



## DB2 for z/OS Storage Tuning

Nigel Slinger

IBM Distinguished Engineer

DB2 for z/OS Development and Technical Support



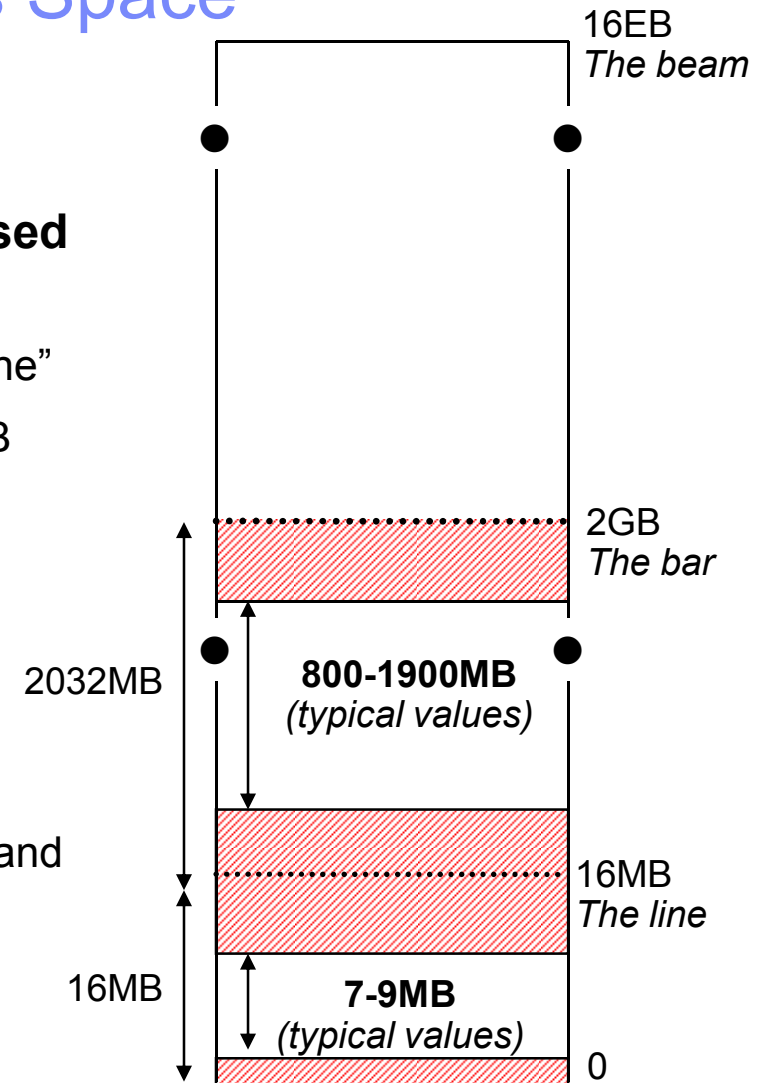
## Why is storage tuning important

- **Leading cause of DB2 outages**
- **If storage is limited, extra DB2 members may be required in the datasharing group**



## Fitting DB2 in the DBM1 Address Space

- **DB2 DBM1 address space now has an addressing range of 16EB (“the beam”) based on 64-bit addressing but**
  - Maximum of 16MB available “below the 16MB line”
  - Maximum of 2032MB available “above the 16MB line” and “below the 2GB bar”
  
- **Practical maximum available to DB2 and specifically DBM1 AS is much less**
  - Typical 7-9MB available “below the line”
  - Typical 800-1900MB available “above the line” and “below the 2GB bar”





## What is the Problem?

- **Storage is allocated into different subpools which have unique characteristics**
  - Storage acquired via MVS GETMAIN
  - Storage released by MVS FREEMAIN
- **GETMAIN processing by DB2 components using DB2 storage manager**
  - Requests may be conditional or unconditional to DB2 storage manager
  - "Short on Storage" condition can occur for both
  - DB2 recovery routines may be able to clean up
  - Individual DB2 threads (allied, DBAT) may abend with 04E/RC=00E200xx when insufficient storage available
    - e.g. 00E20003 & 00E20016
  - Eventually DB2 subsystem may abend with abend S878 or S80A due to non DB2 subsystem component (e.g DFP) issuing unconditional MVS getmain
    - DB2 getmains are MVS conditional getmains so are converted to DB2 abends e.g. 00E20016



# Tracking DB2 Storage

## ▪ RMF for high level

- Virtual Storage (VSTOR) Private Area Report
  - Interval data collected in SMF Type 78-2
  - Collected by RMF Monitor I session option: `VSTOR(D,xxxxDBM1)`
  - Produced by RMF Post Processor option: `REPORTS(VSTOR(D,xxxxDBM1))`

## ▪ IFC Records

- IFCID 225
  - Storage Summary
  - Snapshot value as each DB2 Stats interval comes due (ZPARM = STATIME)
  - Now included in Statistics Trace Class 1
- IFCID 217
  - Storage Detail Record at thread level
  - Effectively a dump SM=1 report but in IFC form
  - Available through Global Trace Class 10



