem und dem Ort, an dem Sie mit der DB2-Datenbank arbeiten, sind möglicherweise weitere Sonderzeichen verfügbar, die ebenfalls verwendet werden können. Diese Zeichen können funktionieren, jedoch gibt es keine Garantie, dass sie funktionieren. Daher wird empfohlen, beim Benennen von Objekten in der Datenbank keine anderen als die oben aufgelisteten Sonderzeichen zu verwenden.
- Benutzer- und Gruppennamen müssen auch die Regeln einhalten, die durch bestimmte Betriebssysteme festgelegt werden. Zum Beispiel sind auf Linux- und UNIX-Plattformen nur die folgenden Zeichen für Benutzernamen und Gruppennamen zulässig: a bis z in Kleinbuchstaben, 0 bis 9 und _ (Unterstreichungszeichen) für Namen, die nicht mit einer Ziffer zwischen 0 und 9 beginnen.

- Längen dürfen die in „SQL- und XML-Begrenzungen“ (SQL and XML limits) im Handbuch *SQL Reference* aufgeführten Längen nicht überschreiten.
- **Einschränkungen zur Berechtigungs-ID AUTHID:** In DB2 Version 9.5 und höher kann die Berechtigungs-ID eine Größe von 128 Byte haben. Wenn die Berechtigungs-ID jedoch als Benutzer-ID oder Gruppenname des Betriebssystems interpretiert wird, gelten die Einschränkungen des Betriebssystems. Benutzer-IDs auf Linux- und UNIX-Betriebssystemen dürfen bis zu 8 Zeichen enthalten. Benutzer-IDs auf Windows-Betriebssystemen dürfen bis zu 30 Zeichen für Benutzer-IDs und Gruppennamen enthalten. Daher können Sie zwar eine Berechtigungs-ID mit einer Größe von 128-Byte erteilen, jedoch ist es nicht möglich, als Benutzer, der diese Berechtigungs-ID hat, eine Verbindung herzustellen. Wenn Sie ein eigenes Sicherheits-Plug-in schreiben, können Sie die erweiterten Größen für die Berechtigungs-ID verwenden. Zum Beispiel können Sie Ihr Sicherheits-Plug-in mit einer Benutzer-ID mit einer Größe von 30-Byte versehen, und das Plug-in gibt während der Authentifizierung eine 128-Byte-Berechtigungs-ID zurück, mit der Sie eine Verbindung herstellen können.

Darüber hinaus müssen Sie auch die Namenskonventionen der einzelnen Objekte, die Namenskonventionen in einer für einen kulturübergreifenden Einsatz geeigneten Umgebung und die Namenskonventionen in einer Unicode-Umgebung beachten.

Anhang B. Rollen

Rollen vereinfachen die Verwaltung von Zugriffsrechten durch die Gruppierung von Zugriffsrechten bei äquivalenter Funktionalität, jedoch ohne die entsprechenden Beschränkungen.

Eine Rolle ist ein Datenbankobjekt, in dem mindestens ein Zugriffsrecht gruppiert wird. Eine Rolle kann mittels einer Anweisung GRANT Benutzern, Gruppen, PUBLIC oder anderen Rollen zugeordnet werden. Eine Rolle kann aber auch mit einer Anweisung CREATE TRUSTED CONTEXT oder ALTER TRUSTED CONTEXT einem gesicherten Kontext zugewiesen werden. Eine Rolle kann für das Verbindungsattribut SESSION_USER ROLE in einer Workloaddefinition angegeben werden.

Rollen bieten verschiedene Vorteile, die die Verwaltung von Zugriffsrechten in einem Datenbanksystem erleichtern:

- Sicherheitsadministratoren können den Zugriff auf ihre Datenbanken auf eine Weise steuern, die die Struktur ihrer Organisationen widerspiegelt (sie können in der Datenbank Rollen erstellen, die den Aufgabenbereichen direkt entsprechen).
- Benutzern werden die Rollen zugewiesen, die den Zuständigkeiten der Benutzer entsprechen. Wenn sich die Zuständigkeiten ändern, kann einem Benutzer ohne großen Aufwand eine Rolle entzogen oder eine andere zugewiesen werden.
- Die Zuweisung von Zugriffsrechten wird vereinfacht. Statt jedem einzelnen Benutzer mit einem bestimmten Aufgabenbereich dieselben Zugriffsrechte zu erteilen, kann der Administrator diese Gruppe von Zugriffsrechten einer Rolle zuweisen, die diesen Aufgabenbereich repräsentiert und kann anschließend jedem Benutzer mit diesem Aufgabenbereich diese Rolle zuweisen.
- Die einer Rolle zugewiesenen Zugriffsrechte können aktualisiert werden. Dieses Update wird an alle Benutzer mit dieser Rolle verteilt, der Administrator muss die Zugriffsrechte nicht für jeden Benutzer einzeln aktualisieren.
- Die den Rollen zugewiesenen Zugriffsrechte und Berechtigungen werden immer verwendet, wenn Sie Sichten, Trigger, MQTs (Materialized Query Tables), statisches SQL und SQL-Routinen erstellen, während die Zugriffsrechte und Berechtigungen, die Gruppen zugewiesen sind, nicht verwendet werden.

Dies liegt daran, dass das DB2-Datenbanksystem nicht ermitteln kann, wann sich die Zugehörigkeit zu einer Gruppe ändert, da die Gruppe durch Software anderer Anbieter verwaltet wird, beispielsweise durch das Betriebssystem oder ein LDAP-Verzeichnis. Da Rollen innerhalb der Datenbank verwaltet werden, kann das DB2-Datenbanksystem ermitteln, wann Änderungen bei der Autorisierung auftreten und kann entsprechend reagieren. Gruppen zugewiesene Rollen bleiben aus demselben Grund unberücksichtigt wie die Gruppen selbst.

- Alle Rollen, die einem Benutzer zugewiesen sind, werden aktiviert, wenn dieser Benutzer eine Verbindung herstellt, d. h., es werden bei der Verbindungsherstellung alle den Rollen eines Benutzers zugewiesenen Zugriffsrechte und Berechtigungen berücksichtigt. Rollen können nicht explizit aktiviert oder inaktiviert werden.
- Der Sicherheitsadministrator kann die Verwaltung einer Rolle Dritten übertragen.

Alle DB2-Zugriffsrechte und Berechtigungen, die innerhalb einer Datenbank zugewiesen werden können, können einer Rolle zugewiesen werden. Einer Rolle können beispielsweise die folgenden Zugriffsrechte und Berechtigungen erteilt werden:

- Die Datenbankberechtigungen DBADM, SECADM, DATAACCESS, ACCESSCTRL, SQLADM, WLMADM, LOAD und IMPLICIT_SCHEMA
- Die Datenbankberechtigungen CONNECT, CREATETAB, CREATE_NOT_FENCED, BINDADD, CREATE_EXTERNAL_ROUTINE oder QUIESCE_CONNECT
- Alle Zugriffsrechte für Datenbankobjekte (einschließlich CONTROL)

Wenn ein Benutzer eine Datenbankverbindung herstellt, werden die Rollen eines Benutzers automatisch aktiviert und bei der Autorisierung berücksichtigt, d. h., Sie brauchen eine Rolle nicht mit der Anweisung SET ROLE zu aktivieren. Wenn Sie beispielsweise eine Sicht, eine MQT (Materialized Query Table), einen Trigger, ein Paket oder eine SQL-Routine erstellen, werden die Zugriffsrechte angewendet, die Sie über Rollen erhalten. Zugriffsrechte, die Sie über Rollen erhalten, die Gruppen zugewiesen sind, in denen Sie Mitglied sind, werden jedoch nicht angewendet.

Eine Rolle hat keinen Eigner. Der Sicherheitsadministrator kann mit der Klausel WITH ADMIN OPTION der Anweisung GRANT die Verwaltung der Rolle an einen anderen Benutzer delegieren, sodass dieser andere Benutzer die Rollenzugehörigkeit steuern kann.

Einschränkungen

Für die Verwendung von Rollen gelten einige Einschränkungen:

- Eine Rolle kann nicht Eigner von Datenbankobjekten sein.
- Gruppen zugewiesene Berechtigungen und Rollen werden bei der Erstellung der folgenden Datenbankobjekte nicht berücksichtigt:
 - Pakete, die statisches SQL enthalten
 - Sichten
 - Materialized Query Tables (MQT)
 - Trigger
 - SQL-Routinen

Bei der Erstellung dieser Objekte werden nur die Rollen berücksichtigt, die (direkt oder indirekt, beispielsweise durch eine Rollenhierarchie) dem Benutzer, der das Objekt erstellt, oder PUBLIC zugewiesen sind.

Anhang C. Gesicherter Kontext und gesicherte Verbindungen

Ein gesicherter Kontext ist ein Datenbankobjekt, das eine Vertrauensbeziehung für eine Verbindung zwischen der Datenbank und einer externen Entität (z. B. einem Anwendungsserver) definiert.

Die Vertrauensbeziehung basiert auf den folgenden Attributen:

- Systemberechtigungs-ID: Stellt den Benutzer dar, der eine Datenbankverbindung herstellt
- IP-Adresse (oder Domänenname): Stellt den Host dar, von dem aus eine Datenbankverbindung hergestellt wird
- Datenstromverschlüsselung: Stellt die Verschlüsselungseinstellung (falls vorhanden) für die Datenübertragung zwischen dem Datenbankserver und dem Datenbankclient dar

Stellt ein Benutzer eine Datenbankverbindung her, überprüft das DB2-Datenbanksystem, ob die Verbindung mit der Definition eines gesicherten Kontextobjekts in der Datenbank übereinstimmt. Liegt eine Übereinstimmung vor, gilt die Datenbankverbindung als gesichert.

Über eine gesicherte Verbindung kann der Initiator dieser gesicherten Verbindung zusätzliche Funktionalität beziehen, die außerhalb des Bereichs einer gesicherten Verbindung möglicherweise nicht verfügbar ist. Die zusätzliche Funktionalität variiert abhängig davon, ob die gesicherte Verbindung explizit oder implizit ist.

