DB2 for z/OS Virtual Storage Management 2005 Revisited

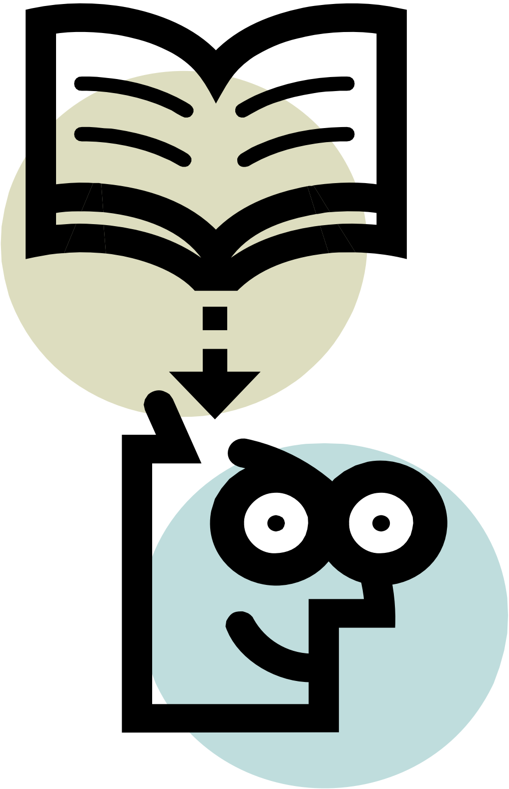
John J. Campbell

Distinguished Engineer

DB2 for z/OS Development

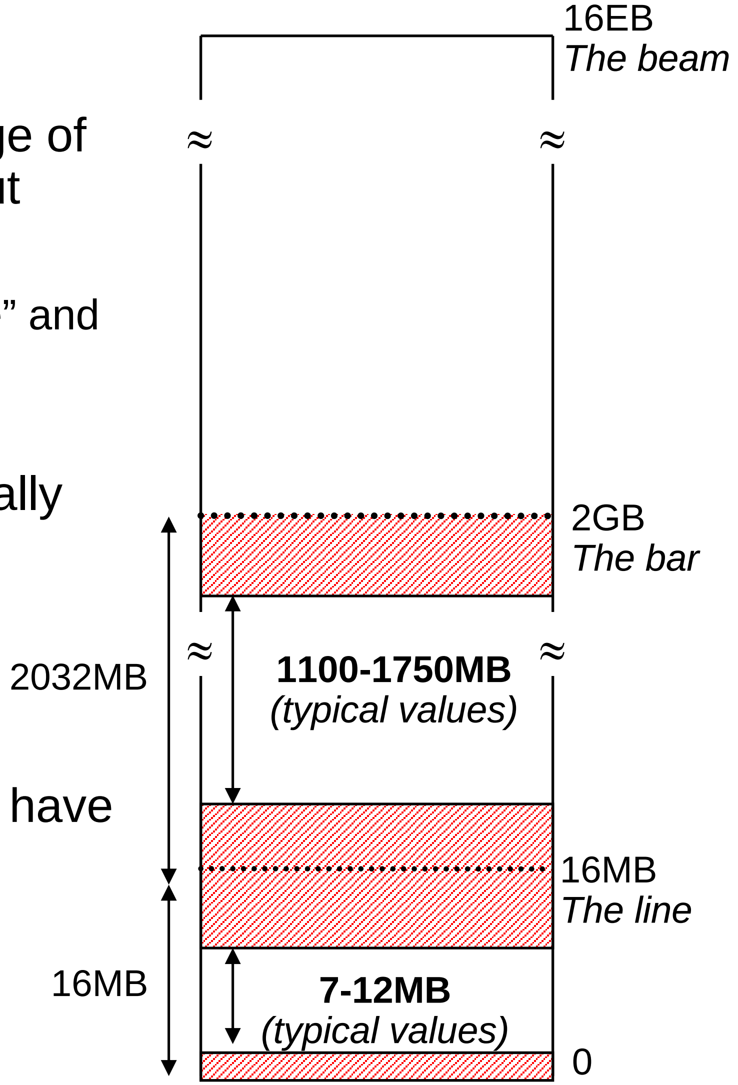
Agenda

- What Is The Problem?
- Analysing Virtual Storage Used
- What Is The Value of V8 64-bit?
- Achieving VSCR
- Protecting The System
- Miscellaneous
- Summary



What Is The Problem?