## Tracking DB2 Storage ...

### ■ IFC Records ...

- First class support provided by OMEGAMON XE for DB2 PM/PE, DB2 PM and DB2 PE
  - Statistics Trace | Report
    - Includes FILE and LOAD data base table support as well as upgrade (ALTER TABLE ....) of already installed table DB2PM\_STAT\_GENERAL
  - Record Trace Report
- New SPREADSHEETDD subcommand option
  - Both DB2PE V2.1 & DB2PM V8.1 via APAR PK31073
  - OMEGAMON XE for DB2 PE V3 & V4 via APARs PK33395 & PK33406
- REXX Tools (MEMU2, MEMUSAGE)
  - Available for download from Developer Works  
[http://www.ibm.com/developerworks/exchange/dw\\_entryView.jspa?externalID=242&categoryID=33](http://www.ibm.com/developerworks/exchange/dw_entryView.jspa?externalID=242&categoryID=33)



## Tracking DB2 Storage ...

- **V8 APAR PK20800 8/07**

- DISPLAY THREAD(\*) SERVICE(STORAGE)
- DSNV492I message that can be used by DB2 service for diagnostics

```
V91A  N *  0 003.RCRSC 02 SYSOPR      0067  0
V490-SUSPENDED 07213-09:59:18.02 DSNRCRSC +00000230 01.51
V492-LONG 252 K VLONG 40 K 64 1028 K
```

- Includes Agent Local Non-System Storage usage
- Does not include Getmained Stack Storage usage
- The key values are the **LONG** storage pool and the **VLONG** storage pool values (252KB + 40KB = 292KB in previous example)
  - Reflect virtual storage consumption below the 2GB bar
  - May be used to identify poorly behaved applications or DB2 code issues



# MVS Storage Overview

- **EXTENDED REGION SIZE (MAX) – QW0225RG**
  - Total theoretical amount DB2 has access to
- **31 BIT EXTENDED LOW PRIVATE – QW0225EL**
  - DB2 uses a small amount of Low private (bottom up storage)
    - DB2 code itself / reservation for pageset storage
- **31 BIT EXTENDED HIGH PRIVATE – QW0225EH**
  - DB2 mostly uses subpool 229 Key 7 (top down storage)
- **Other products also use address space storage**
  - Dataset opens / DFP
  - SMF

DBM1 AND MVS STORAGE BELOW 2 GB	CONTINUED	QUANTITY
24 BIT LOW PRIVATE	(MB)	0.14
24 BIT HIGH PRIVATE	(MB)	0.47
31 BIT EXTENDED LOW PRIVATE	(MB)	130.86
31 BIT EXTENDED HIGH PRIVATE	(MB)	859.40
EXTENDED REGION SIZE (MAX)	(MB)	1296.00
EXTENDED CSA SIZE	(MB)	632.48





## MVS Storage Overview ...

### ▪ **ECSA – QW0225EC**

- Common storage area across all address spaces for a given LPAR
- Large ECSA size would be 1GB with typical sizes being 300-500MB
- Affects maximum available Extended Region
  - Biggest factor
- Some customers due to the needs of other products have huge ECSA leading to very small extended region size
  - Extensive use of ECSA by IMS across dependent regions
    - Mostly buffer pools, control blocks, data are in ECSA
    - Sizes are at user choice – For best performance they tend to be large
    - Not exploiting VSCR features of recent IMS releases
- Generous over allocation for safety of ECSA and other common areas
- Common LPAR image for Sysplex (best practice)



## MVS Storage Overview ...

- **REGION parameter in JCL for ASID**
  - MVS rule is high private cannot grow into low private
  - Region parameter controls how high low-private can grow
    - 0M means all available region can be allocated, could be limited by exit
    - No effect on getmained DB2 storage since DB2 only allocates high private storage
  - Region parameter can be used to save a few MB by forcing some low private growth into 24-bit low private (QW0225LO)
    - Some dataset open activity can be in trouble with a low REGION= parm
  - Usually REGION=0M is preferred
- **MEMLIMIT**
  - Not observed by DB2 DBM1
  - Overridden by DB2 to 4TB