Dem Initiator einer explizit gesicherten Verbindung steht die folgende Funktionalität zur Verfügung:

- Durchführen eines Wechsels von der aktuellen Benutzer-ID für die Verbindung zu einer anderen Benutzer-ID, entweder mit oder ohne Authentifizierung
- Beziehen zusätzlicher Zugriffsrechte über das Rollenvererbungsfeature gesicherter Kontexte

Eine implizit gesicherte Verbindung ist eine gesicherte Verbindung, die nicht explizit angefordert wurde. Die implizit gesicherte Verbindung entsteht aus einer normalen Verbindungsanforderung statt einer Anforderung einer explizit gesicherten Verbindung. Zum Abrufen einer impliziten Verbindung sind keine Änderungen am Anwendungscode erforderlich. Ebenso hat es auf den Rückkehrcode der Verbindung keine Auswirkung, ob sie eine implizit gesicherte Verbindung anfordern oder nicht (wenn Sie eine explizit gesicherte Verbindung anfordern, gibt der Rückkehrcode der Verbindung an, ob die Anforderung erfolgreich war). Der Initiator einer implizit gesicherten Verbindung kann zusätzliche Zugriffsrechte nur über das Rollenvererbungsfeature gesicherter Kontexte anfordern und er kann die Benutzer-ID nicht ändern.

Verbessern der Sicherheit durch das Verwenden von gesichertem Kontext

Das dreischichtige Anwendungsmodell erweitert den zweischichtigen Standardclient und das zweischichtige Standardservermodell durch eine Mittelschicht, die zwischen der Clientanwendung und dem Datenbankserver eingefügt wird. Dies wurde in den vergangenen Jahren vor allem durch das Aufkommen webbasierter Technologien und der J2EE-Plattform (Java 2 Enterprise Edition) zur gängigen Pra-

xis. Ein Beispiel für ein Softwareprodukt, welches das dreischichtige Anwendungsmodell unterstützt, ist IBM WebSphere Application Server (WAS).

In einem dreischichtigen Anwendungsmodell ist die Mittelschicht für das Authentifizieren der Benutzer verantwortlich, die die Clientanwendungen ausführen. Ferner verwaltet sie die Interaktionen mit dem Datenbankserver. Herkömmlicherweise geschehen alle Interaktionen mit dem Datenbankserver über eine von der Mittelschicht eingerichtete Datenbankverbindung, wobei die Mittelschicht mittels einer Benutzer-ID und eines Berechtigungsnachweises gegenüber dem Datenbankserver ausgewiesen wird. Das bedeutet, dass der Datenbankserver für die gesamte Berechtigungsprüfung und zu allen Prüfzwecken, die bei einem Datenbankzugriff (einschließlich aller im Auftrag eines Benutzers von der Mittelschicht durchgeführten Zugriffe) erforderlich werden, die der Benutzer-ID der Mittelschicht zugeordneten Zugriffsrechte verwendet.

Das dreischichtige Anwendungsmodell bietet verschiedene Vorteile, das Ausführen aller Interaktionen mit dem Datenbankserver (z. B. eine Benutzeranforderung) unter der Berechtigungs-ID der Mittelschicht wirft jedoch einige Sicherheitsprobleme auf, die wie folgt zusammengefasst werden können:

- Verlust der Benutzeridentität
Aus Gründen der Zugriffskontrolle bevorzugen es einige Unternehmen, die tatsächliche Identität des Benutzers zu kennen, der auf die Datenbank zugreift.
- Verminderte Benutzerverantwortlichkeit
Die Gewährleistung der Verantwortlichkeit durch Prüfung und Überwachung ist ein Grundprinzip der Datenbanksicherheit. Wenn die Identität des Benutzers nicht bekannt ist, ist es schwierig, die von der Mittelschicht selbst zu eigenen Zwecken durchgeführten Aktionen von den Aktionen zu unterscheiden, die sie im Auftrag eines Benutzers durchführt.
- Übermäßiges Erteilen von Zugriffsrechten für die Berechtigungs-ID der Mittelschicht
Die Berechtigungs-ID der Mittelschicht muss über alle erforderlichen Zugriffsrechte verfügen, um alle Anforderungen aller Benutzer ausführen zu können. Dies führt zu dem Sicherheitsproblem, dass Benutzer, die keinen Zugriff auf bestimmte Informationen benötigen, dennoch darauf zugreifen können.
- Verringerte Sicherheit
Zusätzlich zu dem unter dem vorherigen Punkt genannten Sicherheitsproblem erfordert das aktuelle Konzept, dass die von der Mittelschicht verwendete Berechtigungs-ID Zugriffsrechte für alle Ressourcen erhalten muss, auf die durch Benutzeranforderungen möglicherweise zugegriffen wird. Sollte diese Berechtigungs-ID der Mittelschicht beeinträchtigt werden, werden dadurch alle diese Ressourcen zugänglich.
- Überlauf zwischen Benutzern derselben Verbindung
Durch einen vorherigen Benutzer vorgenommene Änderungen können sich auf den aktuellen Benutzer auswirken.

Es ist offensichtlich ein Mechanismus erforderlich, der dafür sorgt, dass für die von der Mittelschicht für einen Benutzer ausgeführten Datenbankanforderungen auch die Identität und die Datenbankzugriffsrechte dieses Benutzers verwendet werden. Der direkteste Ansatz zum Erreichen dieses Ziels ist, die Mittelschicht für jeden Benutzer unter Verwendung der ID und des Kennworts des Benutzers eine neue Verbindung herstellen zu lassen und dann die Anforderungen des Benutzers über diese Verbindung zu senden. Auch wenn dies einfach klingt, bringt dieser Ansatz verschiedene Nachteile mit sich, u. a. folgende:

- Nichtanwendbarkeit für bestimmte Mittelschichten. Viele Mittelschichtserver verfügen nicht über die erforderlichen Benutzerauthentifizierungsnachweise zum Herstellen einer Verbindung.
- Leistungseinbußen. Mit dem Herstellen einer neuen physischen Verbindung und dem erneuten Authentifizieren des Benutzers auf dem Datenbankserver ist sind offensichtliche Leistungseinbußen verbunden.
- Erhöhter Wartungsaufwand. In Situationen, in denen Sie keine zentralisierte Sicherheitskonfiguration oder keine einmalige Anmeldung (Single Sign-on) verwenden, kommt es durch das parallele Vorhandensein zweier Benutzerdefinitionen (eine in der Mittelschicht und eine auf dem Server) zu erhöhtem Wartungsaufwand. In diesem Szenario muss an verschiedenen Stellen jeweils ein anderes Kennwort eingegeben werden.

Die Funktionalität für gesicherte Kontexte soll dieses Problem beheben. Der Sicherheitsadministrator kann in der Datenbank ein gesichertes Kontextobjekt erstellen, das eine Vertrauensbeziehung zwischen der Datenbank und der Mittelschicht definiert. Die Mittelschicht kann anschließend eine explizit gesicherte Verbindung zur Datenbank herstellen, welche es der Mittelschicht ermöglicht, von der aktuellen Benutzer-ID der Verbindung zu einer anderen Benutzer-ID zu wechseln, entweder mit oder ohne Authentifizierung. Gesicherte Kontexte beheben nicht nur das Problem der Identitätszusicherung für Endbenutzer, sondern sie bieten noch einen weiteren Vorteil. Hierbei handelt es sich um die Möglichkeit, zu steuern, wann einem Datenbankbenutzer ein Zugriffsrecht erteilt wird. Wenn nicht gesteuert werden kann, wann einem Benutzer ein Zugriffsrecht erteilt wird, beeinträchtigt dies die Gesamtsicherheit. Beispielsweise könnten Zugriffsrechte zu anderen Zwecken verwendet werden als ursprünglich beabsichtigt. Der Sicherheitsadministrator kann einer Rolle eine beliebige Anzahl Zugriffsrechte zuweisen und diese Rolle anschließend einem gesicherten Absicherung zuweisen. Die Vorteile, die mit den dieser Rolle zugeordneten Zugriffsrechte verbunden sind, können nur von (explizit oder implizit) gesicherten Datenbankverbindungen genutzt werden, die der Definition dieses gesicherten Kontexts entsprechen.

Verbessern der Leistung

Wenn Sie gesicherte Verbindungen verwenden, wird durch die folgenden Vorteile die Leistung maximiert:

- Beim Wechseln der aktuellen Benutzer-ID der Verbindung wird keine neue Verbindung hergestellt.
- Wenn die Definition des gesicherten Kontexts keine Authentifizierung der Benutzer-ID erfordert, zu der gewechselt werden soll, entfällt der Aufwand, einen neuen Benutzer auf dem Datenbankserver zu authentifizieren.

Beispiel: Erstellen eines gesicherten Kontexts

Angenommen, der Sicherheitsadministrator erstellt das folgende gesicherte Kontextobjekt:

```
CREATE TRUSTED CONTEXT CTX1
  BASED UPON CONNECTION USING SYSTEM AUTHID BENUTZER2
  ATTRIBUTES (ADDRESS '192.0.2.1')
  DEFAULT ROLE managerRole
  ENABLE
```

Wenn Benutzer *benutzer1* über die IP-Adresse 192.0.2.1 eine gesicherte Verbindung anfordert, gibt das DB2-Datenbanksystem eine Warnung (SQLSTATE 01679, SQL-CODE +20360) zurück, die besagt, dass keine gesicherte Verbindung hergestellt werden konnte und dass Benutzer *benutzer1* nur eine nicht gesicherte Verbindung erhielt. Wenn jedoch Benutzer *benutzer2* über die IP-Adresse 192.0.2.1 eine gesicherte Verbindung anfordert, wird diese Anforderung erfüllt, da die Verbindungsattribute durch den gesicherten Kontext CTX1 erfüllt sind. Nachdem Benutzer *benutzer2* eine gesicherte Verbindung hergestellt hat, kann er nun alle Zugriffsrechte und Berechtigungen beziehen, die der gesicherten Kontextrolle *managerRole* zugeordnet sind. Diese Zugriffsrechte und Berechtigungen stehen Benutzer *benutzer2* außerhalb des Bereichs dieser gesicherten Verbindung möglicherweise nicht zur Verfügung.

Anhang D. Übersicht über technische Informationen zu DB2

Technische Informationen zu DB2 liegen in verschiedenen Formaten vor, die auf unterschiedliche Weise abgerufen werden können.