- Each 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 7-12MB available "below the line"
  - Typical 1100-1750MB available "above the line"
- Storage is allocated into different subpools which have unique characteristics e.g., SP229
  - Storage acquired via GETMAIN
  - Storage released by FREEMAIN



What Is The Problem?

- GETMAIN processing by DB2
  - Requests may be conditional or unconditional
  - "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 when critical task and no toleration of error

What Are The Drivers?

- Workload growth
- Shrinking maximum region size (EPVT) available to DB2
  - Conflict between using ECSA (IMS) and EPVT (CICS)
  - 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
  - Other use of ECSA
    - WebSphere
    - IRLM PC=NO
  - Generous over allocation of ECSA and other extended common areas
  - LPAR consolidation
  - Common LPAR image for Sysplex (best practice)

What Are The Drivers? ...

- Increase (regression) in average thread storage footprint across successive DB2 releases
  - Allied threads
  - DBATs
- Long running persistent threads (IMS WFI, CICS Protected Entry Threads, WebSphere) with
  - Plan with many packages with RELEASE(DEALLOCATE)
  - Plan with many packages with RELEASE(COMMIT) and CONTSTOR=NO
- Use of local (thread) dynamic statement caching (BIND option KEEPDYNAMIC YES and zparm MAXKEEPD > 0)
  - ERP & CRM applications e.g., SAP
  - WebSphere with JDBC
- CTHREAD and MAXDBAT throttles set to high values that cannot be supported when system slowdown occurs, workload keeps arriving and more threads (allied, DBATs) are allocated

What Are The Drivers? ...

- Overuse and misuse of ESA Compression particularly when combined with wide partitioning
- Over allocation of fixed pool storage
  - Primary virtual pools
  - EDM Pool
- DB2 subsystems recycled less frequently and more sensitive to storage leaks
- Query workloads
  - Concurrent sort activity
  - Parallelism with high degree
- Heavy concurrent SQL PREPARE activity with CACHEDYN=NO
- Etc…

What Consumes The Virtual Storage Used?

Virtual Buffer Pool	0 to 1024MB
EDM Pool	20 to 400MB
User Thread Storage	Storage Footprint per Thread (allied, DBAT) ▪ Low end 200 to 400 KB – Static SQL ▪ Mid range 500 KB to 2 M – Static / Dynamic SQL ▪ High end 3 MB to 10 MB or even higher – Dynamic SQL, Heavy Sort, Parallelism ▪ Mileage will vary based on specific installation workload
System Thread Storage	40KB per system thread
Compression Dictionary	64KB per open compressed dataset
Local Dynamic Statement Cache	0 to 300MB
Others	200 to 400MB

Note: Range of typical values not minimum/maximum

Analysing Virtual Storage Used

How much virtual storage is available to and used by DBM1 address space?