# DB2 DBM1 Address Space Storage

- **31-bit / 24-bit DB2 storage**

- Getmained
- Variable
- Fixed Storage
- Stack storage

- **Non-DB2 getmained**

- SMF
- Dataset / pageset

DBM1 AND MVS STORAGE BELOW 2 GB		QUANTITY
-----		-----
<b>TOTAL DBM1 STORAGE BELOW 2 GB</b>	(MB)	616.67
<b>TOTAL GETMAINED STORAGE</b>	(MB)	137.96
EDM POOL	(MB)	120.00
<b>TOTAL VARIABLE STORAGE</b>	(MB)	379.23
TOTAL AGENT LOCAL STORAGE	(MB)	279.39
TOTAL AGENT SYSTEM STORAGE	(MB)	187.10
NUMBER OF PREFETCH ENGINES		600.00
NUMBER OF DEFERRED WRITE ENGINES		300.00
NUMBER OF CASTOUT ENGINES		300.00
NUMBER OF GBP WRITE ENGINES		300.00
NUMBER OF P-LOCK/NOTIFY EXIT ENGINES		202.00
TOTAL AGENT NON-SYSTEM STORAGE	(MB)	92.29
TOTAL NUMBER OF ACTIVE USER THREADS		101.12
RID POOL	(MB)	1.13
PIPE MANAGER SUB POOL	(MB)	7.97
LOCAL DYNAMIC STMT CACHE CNTL BLKS	(MB)	0.99
THREAD COPIES OF CACHED SQL STMTS	(MB)	16.24
IN USE STORAGE	(MB)	0.61
STATEMENTS COUNT		82.77
HWM FOR ALLOCATED STATEMENTS	(MB)	0.78
STATEMENT COUNT AT HWM		122.00
DATE AT HWM		02/02/09
TIME AT HWM		10:24:37.33
<b>TOTAL FIXED STORAGE</b>	(MB)	2.61
<b>TOTAL GETMAINED STACK STORAGE</b>	(MB)	96.87
TOTAL STACK STORAGE IN USE	(MB)	49.35
STORAGE CUSHION	(MB)	130.32



## DB2 DBM1 Address Space Storage ...

- **Getmained - QW0225GM**
  - Compression Dictionaries and Bufferpools are now above the bar
  - EDM pool still getmained
- **Variable Storage - QW0225VR**
  - Most interesting from a tuning perspective
  - Variable length blocks
  - Thread pools (AGL)
    - Used by both System and User
  - Local Dynamic Statement Cache
- **Fixed Storage - QW0225FX**
  - High performance storage
  - Fixed length blocks
  - Not usually so interesting from a tuning perspective
    - Small change in the great scheme of things
- **Stack Storage - QW0225GS**
  - Save areas
  - Working program variables
  - Small amounts of high speed storage allocations
  - Cached in the DB2 address space to allow greater performance
  - Compressed only at full system contraction
  - Stacks are bigger in V8 (64K vs 20K)



## Non-DB2 Storage

- **Not tracked by DB2**
- **Non-DB2 storage is high private storage**
  - **TOTAL DBM1 STORAGE** = TOTAL GETMAINED STORAGE QW0225GM + TOTAL GETMAINED STACK STORAGE QW0225GS + TOTAL FIXED STORAGE QW0225FX + TOTAL VARIABLE STORAGE QW0225VR
  - **NON-DB2 STORAGE** = MVS 31 BIT EXTENDED HIGH PRIVATE QW0225EH – TOTAL DB2 DBM1 STORAGE
- **Used usually by MVS functions such as SMF**
- **Parameter DETAIL in SMFPRMxx can cause storage to creep and become very large**
  - The big hit to DB2 in this area is the DDNAME tracking: allocation does not realise that we have closed off a page set and reallocated it again
  - SMF Type 30 subtype 4 and 5 will track all the DDNAMES
  - Most environments do not need SMF Type 30 subtype 4 and 5
  - Recommend NODETAIL