Die technischen Informationen zu DB2 stehen über die folgenden Tools und Methoden zur Verfügung:

- DB2 Information Center
 - Themen (zu Tasks, Konzepten und Referenzinformationen)
 - Beispielprogramme
 - Lernprogramme
- DB2-Bücher
 - PDF-Dateien (für den Download verfügbar)
 - PDF-Dateien (auf der DB2-PDF-DVD)
 - Gedruckte Bücher
- Hilfe für Befehlszeile
 - Hilfe für Befehle
 - Hilfe für Nachrichten

Anmerkung: Die Themen des DB2 Information Center werden häufiger aktualisiert als die PDF- und Hardcopybücher. Um stets die neuesten Informationen zur Verfügung zu haben, sollten Sie die Dokumentationsaktualisierungen installieren, sobald diese verfügbar sind, oder das DB2 Information Center unter ibm.com aufrufen.

Darüber hinaus können Sie auf zusätzliche technische Informationen zu DB2, wie beispielsweise technische Hinweise (Technotes), White Papers und IBM Redbooks, online über ibm.com zugreifen. Rufen Sie dazu die Website 'DB2 Information Management - Software - Library' unter <http://www.ibm.com/software/data/sw-library/> auf.

Feedback zur Dokumentation

Senden Sie uns Ihr Feedback zur DB2-Dokumentation! Wenn Sie Anregungen zur Verbesserung der DB2-Dokumentation haben, senden Sie eine E-Mail an db2docs@ca.ibm.com. Das DB2-Dokumentationsteam bearbeitet das gesamte Feedback, kann jedoch nicht im Einzelnen auf Ihre E-Mails antworten. Nennen Sie uns, wenn möglich, konkrete Beispiele, sodass wir die Problemstellung besser beurteilen können. Wenn Sie uns Feedback zu einem bestimmten Thema oder einer bestimmten Hilfedatei senden, geben Sie den entsprechenden Titel sowie die URL an.

Verwenden Sie diese E-Mail-Adresse nicht, wenn Sie sich an den DB2-Kundendienst wenden möchten. Wenn ein technisches Problem bei DB2 vorliegt, das Sie mithilfe der Dokumentation nicht beheben können, fordern Sie beim zuständigen IBM Service-Center Unterstützung an.

Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format

Die folgenden Tabellen enthalten eine Beschreibung der DB2-Bibliothek, die im IBM Publications Center unter www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss zur Verfügung steht. Über die folgende Adresse können Sie englische Handbücher im PDF-Format sowie übersetzte Versionen zu DB2 Version 10.1 herunterladen: www.ibm.com/support/docview.wss?rs=71&uid=swg27009474.

In den Tabellen sind die Bücher, die in gedruckter Form zur Verfügung stehen, gekennzeichnet; möglicherweise sind diese in Ihrem Land oder Ihrer Region jedoch nicht verfügbar.

Die Formnummer wird bei jeder Aktualisierung eines Handbuchs erhöht. Anhand der nachfolgenden Liste können Sie sicherstellen, dass Sie die jeweils neueste Version des Handbuchs lesen.

Anmerkung: Das *DB2 Information Center* wird häufiger aktualisiert als die PDF- und Hardcopybücher.

Tabelle 278. Technische Informationen zu DB2

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>Administrative API Reference</i>	SC27-3864-00	Ja	April 2012
<i>Administrative Routines and Views</i>	SC27-3865-01	Nein	Januar 2013
<i>Call Level Interface Guide and Reference Volume 1</i>	SC27-3866-01	Ja	Januar 2013
<i>Call Level Interface Guide and Reference Volume 2</i>	SC27-3867-01	Ja	Januar 2013
<i>Command Reference</i>	SC27-3868-01	Ja	Januar 2013
<i>Datenbankverwaltung - Konzepte und Konfiguration - Referenzinformationen</i>	SC12-4673-01	Ja	Januar 2013
<i>Dienstprogramme für das Versetzen von Daten - Handbuch und Referenz</i>	SC12-4691-01	Ja	Januar 2013
<i>Datenbanküberwachung - Handbuch und Referenz</i>	SC12-4674-01	Ja	Januar 2013
<i>Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz</i>	SC12-4692-01	Ja	Januar 2013
<i>Datenbanksicherheit</i>	SC12-4693-01	Ja	Januar 2013
<i>DB2 Workload Management - Handbuch und Referenz</i>	SC12-4683-01	Ja	Januar 2013
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-3873-01	Ja	Januar 2013

Tabelle 278. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
<i>Developing Embedded SQL Applications</i>	SC27-3874-01	Ja	Januar 2013
<i>Developing Java Applications</i>	SC27-3875-01	Ja	Januar 2013
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-3876-00	Nein	April 2012
<i>Developing RDF Applications for IBM Data Servers</i>	SC27-4462-00	Ja	Januar 2013
<i>Developing User-defined Routines (SQL and External)</i>	SC27-3877-01	Ja	Januar 2013
<i>Getting Started with Database Application Development</i>	GI13-2046-01	Ja	Januar 2013
<i>Installation und Verwaltung von DB2 unter Linux und Windows - Erste Schritte</i>	GI11-3285-00	Ja	April 2012
<i>Globalisierung</i>	SC12-4694-00	Ja	April 2012
<i>DB2-Server - Installation</i>	SC12-4677-01	Ja	Januar 2013
<i>IBM Data Server-Clients - Installation</i>	SC12-4678-00	Nein	April 2012
<i>Fehlernachrichten, Band 1</i>	SC12-4686-01	Nein	Januar 2013
<i>Fehlernachrichten, Band 2</i>	SC12-4687-01	Nein	Januar 2013
<i>Net Search Extender - Verwaltung und Benutzerhandbuch</i>	SC12-4689-01	Nein	Januar 2013
<i>Partitionierung und Clustering</i>	SC12-4695-01	Ja	Januar 2013
<i>Preparation Guide for DB2 10.1 Fundamentals Exam 610</i>	SC27-4540-00	Nein	Januar 2013
<i>Preparation Guide for DB2 10.1 DBA for Linux, UNIX, and Windows Exam 611</i>	SC27-4541-00	Nein	Januar 2013
<i>pureXML - Handbuch</i>	SC12-4684-01	Ja	Januar 2013
<i>Spatial Extender - Benutzer- und Referenzhandbuch</i>	SC12-4688-00	Nein	April 2012
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-3896-01	Ja	Januar 2013
<i>SQL Reference Volume 1</i>	SC27-3885-01	Ja	Januar 2013

Tabelle 278. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
SQL Reference Volume 2	SC27-3886-01	Ja	Januar 2013
Text Search	SC12-4690-01	Ja	Januar 2013
Fehlerbehebung und Optimieren der Datenbankleistung	SC12-4675-01	Ja	Januar 2013
Upgrade auf DB2 Version 10.1	SC12-4676-01	Ja	Januar 2013
Neuerungen in DB2 Version 10.1	SC12-4682-01	Ja	Januar 2013
XQuery - Referenz	SC12-4685-01	Nein	Januar 2013

Tabelle 279. Technische Informationen zu DB2 Connect

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
DB2 Connect - Installation und Konfiguration von DB2 Connect Personal Edition	SC12-4679-00	Ja	April 2012
DB2 Connect - Installation und Konfiguration von DB2 Connect-Servern	SC12-4680-01	Ja	Januar 2013
DB2 Connect - Benutzerhandbuch	SC12-4681-01	Ja	Januar 2013

Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor

DB2-Produkte geben für Bedingungen, die aufgrund einer SQL-Anweisung generiert werden können, einen SQLSTATE-Wert zurück. Die SQLSTATE-Hilfe erläutert die Bedeutung der SQL-Statuswerte und der SQL-Statusklassencodes.

Vorgehensweise

Zum Starten der Hilfe für SQL-Statuswerte müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

? SQL-Status oder ? Klassencode

Hierbei steht *SQL-Status* für einen gültigen fünfstelligen SQL-Statuswert und *Klassencode* für die ersten beiden Ziffern dieses Statuswerts.

So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, durch die Eingabe von ? 08 Hilfe für den Klassencode 08.

Zugriff auf verschiedene Versionen des DB2 Information Center

Die Dokumentation für andere Versionen der DB2-Produkte finden Sie in den jeweiligen Information Centers unter ibm.com.

Informationen zu diesem Vorgang

Für Themen aus DB2 Version 10.1 lautet die URL für das *DB2 Information Center* <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1>.

Für Themen aus DB2 Version 9.8 lautet die URL des *DB2 Information Center* <http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/>.

Für Themen aus DB2 Version 9.7 lautet die URL des *DB2 Information Center* <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>.

Für Themen aus DB2 Version 9.5 lautet die URL des *DB2 Information Center* <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>.

Für Themen aus DB2 Version 9.1 lautet die URL des *DB2 Information Center* <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

Für Themen aus DB2 Version 8 lautet die URL des *DB2 Information Center* <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

Aktualisieren des auf Ihrem Computer oder Intranet-Server installierten DB2 Information Center

Ein lokal installiertes DB2 Information Center muss regelmäßig aktualisiert werden.

Vorbereitende Schritte

Ein DB2 Version 10.1 Information Center muss bereits installiert sein. Einzelheiten hierzu finden Sie unter „Installation des DB2 Information Center mit dem DB2-Installationsassistenten“ in *DB2-Server - Installation*. Alle für die Installation des Information Center geltenden Voraussetzungen und Einschränkungen gelten auch für die Aktualisierung des Information Center.

Informationen zu diesem Vorgang

Ein vorhandenes DB2 Information Center kann automatisch oder manuell aktualisiert werden:

- Mit automatischen Aktualisierungen werden vorhandene Komponenten und Sprachen des Information Center aktualisiert. Ein Vorteil von automatischen Aktualisierungen ist, dass das Information Center im Vergleich zu einer manuellen Aktualisierung nur für einen kurzen Zeitraum nicht verfügbar ist. Darüber hinaus können automatische Aktualisierungen so konfiguriert werden, dass sie als Teil anderer, regelmäßig ausgeführter Stapeljobs ausgeführt werden.
- Mit manuellen Aktualisierungen können Sie vorhandene Komponenten und Sprachen des Information Center aktualisieren. Automatische Aktualisierungen reduzieren die Ausfallzeiten während des Aktualisierungsprozesses, Sie müssen jedoch den manuellen Prozess verwenden, wenn Sie Komponenten oder Sprachen hinzufügen möchten. Beispiel: Ein lokales Information Center wurde ursprünglich sowohl mit englischer als auch mit französischer Sprachunterstützung installiert; nun soll auch die deutsche Sprachunterstützung installiert werden. Bei einer manuellen Aktualisierung werden sowohl eine Installation der deutschen Sprachunterstützung als auch eine Aktualisierung der vorhandenen Komponenten und Sprachen des Information Center durchgeführt. Sie müssen jedoch bei einer manuellen Aktualisierung das Information Center manuell stop-

pen, aktualisieren und erneut starten. Das Information Center ist während des gesamten Aktualisierungsprozesses nicht verfügbar. Während des automatischen Aktualisierungsprozesses kommt es zu einem Ausfall des Information Center, und es wird erst wieder nach der Aktualisierung erneut gestartet.

Dieser Abschnitt enthält Details zum Prozess der automatischen Aktualisierung. Anweisungen zur manuellen Aktualisierung finden Sie im Abschnitt „Manuelles Aktualisieren des auf Ihrem Computer oder Intranet-Server installierten DB2 Information Center“.

Vorgehensweise

Gehen Sie wie folgt vor, um das auf Ihrem Computer bzw. Intranet-Server installierte DB2 Information Center automatisch zu aktualisieren:

1. Unter Linux:
 - a. Navigieren Sie zu dem Pfad, in dem das Information Center installiert ist. Standardmäßig ist das DB2 Information Center im Verzeichnis `/opt/ibm/db2ic/V10.1` installiert.
 - b. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis `doc/bin`.
 - c. Führen Sie das Script `update-ic` aus:
`update-ic`
2. Unter Windows:
 - a. Öffnen Sie ein Befehlsfenster.
 - b. Navigieren Sie zu dem Pfad, in dem das Information Center installiert ist. Standardmäßig ist das DB2 Information Center im Verzeichnis `<Programme>\IBM\DB2 Information Center\Version 10.1` installiert, wobei `<Programme>` das Verzeichnis der Programmdateien angibt.
 - c. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis `doc\bin`.
 - d. Führen Sie die Datei `update-ic.bat` aus:
`update-ic.bat`

Ergebnisse

Das DB2 Information Center wird automatisch erneut gestartet. Standen Aktualisierungen zur Verfügung, zeigt das Information Center die neuen und aktualisierten Abschnitte an. Waren keine Aktualisierungen für das Information Center verfügbar, wird eine entsprechende Nachricht zum Protokoll hinzugefügt. Die Protokolldatei befindet sich im Verzeichnis `doc\eclipse\configuration`. Der Name der Protokolldatei ist eine Zufallszahl. Beispiel: `1239053440785.log`.

Manuelles Aktualisieren des auf Ihrem Computer oder Intranet-Server installierten DB2 Information Center

Wenn Sie das DB2 Information Center lokal installiert haben, können Sie Dokumentationsaktualisierungen von IBM abrufen und installieren.

Informationen zu diesem Vorgang

Zur manuellen Aktualisierung des lokal installierten *DB2 Information Center* sind die folgenden Schritte erforderlich:

1. Stoppen Sie das *DB2 Information Center* auf Ihrem Computer und starten Sie das Information Center im Standalone-Modus erneut. Die Ausführung des Information Center im Standalone-Modus verhindert, dass andere Benutzer in Ihrem

Netz auf das Information Center zugreifen, und ermöglicht das Anwenden von Aktualisierungen. Die Workstationversion des DB2 Information Center wird stets im Standalone-Modus ausgeführt.

2. Verwenden Sie die Aktualisierungsfunktion, um zu prüfen, welche Aktualisierungen verfügbar sind. Falls Aktualisierungen verfügbar sind, die Sie installieren müssen, können Sie die Aktualisierungsfunktion verwenden, um diese abzurufen und zu installieren.

Anmerkung: Wenn es in der verwendeten Umgebung erforderlich ist, die Aktualisierungen für das *DB2 Information Center* auf einer Maschine zu installieren, die nicht über eine Verbindung zum Internet verfügt, spiegeln Sie die Aktualisierungsseite auf ein lokales Dateisystem und verwenden Sie dabei eine Maschine, die mit dem Internet verbunden ist und auf der das *DB2 Information Center* installiert ist. Wenn viele Benutzer Ihres Netzes die Dokumentationsaktualisierungen installieren sollen, können Sie die Zeit, die jeder einzelne Benutzer für die Aktualisierungen benötigt, reduzieren, indem Sie die Aktualisierungsseite lokal spiegeln und ein Proxy dafür erstellen.

Ist dies der Fall, verwenden Sie die Aktualisierungsfunktion, um die Pakete abzurufen. Die Aktualisierungsfunktion ist jedoch nur im Standalone-Modus verfügbar.

3. Stoppen Sie das im Standalone-Modus gestartete Information Center und starten Sie das *DB2 Information Center* auf Ihrem Computer erneut.

Anmerkung: Unter Windows 2008 und Windows Vista (und neueren Versionen) müssen die in diesem Abschnitt aufgeführten Befehle mit Administratorberechtigung ausgeführt werden. Zum Öffnen einer Eingabeaufforderung oder eines Grafiktools mit vollen Administratorberechtigungen klicken Sie mit der rechten Maustaste die Verknüpfung an und wählen Sie **Als Administrator ausführen** aus.

Vorgehensweise

Gehen Sie wie folgt vor, um das auf Ihrem Computer bzw. Intranet-Server installierte *DB2 Information Center* zu aktualisieren:

1. Stoppen Sie das *DB2 Information Center*.
 - Unter Windows: Klicken Sie **Start > Systemsteuerung > Verwaltung > Dienste** an. Klicken Sie mit der rechten Maustaste das **DB2 Information Center** an und wählen Sie **Beenden** aus.
 - Unter Linux: Geben Sie den folgenden Befehl ein:

```
/etc/init.d/db2icdv10 stop
```
2. Starten Sie das Information Center im Standalone-Modus.
 - Unter Windows:
 - a. Öffnen Sie ein Befehlsfenster.
 - b. Navigieren Sie zu dem Pfad, in dem das Information Center installiert ist. Standardmäßig ist das *DB2 Information Center* im Verzeichnis *Programme\IBM\DB2 Information Center\Version 10.1* installiert, wobei *Programme* das Verzeichnis der Programmdateien angibt.
 - c. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis *doc\bin*.
 - d. Führen Sie die Datei *help_start.bat* aus:

```
help_start.bat
```
 - Unter Linux:

- a. Navigieren Sie zu dem Pfad, in dem das Information Center installiert ist. Standardmäßig ist das *DB2 Information Center* im Verzeichnis `/opt/ibm/db2ic/V10.1` installiert.
- b. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis `doc/bin`.
- c. Führen Sie das Script `help_start` aus:

```
help_start
```

Der standardmäßig auf dem System verwendete Web-Browser wird geöffnet und zeigt die Standalone-Version des Information Center an.

3. Klicken Sie die Aktualisierungsschaltfläche (🔄) an. (JavaScript muss im verwendeten Browser aktiviert sein.) Klicken Sie im rechten Fenster des Information Center die Schaltfläche für die Suche nach Aktualisierungen an. Eine Liste der Aktualisierungen für die vorhandene Dokumentation wird angezeigt.
4. Wählen Sie zum Initiieren des Installationsprozesses die gewünschten Aktualisierungen aus und klicken Sie anschließend die Schaltfläche für die Installation der Aktualisierungen an.
5. Klicken Sie nach Abschluss des Installationsprozesses **Fertigstellen** an.
6. Stoppen Sie das im Standalone-Modus gestartete Information Center:
 - Unter Windows: Navigieren Sie innerhalb des Installationsverzeichnisses zum Verzeichnis `doc\bin`, und führen Sie die Datei `help_end.bat` aus:

```
help_end.bat
```

Anmerkung: Die Stapeldatei `help_end` enthält die Befehle, die erforderlich sind, um die Prozesse, die mit der Stapeldatei `help_start` gestartet wurden, ordnungsgemäß zu stoppen. Verwenden Sie nicht die Tastenkombination `Strg+C` oder eine andere Methode, um `help_start.bat` zu stoppen.

- Unter Linux: Navigieren Sie innerhalb des Installationsverzeichnisses zum Verzeichnis `doc/bin`, und führen Sie das Script `help_end` aus:

```
help_end
```

Anmerkung: Das Script `help_end` enthält die Befehle, die erforderlich sind, um die Prozesse, die mit dem Script `help_start` gestartet wurden, ordnungsgemäß zu stoppen. Verwenden Sie keine andere Methode, um das Script `help_start` zu stoppen.

7. Starten Sie das *DB2 Information Center* erneut.
 - Unter Windows: Klicken Sie **Start > Systemsteuerung > Verwaltung > Dienste** an. Klicken Sie mit der rechten Maustaste das **DB2 Information Center** an und wählen Sie **Start** aus.
 - Unter Linux: Geben Sie den folgenden Befehl ein:

```
/etc/init.d/db2icdv10 start
```

Ergebnisse

Im aktualisierten *DB2 Information Center* werden die neuen und aktualisierten Themen angezeigt.

DB2-Lernprogramme

Die DB2-Lernprogramme unterstützen Sie dabei, sich mit den unterschiedlichen Aspekten der DB2-Produkte vertraut zu machen. Die Lerneinheiten bieten eine in einzelne Schritte unterteilte Anleitung.

Vorbereitungen

Die XHTML-Version des Lernprogramms kann über das Information Center unter <http://publib.boulder.ibm.com/infocenter/db2luw/v10r1/> angezeigt werden.

In einigen der Lerneinheiten werden Beispieldaten und Codebeispiele verwendet. Informationen zu bestimmten Voraussetzungen für die Ausführung der Tasks finden Sie in der Beschreibung des Lernprogramms.

DB2-Lernprogramme

Klicken Sie zum Anzeigen des Lernprogramms den Titel an.

„pureXML“ in *pureXML - Handbuch*

Einrichten einer DB2-Datenbank, um XML-Daten zu speichern und Basisoperationen mit dem nativen XML-Datenspeicher auszuführen.

Informationen zur Fehlerbehebung in DB2

Es steht eine breite Palette verschiedener Informationen zur Fehlerbestimmung und Fehlerbehebung zur Verfügung, um Sie bei der Verwendung von DB2-Datenbankprodukten zu unterstützen.

DB2-Dokumentation

Informationen zur Fehlerbehebung stehen im Handbuch *Fehlerbehebung und Optimieren der Datenbankleistung* oder im Abschnitt mit grundlegenden Informationen zu Datenbanken im *DB2 Information Center* zur Verfügung, darunter:

- Informationen zum Eingrenzen und Aufdecken von Problemen mithilfe der Diagnosetools und -dienstprogramme von DB2.
- Lösungsvorschläge zu den am häufigsten auftretenden Problemen.
- Ratschläge zum Lösen anderer Probleme, die bei Verwendung der DB2-Datenbankprodukte auftreten können.