# DB2 DBM1 Address Space Storage ...

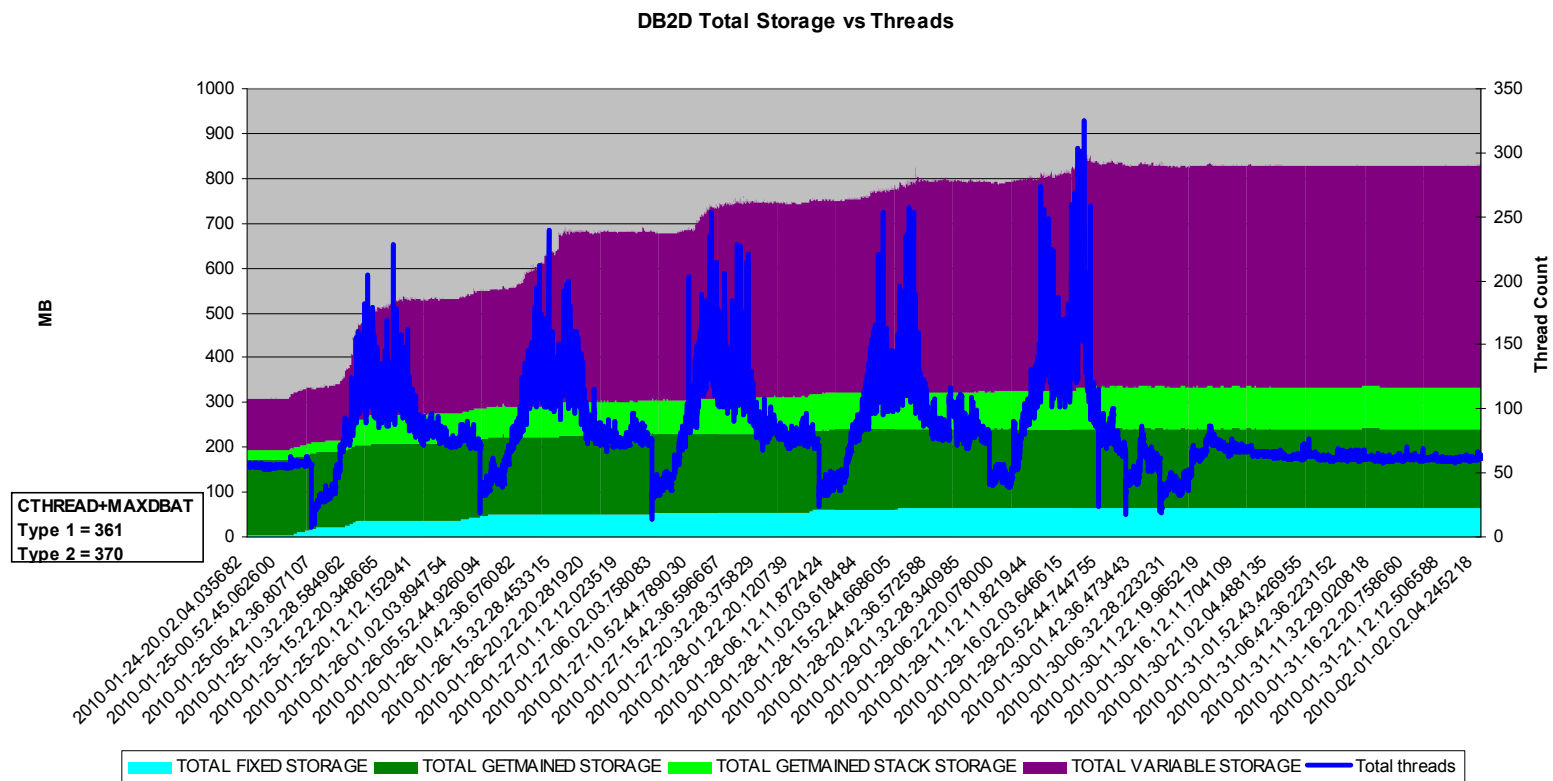
## 64-bit DB2 storage

- Getmained
  - Fixed
  - Variable
  - Compression Dictionaries
  - DBD Pool
  - Dynamic Statement Cache
  - RDS Pool Above (V9)
  - Skeleton Pool (V9)
- Buffer Control Blocks
- Castout Buffers
- 64-bit Shared Private Storage (V9)
- Virtual Buffer Pools are not managed by DB2 storage manager (not part of Getmained storage). In V7 and below they were managed by DB2 storage manager

DBM1 STORAGE ABOVE 2 GB		QUANTITY
FIXED STORAGE	(MB)	25.99
<b>GETMAINED STORAGE</b>	(MB)	3951.66
COMPRESSION DICTIONARY	(MB)	48.38
IN USE EDM DBD POOL	(MB)	255.69
IN USE EDM STATEMENT POOL	(MB)	93.98
IN USE EDM RDS POOL	(MB)	0.09
IN USE EDM SKELETON POOL	(MB)	96.61
<b>VIRTUAL BUFFER POOLS</b>	(MB)	3460.16
<b>VIRTUAL POOL CONTROL BLOCKS</b>	(MB)	0.91
<b>CASTOUT BUFFERS</b>	(MB)	37.50
VARIABLE STORAGE	(MB)	1255.45
THREAD COPIES OF CACHED SQL STMTS	(MB)	126.59
IN USE STORAGE	(MB)	1.08
HWM FOR ALLOCATED STATEMENTS	(MB)	1.40
<b>SHARED MEMORY STORAGE</b>	(MB)	2579.48
TOTAL FIXED VIRTUAL 64BIT SHARED	(MB)	15.50
TOTAL GETMAINED VIRTUAL 64BIT SHARED	(MB)	1194.41
TOTAL VARIABLE VIRTUAL 64BIT SHARED	(MB)	1369.57