IBM Support Portal

Im IBM Support Portal finden Sie Informationen zu Problemen und den möglichen Ursachen und Fehlerbehebungsmaßnahmen. Die Website mit technischer Unterstützung enthält Links zu den neuesten DB2-Veröffentlichungen, technischen Hinweisen (TechNotes), APARs (Authorized Program Analysis Reports) und Fehlerkorrekturen, Fixpacks sowie weiteren Ressourcen. Sie können diese Wissensbasis nach möglichen Lösungen für aufgetretene Probleme durchsuchen.

Sie können auf das IBM Support Portal über die folgende Website zugreifen: http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_for_Linux,_UNIX_and_Windows.

Bedingungen

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

Anwendbarkeit: Diese Bedingungen gelten zusätzlich zu den Nutzungsbedingungen für die IBM Website.

Persönliche Nutzung: Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile dieser Veröffentlichungen ohne ausdrückliche Genehmigung von IBM nicht weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Kommerzielle Nutzung: Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile dieser Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens nicht vervielfältigen, weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Rechte: Abgesehen von den hier gewährten Berechtigungen erhalten Sie keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum.

IBM behält sich das Recht vor, die in diesem Dokument gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Vorschriften, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Informationen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelsüblichkeit, die Verwendungsfähigkeit oder die Freiheit der Rechte Dritter zur Verfügung gestellt.

IBM Marken: IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der International Business Machines Corporation. Weitere Produkt- oder Servicenamen können Marken von IBM oder anderen Herstellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite www.ibm.com/legal/copytrade.shtml.

Anhang E. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Die Informationen über Produkte anderer Hersteller als IBM basieren auf den zum Zeitpunkt der ersten Veröffentlichung dieses Dokuments verfügbaren Informationen und können geändert werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuauflage veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung kann Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes enthalten. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM haftet nicht für Schäden, die durch Verwendung der Beispielprogramme entstehen.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *Jahr/Jahre angeben*. Alle Rechte vorbehalten.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- oder Servicennamen können Marken von oder anderen Herstellern sein. IBM oder anderen Herstellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite „Copyright and trademark information“ unter www.ibm.com/legal/copytrade.shtml.

Die folgenden Namen sind Marken oder eingetragene Marken anderer Unternehmen.

- Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.
- Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken von Oracle und/oder ihren verbundenen Unternehmen.
- UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.
- Intel, das Intel-Logo, Intel Inside, Intel Inside logo, Celeron, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA und anderen Ländern.
- Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicennamen können Marken anderer Hersteller sein.

Index

A

- Abfragen
 - Monitorelemente
 - queue_assignments_total 475
 - queue_size_top 475
 - queue_time_total 476
- Abschnitte
 - Monitorelemente
 - section_env 484
- ACTIVITYTOTALTIME (Aktivitätsschwellenwert)
 - Details 143
- Agenten
 - Serviceklassennutzung 98
- Agentenpriorität
 - Serviceklassen 80
- AGGSQLTEMPSPACE (Schwellenwert) 153
- AIX Workload Manager
 - DB2-Workload-Management integrieren 305
 - Prozessorpriorität 80
- Aktivitäten
 - abbrechen
 - Prozedur 240, 300
 - Szenarios 376, 379, 382
 - Analysebeispiel 66
 - Anwendung von Arbeitsaktionen 118
 - Business-Intelligence-Berichte
 - steuern (Szenario) 169
 - Datenerfassung
 - Beispiel 302
 - Prozedur 297
 - Designadvisor 299
 - lang laufende
 - Szenario 376
 - Messdaten zusammenfassen
 - über untergeordnete Aktivitäten hinweg 373
 - Monitorelemente
 - act_throughput 442
 - act_total 442
 - activity_collected 443
 - activity_id 444
 - activity_secondary_id 444
 - activity_type 445
 - coord_act_aborted_total 457
 - coord_act_completed_total 457
 - coord_act_rejected_total 464
 - parent_activity_id 473
 - niedriger geschätzter Aufwand und lange Laufzeit
 - Szenario 382
 - Rogue 300
 - Schwellenwerte 142
 - Status in Serviceklassen 83
 - steuern
 - Szenario 171
 - Szenario (Business-Intelligence-Berichte) 169
 - Übersicht 15
 - Warteschlange 152
 - zu Arbeitsklassen zuordnen 64
 - zu Serviceklassen zuordnen 76
- Aktivitäten (Ereignismonitore)
 - WLM 254
 - Aktivitäten neu zuordnen
 - Beispielscripts 166
 - Details 172
- Aktivitätsdurchsatz
 - Monitorelemente
 - act_throughput 442
- Aktualisierungen
 - DB2 Information Center 553, 554
- Anweisungsaufwurf-ID (Monitorelement) 488
- APIs
 - sqlseti
 - Workloadzuordnung 41
- Arbeitsaktionen
 - ändern 108
 - Arbeitsaktionssets 114
 - erstellen 104
 - inaktivieren 109
 - löschen 110
 - Schwellenwerte 118
 - zu Datenbankaktivitäten zuordnen 118
 - Zuordnung zu anderen Objekten (Beispiel) 100
- Arbeitsaktionssets
 - ändern 112
 - Beispiele
 - Arbeitsaktionsset und Datenbankschwellenwert 125
 - Typ der auszuführenden Arbeit festlegen 127
 - Zuordnung zu anderen Objekten 100
 - Domäne und zulässige Arbeitsaktionen 114
 - erstellen 110
 - inaktivieren 113
 - löschen 114
 - Schwellenwerte angegebene Arbeitsaktionen 118
 - Steuerung des gemeinsamen Zugriffs 120
 - Übersicht 102
 - Workloadenebene
 - Steuerung des gemeinsamen Zugriffs 120
- Arbeitsklassen
 - Aktivitäten zuordnen 64
 - ändern 61
 - Beispiele
 - mit Schlüsselwort ALL definiert 68
 - Zuordnung zu anderen Objekten 100
 - Bewertungsreihenfolge 63
 - erstellen 58
 - löschen 61
 - Übersicht 52
 - unterstützte Schwellenwerte 65
- Arbeitsklassensets
 - ändern 62
 - Bewertungsreihenfolge von Arbeitsklassen 63
 - DML-Aktivitäten verwalten (Beispiel) 67
 - erstellen 62
 - löschen 63
 - Übersicht 55
 - Zuordnung zu anderen Objekten (Beispiel) 100
- Auslastungen
 - Monitorelemente
 - wlo_completed_total 512
 - workload_id 514
 - workload_name 515
 - workload_occurrence_id 517

Auslastungen (Forts.)
 Monitorelemente (Forts.)
 workload_occurrence_state 517
AUTHID, Berechtigungs-ID
 Einschränkungen 541

B

Bedingungen
 Veröffentlichungen 558
Befehle
 SET WORKLOAD
 Details 518
 Maßnahmen zur Fehlerbehebung ergreifen 29
Beispiele
 Aktivitäten mit Arbeitsklassensets verwalten 67
 Arbeitsaktionsset und Schwellenwert 125
 mit Schlüsselwort ALL definierte Arbeitsklasse 68
 Serviceklassen 92
 Workloads Verbindungen zuordnen 45
Bemerkungen 559
Benachrichtigungen
 Schwellenwertverstöße 294
Bereiche
 Monitorelemente
 untere Grenze 447
Betriebssysteme
 DB2-Workload-Management integrieren 305
Bewertungsreihenfolge
 DB2-Workload-Management-Schwellenwerte 134
 Workloads 24
Bins
 Beispiel 284

C

CALL (Anweisung)
 Klassifizierung nach Schema 55
CONCURRENTDBCOORDACTIVITIES (zusammengefasster
 Schwellenwert) 153
CONCURRENTWORKLOADACTIVITIES (zusammengefasster
 Schwellenwert) 155
CONCURRENTWORKLOADOCCURRENCES (zusammenge-
 fasster Schwellenwert) 157
CONNECTIONIDLETIME (Verbindungsschwellenwert) 141
CPU-Anteile
 Monitorelemente
 cpu_shares 466
CPU-Anteile des Workload-Manager-Dispatchers (Konfigurati-
 onsparameter) 522
CPU-Anteilstyp
 Monitorelemente
 cpu_share_type 466
CPU-Auslastung
 Monitorelemente
 cpu_utilization 466
CPU-Geschwindigkeit
 Monitorelemente
 cpu_velocity 467
CPU-Grenzwert
 festlegen 228
 Monitorelemente
 cpu_limit 466
CPUTIME (Aktivitätsschwellenwert) 144
CPUTIMEINSC (Aktivitätsschwellenwert) 145

D

DATATAGINSC (Aktivitätsschwellenwert)
 Details 146
Dateinamen
 allgemein 541
Daten
 zusammenfassen 252
Datenbankobjekte
 DB2-Workload-Management 18
 Rollen 543
DB2 Information Center
 Aktualisierung 553, 554
 Versionen 553
DB2-Workload-Management
 abbrechen
 Aktivitäten 379
Agenten
 Nutzung nach Serviceklasse (Beispiel) 98
 Priorität 80
Aktivitäten
 abbrechen 240, 300, 355
 Aktivitäten mit niedrigem geschätztem Aufwand und
 langen Laufzeiten erkennen (Beispiel) 382
 Arbeitsaktionen zuordnen 118
 Business-Intelligence-Berichte steuern (Szenario) 169,
 171
 Datenerfassung 297, 359
 Datenerfassung (Beispiel) 302
 eingrenzen 325
 ermitteln 356
 Informationen in Designadvisor importieren 299
 lange laufende Aktivitäten ermitteln (Beispiel) 376
 Neuzuordnung 166
 Rogue 300
 Schwellenwerte 142
 Übersicht 15
 unterscheiden 335
 Workloads nach Aktivitätstyp analysieren (Beispiel) 66
 zu Arbeitsklassen zuordnen 64
 zu Serviceklassen zuordnen 76
Aktivitätenwarteschlangen 152
Arbeit
 angeben 15
 steuern 71
Arbeitsaktionen
 ändern 108
 erstellen 104
 für ein Arbeitsaktionsset definieren 114
 inaktivieren 109
 löschen 110
 Schwellenwerte 118
 zu Datenbankaktivitäten zuordnen 118
Arbeitsaktionssets
 ändern 112
 Arbeitsaktionen 99
 erstellen 110
 inaktivieren 113
 löschen 114
 mit Datenbankschwellenwerten verwenden (Bei-
 spiel) 125
 Typ der auszuführenden Arbeit festlegen (Beispiel) 127
 Übersicht 102
Arbeitsklassen
 ändern 61
 Anweisung CALL nach Schema klassifizieren 55
 Bewertungsreihenfolge 63
 erstellen 58

DB2-Workload-Management (*Forts.*)

- Arbeitsklassen (*Forts.*)
 - löschen 61
 - mit Schlüsselwort ALL definiert (Beispiel) 68
 - Übersicht 52
- Arbeitsklassensets
 - ändern 62
 - DML-Aktivitätenverwaltung (Beispiel) 67
 - erstellen 62
 - löschen 63
 - Übersicht 55
- Beispielanwendung 371
- Berichte 361
- CPU-Anteile 212
- CPU-Grenzwert
 - Details 214
 - festlegen 228
- DB2 Governor
 - Beziehung 4
- DDL-Anweisungen 18
- Dispatcher
 - aktivieren 198
 - Details 176
 - maximale Planungsgenauigkeit sicherstellen 199
 - Übersicht 175
 - Umgebungen mit mehreren Membern 197
- Dispatcher-Thread-Parallelität
 - Details 234
 - festlegen 235
- Domänen 1
- Ereignismonitore
 - Übersicht 254
- erweiterte Zusammenfassungen 364
- feste CPU-Anteile
 - Details 200
- gespeicherte Prozeduren
 - WLM_CANCEL_ACTIVITY 263
 - WLM_CAPTURE_ACTIVITY_IN_PROGRESS 263
 - WLM_COLLECT_STATS 263
 - WLM_SET_CLIENT_INFO 263
- häufig gestellte Fragen 4
- Histogramme
 - Durchschnittswerte und Standardabweichung berechnen (Beispiel) 284
 - erstellen 343
 - Übersicht 276
- Histogrammschablonen
 - ändern 283
 - erstellen 282
 - löschen 283
- Integration in Betriebssystem-Workload-Manager 305
- Integration von AIX Workload Manager 305
- Integration von Linux-Workload-Management 312
- Leistungsmodellierung 301
- Lernprogramm 319
- Messdaten 290
- Messdaten zusammenfassen
 - Aktivitäten 373
- minimale CPU-Auslastung
 - Details 230
 - festlegen 233
- Monitorelemente
 - Übersicht 439
 - wlm_queue_assignments_total 509
 - wlm_queue_time_total 510
- Objekteigentumsrecht 18

DB2-Workload-Management (*Forts.*)

- optimieren
 - Kapazitätsplanungsdaten nicht verfügbar (Beispiel) 387
 - Kapazitätsplanungsdaten verfügbar (Beispiel) 385
- Prioritätssteuerung nach Verweildauer
 - Beispielscripts 166
 - Details 161
- Protokollanalysetool 286
- Protokolldaten 361
- Query Patroller 4
- Schwellenwerte
 - ACTIVITYTOTALTIME 143
 - AGGSQLTEMPSPACE 153
 - Aktivität 142
 - Aktivitäten neu zuordnen 161, 172
 - ändern 137
 - Arbeit steuern 128
 - Arbeitsaktionsset und Datenbankschwellenwert (Beispiel) 125
 - Bereichsauflösung 142
 - Bewertungsreihenfolge 134
 - CONCURRENTDBCOORDACTIVITIES 153
 - CONCURRENTWORKLOADACTIVITIES 155
 - CONCURRENTWORKLOADOCCURRENCES 157
 - CONNECTIONIDLETIME 141
 - CPUTIME 144
 - CPUTIMEINSC 145
 - DATATAGINSC 146
 - Datenbankressourcen abteilungsübergreifend verwalten (Beispiel) 139
 - erstellen 136
 - ESTIMATEDSQLCOST 147
 - löschen 138
 - Rogue-Aktivitäten 331
 - SQLROWSREAD 148
 - SQLROWSREADINSC 149
 - SQLROWSRETURNED 150
 - SQLTEMPSPACE 151
 - TOTALMEMBERCONNECTIONS 158
 - TOTALSCMEMBERCONNECTIONS 159
 - Übersicht 128
 - UOWTOTALTIME 160
 - Verbindung 141
 - Verstoßüberwachung 292
 - Zusammenfassung 132, 152
- Serviceklassen
 - ändern 87
 - Beispiel 92
 - erstellen 85
 - Leistungsminderung analysieren (Beispiel) 97
 - löschen 90
 - nicht verfolgte Entitäten 85
 - Ressourcenzuordnung 71
 - Vorablesezugriffspriorität 81, 82
 - Zeitpunktstatistikdaten abrufen (Beispiel) 251
- Serviceklassenkonfiguration mit mehreren leistungsorientierten Schichten
 - Beispielscripts 166
 - Übersicht 161
- Serviceunterklasse 71
- SET WORKLOAD (Befehl) 518
- Statistikdaten
 - mit Statistikereignismonitor erfassen 286
 - Übersicht 264
 - zurücksetzen 289

- DB2-Workload-Management (*Forts.*)
 - Tabellenfunktionen
 - Betriebsdaten 241
 - Daten zusammenfassen (Beispiel) 252
 - erkennen, was auf dem Datenserver ausgeführt wird (Beispiel) 246
 - Informationen der WLM-Schwellenwertwarteschlange abrufen (Beispiel) 253
 - mit Tabellenfunktionen von Snapshot Monitor 291
 - Übersicht 1
 - Überwachung
 - Anwendungsverzögerungen 352
 - Arbeit 241
 - Daten 260
 - Datenerfassung 319
 - Echtzeitüberwachung 241
 - Ereignismonitore 254
 - Serviceklassenschichten 274
 - Systemverhalten auf verschiedenen Ebenen (Beispiel) 248
 - Übersicht 241
 - UOW
 - Workloadzuordnung bei mehreren Workloads (Beispiel) 47
 - veränderliche CPU-Anteile
 - Details 208
 - Verbindungen
 - Status in Serviceklassen 83
 - zu Workloads zuordnen 24
 - Verwaltungsphase 71, 99, 128
 - WLMADM (Berechtigung)
 - Übersicht 4
 - Workload-Manager-Dispatcher
 - CPU-Grenzwert 214
 - feste CPU-Anteile 200
 - minimale CPU-Auslastung 230
 - veränderliche CPU-Anteile 208
 - Workloads
 - aktivieren 35
 - ändern 33
 - Datenbankzugriff verhindern 38
 - Datenbankzugriffsberechtigung 37
 - erstellen 31
 - inaktivieren 36
 - Leistungsminderung analysieren (Beispiel) 372
 - löschen 36
 - Standardworkloads 26
 - Übersicht 19
 - Zuordnung (Beispiel) 41
 - Zuordnung bei mehreren Workloads (Beispiel) 47
 - Zuordnung bei Workloadattributen mit mehreren Werten (Beispiel) 50
 - Zuordnung einer Verbindung zur Standardverwaltungsworkload 29
 - Zugriffsrecht USAGE für Workloads
 - erteilen 39
 - widerrufen 40
- DDL
 - Anweisungen
 - DB2-Workload-Management 18
- Designadvisor
 - Aktivitätsinformationen importieren 299
- Dispatcher-Thread-Parallelität
 - festlegen 235
- Dokumentation
 - gedruckt 550
 - Nutzungsbedingungen 558

- Dokumentation (*Forts.*)
 - PDF-Dateien 550
 - Übersicht 549

E

- Eigentumsrecht
 - DB2-Workload-Management-Objekte 18
- Ereignismonitore
 - Aktivitäten
 - Daten erfassen 297
 - DB2-Workload-Management
 - Statistikerfassung 286
 - Schwellenwertverstöße
 - Überwachung 292
 - Typen 254
- ESTIMATEDSQLCOST (Aktivitätsschwellenwert) 147

F

- Fehlerbehebung
 - Lernprogramme 557
 - Onlineinformationen 557
- Fehlerbestimmung
 - Lernprogramme 557
 - verfügbare Informationen 557
- Feste CPU-Anteile
 - aktivieren und festlegen 212
- Funktionen
 - Tabelle
 - WLM_GET_ACTIVITY_DETAILS 398
 - WLM_GET_QUEUE_STATS 404
 - WLM_GET_SERVICE_CLASS_AGENTS 407
 - WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES 415
 - WLM_GET_SERVICE_SUBCLASS_STATS 418
 - WLM_GET_SERVICE_SUPERCLASS_STATS 425
 - WLM_GET_WORK_ACTION_SET_STATS 427
 - WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES 428
 - WLM_GET_WORKLOAD_STATS 433

G

- Gemeinsamer Threadzugriff für Workload-Manager-Dispatcher (Konfigurationsparameter) 521
- Gesamtwartezeit in Dispatcherwarteschlange
 - Monitorelemente
 - total_disp_run_queue_time 497
- Gesamtzahl beendeter UOWs
 - Monitorelemente
 - uow_completed_total 499
- Geschätztes CPU-Nutzungsrecht
 - Monitorelemente
 - estimated_cpu_entitlement 470
- Gesicherte Verbindung
 - Übersicht 545
- Gesicherter Kontext
 - Übersicht 545
- Gespeicherte Prozeduren
 - WLM_CANCEL_ACTIVITY 263
 - WLM_CAPTURE_ACTIVITY_IN_PROGRESS 263
 - WLM_COLLECT_STATS 263
 - WLM_SET_CLIENT_INFO 263

H

- Hilfe
 - SQL-Anweisungen 552
- Histogramme
 - Beispiel 284
 - Monitorelemente
 - histogram_type 470
 - number_in_bin 473
 - oberer Grenzwert 496
 - Übersicht 276
- Histogrammschablonen
 - ändern 283, 289
 - erstellen 282
 - löschen 283