# Basic graphing of storage - leaky subsystem? 7 days data (Mon-Sun)





## Basic graphing

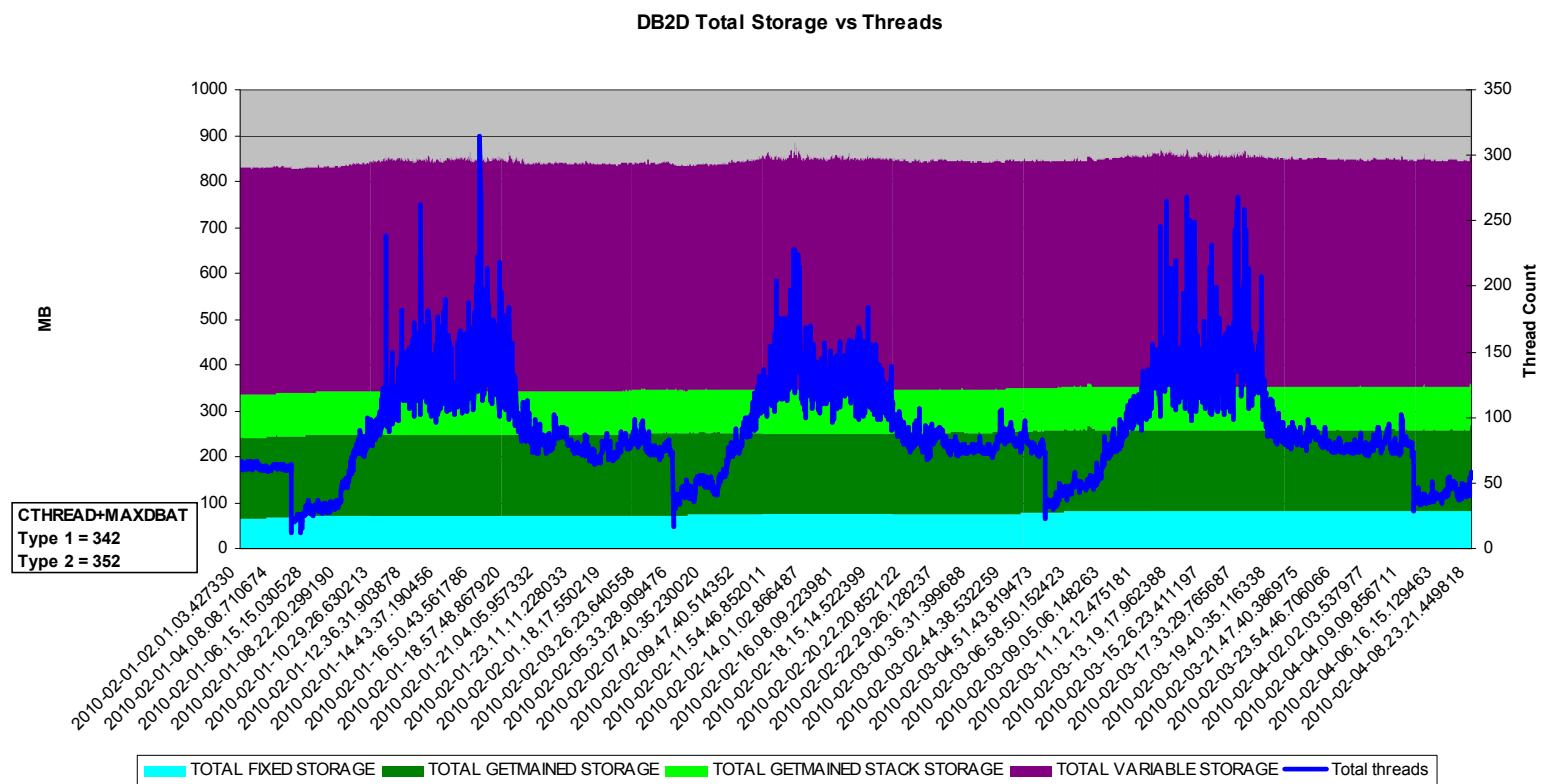
- **Check the major components of DB2 storage**
  - Fixed
  - Getmained
  - Stack
  - Variable
  - Thread counts to give an idea of workload
  
- **Previous graph shows a leak?**
  - Or does it?





# Basic graphing – what happened next

## Mon-Wed This DB2 took a full week to “warm up”





## Lessons to be learned

- **Wrong data time can lead to erroneous conclusion**
  - Full week showed a possible leak, 10 days showed DB2 stable storage usage
  
- **Ideal data is from DB2 startup to DB2 shutdown**
  - If not possible get as much as you can
  
- **Do not think you know where the maximum usage is – you may be way off the mark**
  
- **More data will lead you to a better result**



## How much is left available in the address space

- **QW0225AV – DB2 running total**
  - Possibly inaccurate since DB2 storage manager has no idea about storage obtained by other products directly from z/OS
  
- **What's really left**
  - $QW0225RG - (QW0225EL + QW0225EH)$
  - Numbers directly obtained from z/OS



## Storage Overuse: DB2 Storage Contraction

- **When 'running low' on extended virtual, DB2 begins system contraction process which attempts to freemain any available segments of storage**
  - Contraction can be
    - Normal
    - A sign of a poorly tuned system
- **3 critical numbers for contraction**
  - Storage reserved for must complete (e.g. ABORT, COMMIT) – QW0225CR
    - =  $(CTHREAD+MAXDBAT+1)*\underline{64K}$  (Fixed, real value) **+25M**
  - Storage reserved for open/close of datasets – QW0225MV
    - =  $(DSMAX*1300)+40K$  (space reserved in low private)
  - Warning to contract – QW0225SO
    - = Max (5% of Extended Region Size, **QW0225CR -25M**)
  - Storage Cushion = QW0225CR + QW0225MV + QW0225SO
    - Note: QW0225MV will decrease as pagesets are opened



## Storage Overuse: DB2 Storage Contraction

- Examples:**

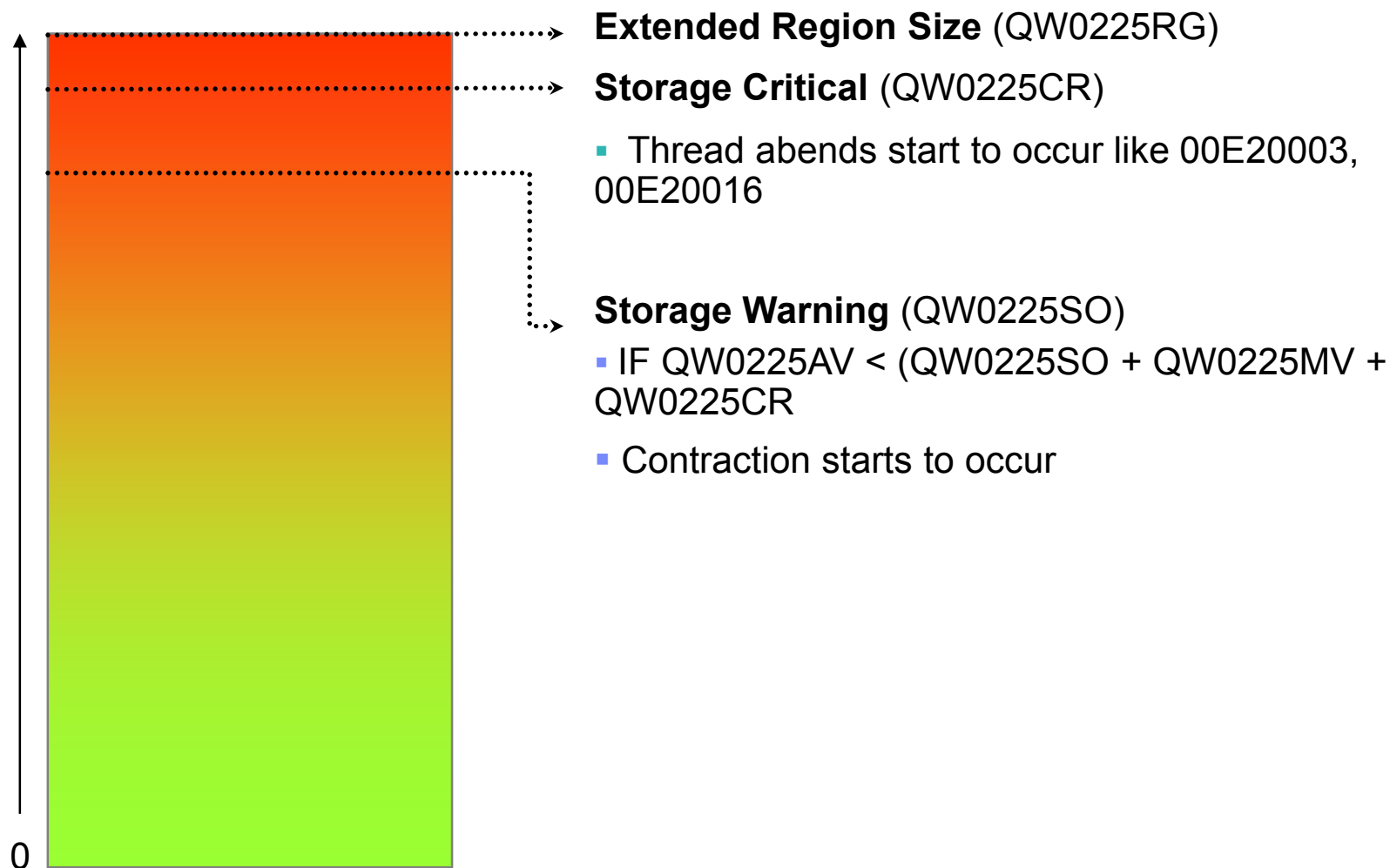
	Case 1	Case 2	Case 3
<b>CTHREAD</b>	<b>2000</b>	<b>400</b>	<b>400</b>
<b>MAXDBAT</b>	<b>2000</b>	<b>2000</b>	<b>150</b>
<b>DSMAX</b>	<b>15000</b>	<b>15000</b>	<b>15000</b>
<b>MVS extended region size (MB)</b>	<b>1700</b>	<b>1000</b>	<b>1000</b>
<b>Storage reserved for must complete (MB)</b>	<b>275</b>	<b>175</b>	<b>63</b>
<b>Storage reserved for datasets (MB)</b>	<b>19</b>	<b>19</b>	<b>19</b>
<b>Warning to contract (MB)</b>	<b>250</b>	<b>150</b>	<b>50</b>
Storage Cushion (MB)	544	344	132

**\*\*WARNING\*\* DO NOT SPECIFY CTHREAD + MAXDBAT TOO HIGH  
IN DB2 V8 OR THE CUSHION WILL BE VERY LARGE**



## Storage cushion

- **QW0225AV reports how much storage DB2 thinks is available**





## Storage Overuse: Large Contributors

- **Stack use (QW0225GS)**
  - Normal range is typically 100-300MB
  - Compressed only at full system contraction
- **System agents (QW0225AS)**
  - Some agents once allocated are never deallocated
    - For example: P-lock engine, prefetch engine
  - # engines: QW0225CE, QW0225DW, QW0225GW, QW0225PF, QW0225PL
    - If these counts are very low and system is on the brink of storage overuse, it is possible that the allocation of more engines could send the system into contraction
- **User threads (QW0225VR-QW0225AS)**
  - Typical user thread storage footprint can be 500KB to 10MB per thread depending on thread persistence, variety and type of SQL used
    - SAP Threads 10MB
    - CICS Threads 500KB
    - Number of threads obtained via QW0225AT + QDSTCNAT