K

- Katalogsichten
 - HISTOGRAMTEMPLATEBINS 524
 - HISTOGRAMTEMPLATES 524
 - HISTOGRAMTEMPLATEUSE 525
 - SERVICECLASSES 525
 - THRESHOLDS 528
 - WORKACTIONS 531
 - WORKACTIONSETS 533
 - WORKCLASSES 534
 - WORKCLASSSETS 534
 - WORKLOADAUTH 535
 - WORKLOADCONNATTR 535
 - WORKLOADS 536
- Kennung
 - Monitorelemente
 - arm_correlator 446
 - bin_id 446
 - db_work_action_set_id 468
 - db_work_class_id 469
 - sc_work_action_set_id 483
 - sc_work_class_id 483
 - service_class_id 484
 - work_action_set_id 512
 - work_class_id 513
- Konfigurationsparameter
 - wlm_collect_int 519
 - wlm_disp_concur 521
 - wlm_disp_cpu_shares 522
 - wlm_disp_min_util 523
 - wlm_dispatcher 520

L

- Leistung
 - DB2-Workload-Management
 - Beispiele 385, 387
 - Leistungsmodellierung 301
- Lernprogramme
 - Fehlerbehebung 557
 - Fehlerbestimmung 557
 - Liste 557
 - pureXML 557
- Linux
 - Workload-Management-Integration in DB2-Workload-Management 312

M

- Messdaten
 - DB2-Workload-Management-Objekte 290
- Minimale CPU-Auslastung
 - festlegen 233
- Minimale CPU-Auslastung des Workload-Manager-Dispatchers (Konfigurationsparameter) 523
- Momentaufnahmeüberwachung
 - Tabellenfunktionen ergänzen 291
- Monitorelemente
 - concurrentdbcoordactivities_db_threshold_value 450
 - concurrentdbcoordactivities_db_threshold_violated 451
 - concurrentdbcoordactivities_subclass_threshold_id 451
- Monitorelemente für Aktivitäten
 - Abfragen
 - queue_assignments_total 475
 - queue_size_top 475
 - queue_time_total 476
 - Abschnitte
 - section_env 484
 - act_exec_time 440
 - act_remapped_in 440
 - act_remapped_out 441
 - agg_temp_tablespace_top 446
 - Aktivierungszeit
 - last_wlm_reset 471
 - Aktivitäten
 - act_throughput 442
 - act_total 442
 - activity_collected 443
 - activity_id 444
 - activity_secondary_id 444
 - activity_type 445
 - coord_act_aborted_total 457
 - coord_act_completed_total 457
 - coord_act_rejected_total 464
 - parent_activity_id 473
 - Anweisungen
 - stmt_invocation_id 488
 - Auslastungen
 - wlo_completed_total 512
 - workload_id 514
 - workload_name 515
 - workload_occurrence_id 517
 - workload_occurrence_state 517
 - Bereiche
 - untere Grenze 447
- CONCURRENTDBCOORDACTIVITIES (Schwellenwert)
 - concurrentdbcoordactivities_wl_was_threshold_id 455
 - concurrentdbcoordactivities_wl_was_threshold_queued 455
 - concurrentdbcoordactivities_wl_was_threshold_value 456
 - concurrentdbcoordactivities_wl_was_threshold_violated 456
 - concurrentdbcoordactivities_db_threshold_id 449
 - concurrentdbcoordactivities_subclass_threshold_value 452
 - concurrentdbcoordactivities_subclass_threshold_queued 451
 - concurrentdbcoordactivities_subclass_threshold_violated 452
 - concurrentdbcoordactivities_superclass_threshold_id 453
 - concurrentdbcoordactivities_superclass_threshold_queued 453

Monitorelemente für Aktivitäten (Forts.)

- concurrentdbcoordactivities_superclass
 - _threshold_value 454
- concurrentdbcoordactivities_superclass
 - _threshold_violated 454
- coord_act_est_cost_avg 458
- coord_act_exec_time_avg 459
- coord_act_interarrival_time_avg 460
- coord_act_lifetime_avg 461
- coord_act_queue_time_avg 463
- destination_service_class_id 469
- Histogramme
 - histogram_type 470
 - number_in_bin 473
 - oberer Grenzwert 496
- Höchstwerte
 - act_cpu_time_top 439
 - act_rows_read_top 441
 - concurrent_act_top 447
 - concurrent_connection_top 448
 - concurrent_wlo_act_top 448
 - concurrent_wlo_top 449
 - coord_act_lifetime_top 462
 - cost_estimate_top 465
 - rows_returned_top 482
 - temp_tablespace_top 489
 - uow_total_time_top 507
- Kennung
 - arm_correlator 446
 - bin_id 446
 - db_work_action_set_id 468
 - db_work_class_id 469
 - sc_work_action_set_id 483
 - sc_work_class_id 483
 - service_class_id 484
 - work_action_set_id 512
 - work_class_id 513
- Namen
 - service_subclass_name 485
 - service_superclass_name 486
 - work_action_set_name 513
 - work_class_name 514
- num_remaps 472
- Partitionen
 - coord_partition_num 465
- Protokollspeicherbereich
 - uow_log_space_used 503
- request_exec_time_avg 476
- Routinen
 - routine_id 477
- Schwellenwerte
 - num_threshold_violations 472
 - thresh_violations 490
 - threshold_action 491
 - threshold_domain 492
 - threshold_maxvalue 493
 - threshold_name 493
 - threshold_predicate 493
 - threshold_queue_size 494
 - thresholdid 495
- source_service_class_id 487
- Sperren
 - uow_lock_wait_time 503
- Units of Work (UOWs)
 - uow_throughput 506
- UOWs (Units of Work)
 - parent_uow_id 474

Monitorelemente für Aktivitäten (Forts.)

- UOWs (Units of Work) (Forts.)
 - uow_comp_status 500
 - uow_completed_total 499
 - uow_elapsed_time 500
 - uow_id 501
 - uow_lifetime_avg 502
 - uow_start_time 504
 - uow_status 505
 - uow_stop_time 505
- WLM-Dispatcher
 - cpu_limit 466
 - cpu_share_type 466
 - cpu_shares 466
 - cpu_utilization 466
 - cpu_velocity 467
 - estimated_cpu_entitlement 470
 - total_disp_run_queue_time 497
- Workload-Management
 - Übersicht 439
 - wl_work_action_set_id 508
 - wl_work_class_id 508
 - wlm_queue_assignments_total 509
 - wlm_queue_time_total 510
- Zeilen
 - rows_fetched 478
 - rows_modified 478
 - rows_returned 480
- Zeit
 - prep_time 474
 - time_completed 495
 - time_created 496
 - time_of_violation 496
 - time_started 496
- Zeitmarken
 - activate_timestamp 443
 - statistics_timestamp 488
- Monitorelemente für Höchstwerte
 - act_cpu_time_top 439
 - act_rows_read_top 441
 - concurrent_act_top 447
 - concurrent_connection_top 448
 - concurrent_wlo_act_top 448
 - concurrent_wlo_top 449
 - coord_act_lifetime_top 462
 - cost_estimate_top 465
 - rows_returned_top 482
 - temp_tablespace_top 489
 - uow_total_time_top 507

N

Namen

- Monitorelemente
 - service_subclass_name 485
 - service_superclass_name 486
 - work_action_set_name 513
 - work_class_name 514

Namenskonventionen

- allgemein 541

P

partitionierte Datenbanken

- Monitorelemente
 - coord_partition_num 465

- Pfade
 - Namenskonventionen 541
- Protokolle
 - Monitorelemente
 - uow_log_space_used 503
- Prozeduren
 - WLM_CANCEL_ACTIVITY 393
 - WLM_CAPTURE_ACTIVITY_IN_PROGRESS 394
 - WLM_COLLECT_STATS 396
 - WLM_SET_CLIENT_INFO 436
- Pufferpoolpriorität
 - Serviceklassen 82

R

- REMAP ACTIVITY (Aktion)
 - Beispielscripts 166
 - definieren 161
- Rollen
 - Details 543
- routine_id (Monitorelement) 477
- Routinen
 - Monitorelemente
 - routine_id 477
 - WLM_CANCEL_ACTIVITY (Beispiel) 376

S

- Schemata
 - Anweisung CALL klassifizieren 55
- Schwellenwerte
 - ACTIVITYTOTALTIME 143
 - AGGSQLTEMPSPACE
 - Details 153
 - Aktion 128
 - Aktivität 142
 - Aktivitäten neu zuordnen 161, 172
 - Aktivitätenwarteschlange 152
 - ändern 137
 - Arbeitsaktionen 118
 - Arbeitsklassifikationen 65
 - Beispiele
 - Arbeitsaktionsset und Datenbankschwellenwert 125
 - Verwendung 139
 - Bereichsauflösung von Aktivitätsschwellenwerten 142
 - Bewertungsreihenfolge 134
 - CONCURRENTDBCOORDACTIVITIES 153
 - CONCURRENTWORKLOADACTIVITIES 155
 - CONCURRENTWORKLOADOCCURRENCES 157
 - CONNECTIONIDLETIME 141
 - CPUTIME
 - Details 144
 - CPUTIMEINSC 145
 - DATATAGINSC 146
 - Domäne
 - Übersicht 132
 - erstellen 136
 - ESTIMATEDSQLCOST 147
 - löschen 138
 - Monitorelemente
 - num_threshold_violations 472
 - thresh_violations 490
 - threshold_action 491
 - threshold_domain 492
 - threshold_maxvalue 493
 - threshold_name 493

- Schwellenwerte (*Forts.*)
 - Monitorelemente (*Forts.*)
 - threshold_predicate 493
 - threshold_queuesize 494
 - thresholdid 495
 - SQLROWSREAD
 - Details 148
 - SQLROWSREADINSC 149
 - SQLROWSRETURNED 150
 - SQLTEMPSPACE 151
 - TOTALMEMBERCONNECTIONS 158
 - TOTALSCMEMBERCONNECTIONS 159
 - Übersicht 128
 - Umsetzungsbereich 132
 - UOW (Unit of Work) 160
 - UOWTOTALTIME 160
 - Verbindung 141
 - Verstöße überwachen 292
 - Zusammenfassung 128, 152
- Schwellenwertverstöße
 - E-Mail-Benachrichtigungen 294
- Schwellenwertverstöße (Ereignismonitor) 254
- Serviceklassen
 - Agentenpriorität 80
 - Aktivitätszustände 83
 - ändern
 - Änderungen bei Zurücksetzung von Statistikdaten 289
 - Prozedur 87
 - Beispiele 92, 97
 - erstellen 85
 - Leistungsminderung analysieren 97
 - löschen 90
 - nicht verfolgte Entitäten 85
 - Pufferpoolpriorität 82
 - Standardservicesuperklassen 75
 - Standardserviceunterklassen 75
 - Verbindungsstatus 83
 - Vorablesezugriffspriorität 81
 - Zeitpunktstatistikdaten 251
 - Zuordnung von Aktivitäten 76
- Serviceklasseninterne Schwellenwerte
 - Prioritätssteuerung nach Verweildauer 161
- Servicesuperklassen
 - ändern 87
 - erstellen 85
 - löschen 90
 - Überwachungsdaten 260
- Serviceunterklassen
 - ändern 87
 - erstellen 85
 - löschen 90
 - Überwachungsdaten 260
- SET WORKLOAD (Befehl)
 - Details 518
 - Verbindung der Standardverwaltungsworkload zuordnen 29
- Sicherheit
 - gesicherter Kontext 545
- Sperren
 - Monitorelemente
 - uow_lock_wait_time 503
- SQL-Anweisungen
 - Hilfe
 - anzeigen 552
 - Monitorelemente
 - stmt_invocation_id 488