# CONTSTOR

- **Thread storage contraction turned on by ZPARM CONTSTOR=YES**
  - Online changeable with **immediate** effect
  - Only compresses LONG storage (as per SERVICE(STORAGE))
  - Maximum of 1 compress every 5 commits (so very cheap to implement)
- **Associated CPU overhead**
  - Benefit should be carefully evaluated before enabling
  - Ineffective for long-running persistent threads with use of RELEASE(DEALLOCATE)
- **Compresses out part of Agent Local Non-System storage**
  - Does not compress
    - Agent Local System
    - Getmained Stack Storage, LDSC
- **Controlled by two hidden zparms**
  - SPRMSTH @ 1048576 and SPRMCTH @ 10
- **Triggers**
  - No. of Commits > SPRMCTH, or
  - Agent Local Non-System > SPRMSTH and No. of Commits > 5





# MINSTOR

- **Best fit algorithm for thread storage turned on by ZPARM MINSTOR=YES**
  - Online changeable, may not have an effect due to already cached pools
  - Restart recommended if this parm changed
  
- **Changes the storage management of the user AGL POOL to “Best fit” rather than “First fit”**
  - In order to find the best fit piece of storage, CPU cycles are used to scan and maintain ordered storage
  - In a POOL with low fragmentation, MINSTOR may not have a great effect but will cost CPU
  
- **Only enable if fragmentation is a big issue**
  - Only the SM=4 option of the DB2 Dump Formatter and a dump will really give you the definitive answer



## Protecting the System

- **Plan on a 'Basic' storage cushion (free)**
  - To avoid hitting short on storage and driving Full System Contraction
  - To provide some headroom for
    - Tuning, some growth, Fast Log Apply, abnormal operating conditions
  - Basic cushion = Storage cushion + 100M
    - The Basic cushion should be less than 25% of the Extended Region Size, otherwise CTHREAD and/or MAXDBAT are probably set too high

	<i>Case 1</i>	<i>Case 2</i>	<i>Case 3</i>
<b>CTHREAD</b>	<b>2000</b>	<b>450</b>	<b>450</b>
<b>MAXDBAT</b>	<b>2000</b>	<b>2000</b>	<b>200</b>
<b>MVS extended region size (MB)</b>	<b>1700</b>	<b>1000</b>	<b>1000</b>
<b>Storage Cushion (MB)</b>	<b>544</b>	<b>344</b>	<b>132</b>
<b>Basic Cushion (MB)</b>	<b>644</b>	<b>444</b>	<b>232</b>
<b>% of Extended Region Size</b>	<b>37%</b>	<b>44%</b>	<b>23%</b>



## Protecting the System ...

- **Estimate the maximum number of threads that can be supported**
  - Assuming the storage is proportional to the amount of threads, it is possible to predict a theoretical max. number of concurrent threads
  - It may be possible to run the system with more threads than the formula dictates, but there is the danger that the large threads may come in and cause out of storage conditions
- **Set ZPARMS CTHREAD and MAXDBAT to protect the system**
  - CTHREAD and MAXDBAT are the brakes on the DB2 subsystem
    - Theoretical maximum:  $CTHREAD + MAXDBAT = 2000$
    - Practical maximum is much less (typical range 300-850)
  - Avoid over committing resources
  - [Deny service and queue work outside the system to keep system alive](#)



## Estimating Maximum Number of Threads

- **Collect IFCID 225 since the start of DB2, Statistics class 1 (SMF100 and 102 in V8, SMF100 in V9)**
  - Month end processing
  - Weekly processing
  - Utilities processing
  - Try to use a full application mix cycle
- **Focus on time periods with**
  - Increasing number of allied threads + active DBATs
  - Increasing use of getmained stack storage
  - Increasing use of AGL non-system
- **Adjust the formula based on workload variations**
- **Protect the system by always using a pessimistic approach to formulating the numbers**
  - Optimistic may mean a DB2 outage
- **Always recalculate on a regular basis as new workloads and/or parameters are changed**



## Estimating Maximum Number of Threads...

- **Remember to use the MAX impact value across all available data, e.g. MAX system storage**
- **'Basic' storage cushion (BC)**
  - $(BC) = QW0225CR + QW0225MV + QW0225SO + 5\% \text{ of } QW0225RG$
- **Calculate Max non-DB2 storage (ND)**
  - $(ND) = \text{MAX}(\text{MVS 31 BIT EXTENDED HIGH PRIVATE } QW0225EH - \text{TOTAL GETMAINED STORAGE } QW0225GM - \text{TOTAL GETMAINED STACK STORAGE } QW0225GS - \text{TOTAL FIXED STORAGE } QW0225FX - \text{TOTAL VARIABLE STORAGE } QW0225VR)$
- **Max. allowable storage (AS)**
  - $(AS) = QW0225RG - (BC) - (ND)$
- **Max. allowable storage for thread use (TS)**
  - $(TS) = (AS) - \text{MAX}(\text{TOTAL AGENT SYSTEM STORAGE } QW0225AS - \text{TOTAL FIXED STORAGE } QW0225FX - \text{TOTAL GETMAINED STORAGE } QW0225GM - \text{MVS 31 BIT EXTENDED LOW PRIVATE } QW0225EL)$
- **Average thread footprint (TF)**
  - $(TF) = (\text{TOTAL VARIABLE STORAGE } QW0225VR - \text{MAX}(\text{TOTAL AGENT SYSTEM STORAGE } QW0225AS) + \text{TOTAL GETMAINED STACK STORAGE } QW0225GS) / (\text{Allied threads } QW0225AT + \text{DBATs } QDSTCNAT)$
- **Max threads supported = (TS) / (TF)**