- sqleseti-API
 - Workloadzuordnung 41
- SQLROWSREAD (Aktivitätsschwellenwert)
 - Details 148
- SQLROWSREADINSC (Aktivitätsschwellenwert) 149
- SQLROWSRETURNED (Aktivitätsschwellenwert) 150
- SQLTEMPSPACE (Aktivitätsschwellenwert)
 - Details 151
- Standardbenutzerservicesuperklasse 75
- Standardserviceunterklassen 75
- Standardsystemservicesuperklasse 75
- Standardwartungsservicesuperklasse
 - Übersicht 75
- Standardworkloads 26
- Statistikdaten
 - DB2-Workload-Management-Objekte 264
 - Ereignismonitor 254
 - Erfassung
 - Workload-Management 286
- SYSDEFAULTMAINTENANCECLASS (Servicesuperklasse)
 - Übersicht 75
- SYSDEFAULTSYSTEMCLASS (Servicesuperklasse)
 - Übersicht 75
- SYSDEFAULTUSERCLASS (Servicesuperklasse)
 - Übersicht 75
- Szenarios
 - abbrechen
 - Aktivitäten 379
 - Aktivitätsmessdaten zusammenfassen
 - über untergeordnete Aktivitäten hinweg 373

T

- Tabellenbereiche
 - SQLTEMPSPACE (Schwellenwert) 151
- Tabellenfunktionen
 - auf unterschiedlichen Ebenen überwachen
 - Beispiel 248
 - Daten zusammenfassen 252
 - Informationen der WLM-Schwellenwertwarteschlange ermitteln
 - Beispiel 253
 - Nutzungsbeispiel 246
 - Snapshot Monitor 291
 - WLM_COLLECT_STATS 289
 - WLM_GET_SERVICE_CLASS_AGENTS 407
 - WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES 415
 - WLM_GET_SERVICE_SUBCLASS_STATS 418
 - WLM_GET_WORKLOAD_STATS 433
- TOTALMEMBERCONNECTIONS (Verbindungsschwellenwert) 158
- TOTALSCMEMBERCONNECTIONS (Verbindungsschwellenwert) 159

U

- Überwachung
 - Daten
 - Workload-Management 260
 - Echtzeitüberwachung 241
 - nicht von Serviceklassen verfolgte Entitäten 85
 - Prioritätssteuerung nach Verweildauer 274
 - Protokollstatistiken 254
 - Serviceklassenschichten 274
 - Übersicht 241

- Überwachung (*Forts.*)
 - Workload-Manager-Dispatcher
 - Details 236
 - Leistung 236
 - Workloadtypen 236
- UOW
 - mit Workload abgleichen
 - Beispiel 47
 - Schwellenwerte 160
 - zur Standardworkload zuordnen 26
- UOW-Durchsatz
 - Monitorelemente
 - uow_throughput 506
- UOW-Lebensdauer, Durchschnitt
 - Monitorelemente
 - uow_lifetime_avg 502
- UOWs (Units of Work)
 - Monitorelemente
 - parent_uow_id 474
 - uow_comp_status 500
 - uow_completed_total 499
 - uow_elapsed_time 500
 - uow_id 501
 - uow_lifetime_avg 502
 - uow_lock_wait_time 503
 - uow_log_space_used 503
 - uow_start_time 504
 - uow_status 505
 - uow_stop_time 505
 - uow_throughput 506
- UOWTOTALTIME (Schwellenwert) 160

V

- Veränderliche CPU-Anteile
 - aktivieren und festlegen 212
- Verbindungen
 - Status in Serviceklassen 83
 - temporär 159
 - zu Workloads zuordnen 45
 - zuordnen
 - Standardverwaltungsworkload 29
 - Workload-Verbindungsattribute mit mehreren Werten 50
 - Workloads 24
- Vorablesezugriff
 - Priorität von Serviceklassen 81

W

- Warteschlangen
 - Vorablesezugriff 81
- Widerrufen
 - Zugriffsrecht USAGE für Workload 40
- WLM_CANCEL_ACTIVITY (Prozedur) 393
- WLM_CAPTURE_ACTIVITY_IN_PROGRESS (Prozedur) 394
- wlm_collect_int (Datenbankkonfigurationsparameter) 519
- WLM_COLLECT_STATS (Prozedur)
 - Details 396
 - Statistikdaten zurücksetzen 289
- wlm_disp_concur (Konfigurationsparameter) 521
- wlm_disp_concur (Konfigurationsparameter des Datenbankmanagers)
 - festlegen 235
- wlm_disp_cpu_shares (Konfigurationsparameter) 522
- wlm_disp_min_util (Konfigurationsparameter) 523

- wlm_disp_min_util (Konfigurationsparameter des Datenbankmanagers)
 - festlegen 233
 - wlm_dispatcher (Konfigurationsparameter) 520
 - wlm_dispatcher (Konfigurationsparameter des Datenbankmanagers)
 - festlegen 198
 - WLM_GET_ACTIVITY_DETAILS (Tabellenfunktion) 398
 - WLM_GET_QUEUE_STATS (Tabellenfunktion) 404
 - WLM_GET_SERVICE_CLASS_AGENTS (Tabellenfunktion)
 - Agentennutzung nach Serviceklasse untersuchen (Beispiel) 98
 - Details 407
 - Informationen der WLM-Warteschlange
 - Beispiel 253
 - WLM_GET_SERVICE_CLASS_WORKLOAD_OCCURRENCES (Tabellenfunktion)
 - Beispiele
 - Daten zusammenfassen 252
 - Details 415
 - WLM_GET_SERVICE_SUBCLASS_STATS (Tabellenfunktion)
 - Beispiele
 - Daten zusammenfassen 252
 - Leistungsminderung analysieren 97, 372
 - Zeitpunktstatistikdaten abrufen 251
 - Details 418
 - WLM_GET_SERVICE_SUPERCLASS_STATS (Tabellenfunktion) 425
 - WLM_GET_WORK_ACTION_SET_STATS (Tabellenfunktion)
 - Details 427
 - Workloads analysieren (Beispiele) 66
 - WLM_GET_WORKLOAD_OCCURRENCE_ACTIVITIES (Tabellenfunktion)
 - Beispiele
 - lange laufende Aktivitäten ermitteln 376
 - Beschreibung 428
 - Daten zusammenfassen (Beispiele) 252
 - WLM_GET_WORKLOAD_STATS (Tabellenfunktion) 433
 - WLM_SET_CLIENT_INFO (Prozedur) 436
 - WLMADM (Berechtigung)
 - Details 4
 - Workload-Management
 - Beispiele
 - Abbrechen aller Aktivitäten 382
 - Trennen der Verbindung aller Anwendungen 383
 - Schwellenwertverstöße
 - E-Mail-Benachrichtigungen 294
 - Workload-Management - Dispatcher
 - Monitorelemente
 - cpu_limit 466
 - cpu_share_type 466
 - cpu_shares 466
 - cpu_utilization 466
 - cpu_velocity 467
 - estimated_cpu_entitlement 470
 - total_disp_run_queue_time 497
 - Workload-Management-Dispatcher (Konfigurationsparameter) 520
 - Workload-Manager-Dispatcher
 - AIX
 - Mikropartitionsumgebungen 197
 - aktivieren 198
 - CPU-Anteile
 - aktivieren 212
 - festlegen 212
 - CPU-Grenzwert 214
 - festlegen 228
 - Workload-Manager-Dispatcher (Forts.)
 - Details 176
 - Dispatcher-Thread-Parallelität 234
 - festlegen 235
 - feste CPU-Anteile 200
 - maximale Planungsgenauigkeit sicherstellen 199
 - Mikropartitionsumgebungen 197
 - minimale CPU-Auslastung 230
 - festlegen 233
 - Übersicht 175
 - Überwachung
 - Details 236
 - Umgebungen mit mehreren Mitgliedern
 - Verhalten 197
 - veränderliche CPU-Anteile 208
 - Workloads
 - aktivieren 35
 - ändern 33
 - Arbeitsaktionsset (Vergleich) 122
 - Beispiele
 - Leistungsminderung analysieren 372
 - Zuordnung bei mehreren Workloads 47
 - Zuordnung bei Workloadattributen mit Einzelwerten 45
 - Zuordnung bei Workloadattributen mit mehreren Werten 50
 - Bewertungsreihenfolge 24
 - Datenbankzugriff erlauben 37
 - Datenbankzugriff verhindern 38
 - erstellen 31
 - inaktivieren 36
 - löschen 36
 - Position in Workloadliste 24
 - Standardworkloads 26
 - Übersicht 19
 - Überwachungsdaten 260
 - USAGE (Zugriffsrecht)
 - erteilen 39
 - widerrufen 40
 - Zuordnung
 - Beispiele 41
 - Details 24
 - Zuordnung einer Verbindung zur Standardverwaltungsworkload 29
- ## Z
- Zeilen
 - Monitorelemente
 - rows_fetched 478
 - rows_modified 478
 - rows_returned 480
 - rows_returned_top 482
 - Zeit
 - Monitorelemente
 - prep_time 474
 - time_completed 495
 - time_created 496
 - time_of_violation 496
 - time_started 496
 - Zeitmarken
 - Monitorelemente
 - activate_timestamp 443
 - statistics_timestamp 488
 - uow_start_time 504
 - uow_stop_time 505

Zugriffsrechte
Rollen 543
Zusammenfassung
Daten
Beispiel 252
Zusammengefasste Schwellenwerte
Übersicht 152



SC12-4683-01



Spine information:

IBM DB2 10.1 for Linux, UNIX and Windows

DB2 Workload Management - Handbuch und Referenz