## Virtual vs. REAL Storage

- **Important subsystems such as DB2 should not be paging IN from auxiliary storage (DASD)**
  - Recommendation to keep page in rates low (near zero)
  - Monitor using RMF Mon III
- **V8 introduces very large memory objects that may not be backed by REAL storage frames**
  - Virtual storage below 2GB bar is usually densely packed (as before in V7)
    - VIRTUAL=REAL is a fair approximation
  - Virtual storage above the bar number may be misleading
    - Backing rate is low for 64-bit storage
    - No need to back until first reference
  - For an LPAR with greater than 16GB of defined real storage, DB2 will obtain a minimum starting memory object above the bar of 16GB
    - This memory is sparsely populated
    - Virtual will not equal REAL



## Monitoring REAL Storage

REAL AND AUXILIARY STORAGE		QUANTITY
REAL STORAGE IN USE	(MB)	5958.66
AUXILIARY STORAGE IN USE	(MB)	0.00

- **Real storage needs to be monitored as much as Virtual storage**
  - Need to pay careful attention to QW0225RL (Real frames in use by DBM1) and QW0225AX (Auxiliary frames)
    - Ideally QW0225RL should be significantly less than the amount of virtual consumed
- **An indication of either (a) a DB2 code error or (b) an under provisioned system will see**
  - 100% real frames consumed
    - It will be important to know how much real is dedicated to a given LPAR
      - Although a physical machine may have 30GB real, a given LPAR may only have a fraction of this real dedicated
  - An extensive number of auxiliary frames in use
  - Performance degradation
- **V9 – Shared object storage can only be monitored at the LPAR level so is only accurate for a single DB2 LPAR assuming no other exploiters of shared storage**



## Monitoring REAL Storage - Warning

- **Excessive amounts of storage on AUX may cause long DUMP times and severe performance issues.**
  - Paging may become severe
  
- **Make sure enough REAL storage is available in case DB2 has to take a DUMP**
  - DUMP should complete in seconds to make sure no performance problems ensue. Once paging begins it is possible to have the DUMP take 10s of minutes





## How to Limit REAL Storage

- **New Hidden ZPARM SPRMRSMX**
- **Causes a DB2 outage when the limit hits**
- **Delivered in APAR PK18354**
- **Not widely broadcast**
- **Preferable to monitor the REAL storage numbers in IFCID 225 and generate alerts when large increase in AUX or REAL approaches max available**



## DB2 Service Monitor (V9 CM)

- **Automatically issues console messages when DBM1 virtual storage below the 2GB bar reaches critical usage thresholds**
  - 88, 92, 96, or 98 percent of available storage
- **Identifies the agents that consume the most storage**

```
DSNV508I -SE20 DSNVMON - DB2 DBM1 BELOW-THE-BAR
STORAGE NOTIFICATION
          91% CONSUMED
          87% CONSUMED BY DB2

DSNV510I -SE20 DSNVMON - BEGINING DISPLAY OF LARGEST
STORAGE CONSUMERS IN DBM1
DSNV512I -SE20 DSNVMON - AGENT 1: 094
NAME     ST  A  REQ  ID           AUTHID PLAN
-----  --  -  ---  --           -
SERVER RA * 18461 SE2DIA004 R3USER DISTSERV
LONG 1720K VLONG 388K 64BIT 2056K

DSNV512I -SE20 DSNVMON - AGENT 2: 095
NAME     ST  A  REQ  ID           AUTHID PLAN
-----  --  -  ---  --           -
SERVER RA * 9270  SE2DIA001 R3USER  DISTSERV
LONG 1672K VLONG 388K 64BIT 2056K
```



# New CPU AND STORAGE Metrics (V8 and V9)

- **Introduced by PK62116**
  - APAR PK66373 must be applied
  - APAR OA24404 for RMF must be applied
  - ZPARM ZOSMETRICS must be set to YES (default is NO)
  - RMF Monitor Type III must be running
- **Adds the following metrics to IFCID 001 (QWOS)**
  - Number of CPs
  - CPU utilisation
  - Storage values
  - Unreferenced Interval Count (UIC)
    - Indicator for real-storage constraints
- **Support in Omegamon XE for DB2 PM/PE V4.20**

CPU AND STORAGE METRICS	QUANTITY
CP LPAR	4.00
CPU UTILIZATION LPAR	39.20
CPU UTILIZATION DB2	0.00
CPU UTILIZATION DB2 MSTR	0.00
CPU UTILIZATION DB2 DBM1	0.00
UNREFERENCED INTERVAL COUNT	65535.00
REAL STORAGE LPAR (MB)	3071.00
FREE REAL STORAGE LPAR (MB)	1537.10
USED REAL STORAGE DB2 (MB)	194.65
VIRTUAL STORAGE LPAR (MB)	10269.35
FREE VIRTUAL STOR LPAR (MB)	8936.10
USED VIRTUAL STOR DB2 (MB)	194.65