- 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))
  - Use to identify potential storage shortages and to get historical view of virtual storage consumption
  - Calculate amount of storage available above the line by subtracting MAX LSQA/SWA/229/230 PAGES ALLOCATED and MAX USER REGION PAGES ALLOCATED from REGION ASSIGNED
  - How much is enough?
    - Greater than 500MB spare is AOK (GREEN)
    - Between 200-500MB spare is boundary condition (AMBER)
    - Less than 200MB action is required (RED)

```
                              V I R T U A L   S T O R A G E   A C T I V I T Y

z/OS V1R5                 SYSTEM ID X9               DATE 05/25/2005            INTERVAL 10.00.68
                          RPT VERSION V1R5 RMF       TIME 15.00.00              CYCLE 0.250 SECONDS

                                       PRIVATE AREA SUMMARY
     JOB NAME -        DBL1DBM1                                      REGION REQUESTED                0K
     STEP NAME -       DBL1DBM1                                      REGION ASSIGNED (BELOW 16M)  11.0M
     PROGRAM NAME -    DSNYASCP                                      REGION ASSIGNED (ABOVE 16M)  1635M
     NUMBER OF SAMPLES -    240
                                        PRIVATE STORAGE MAP
                        BELOW 16M                                      EXTENDED (ABOVE 16M)
                  AFFFFF ________________________                   ________________________ 7FFFFFF
                         | LSQA/SWA              |                  | LSQA/SWA              |
                         | 229/230           328K|    BOTTOM OF     | 229/230          1089M|
                  AAE000 |_____15.00.00__________| ALLOCATED AREA   |_____15.05.32__________| 3BEF700
                         | UNUSED              0K|                  | UNUSED              0K|
                  B00000 |_______________________|  GETMAIN LIMIT   |_______________________| 7FFFFFF
                         | UNUSED           10.4M|                  | UNUSED            521M|
                   3F000 |_____15.00.00__________|     TOP OF       |_____15.00.00__________| 1B58F00
                         | USER                  | ALLOCATED AREA   | USER                  |
                         | REGION            228K|                  | REGION                |
                    6000 |_______________________|                  |                  24.6M|
                         | SYSTEM REGION      16K|                  |                       |
                    2000 ------------------------                   ------------------------ 19D0000
                       ---------- BELOW 16M ----------------  --------------- ABOVE 16M -----------
                         MIN              MAX              AVG    MIN              MAX             AVG
LSQA/SWA/229/230
 FREE PAGES (BYTES)       40K 15.00.00     40K 15.00.00     40K     28K 15.02.37    724K 15.00.00    210K
 LARGEST FREE BLOCK       36K 15.00.00     36K 15.00.00     36K      4K 15.02.37    320K 15.00.00     86K
 PAGES ALLOCATED
  (IN BYTES)             288K 15.00.00    288K 15.00.00    288K   92.8M 15.00.00   1089M 15.05.32    828M
USER REGION
 FREE PAGES (BYTES)     10.4M 15.00.00   10.4M 15.00.00   10.4M    521M 15.05.32   1517M 15.00.00    783M
 LARGEST FREE BLOCK
   IN GETMAIN LIMIT     10.4M 15.00.00   10.4M 15.00.00   10.4M    521M 15.05.32   1517M 15.00.00    783M
 PAGES ALLOCATED
   (IN BYTES)            228K 15.00.00    228K 15.00.00    228K   24.6M 15.00.00   24.6M 15.00.00   24.6M
```

1635 – (1089 + 24.6) = 521.4MB available

Analysing Virtual Storage Used …

What consumes the virtual storage used by DBM1 address space?

- DB2 instrumentation for detail
  - IFCID 225
    - Summary Information – snapshot as each DB2 Statistics interval becomes due
    - Available through DB2 Statistics Trace Class 6
    - Tiny overhead in terms of increased CPU resource consumption and SMF data volume
    - Start automatically via zparm SMFSTAT(...,6)
    - Recommend zparms STATIME=5 and SYNCVAL=0
    - Will be available in the default Statistics Trace Class 1 with V7 PQ99658
  - IFCID 217
    - Detail Information at thread level
    - Available through Global Trace Class 10
  - For description of new IFCIDs
    - DSN710.SDSNSAMP(DSNWMSGS)
    - DSN810.SDSNIVPD(DSNWMSGS)

Analysing Virtual Storage Used …

What consumes the virtual storage used by DBM1 address space? ...

- First class support provided by DB2PE/PM
  - 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
    - IFCID 217 and 225 supported independent of DB2 release which created records

- REXX Tools (MEMU2, MEMUSAGE) available from JC

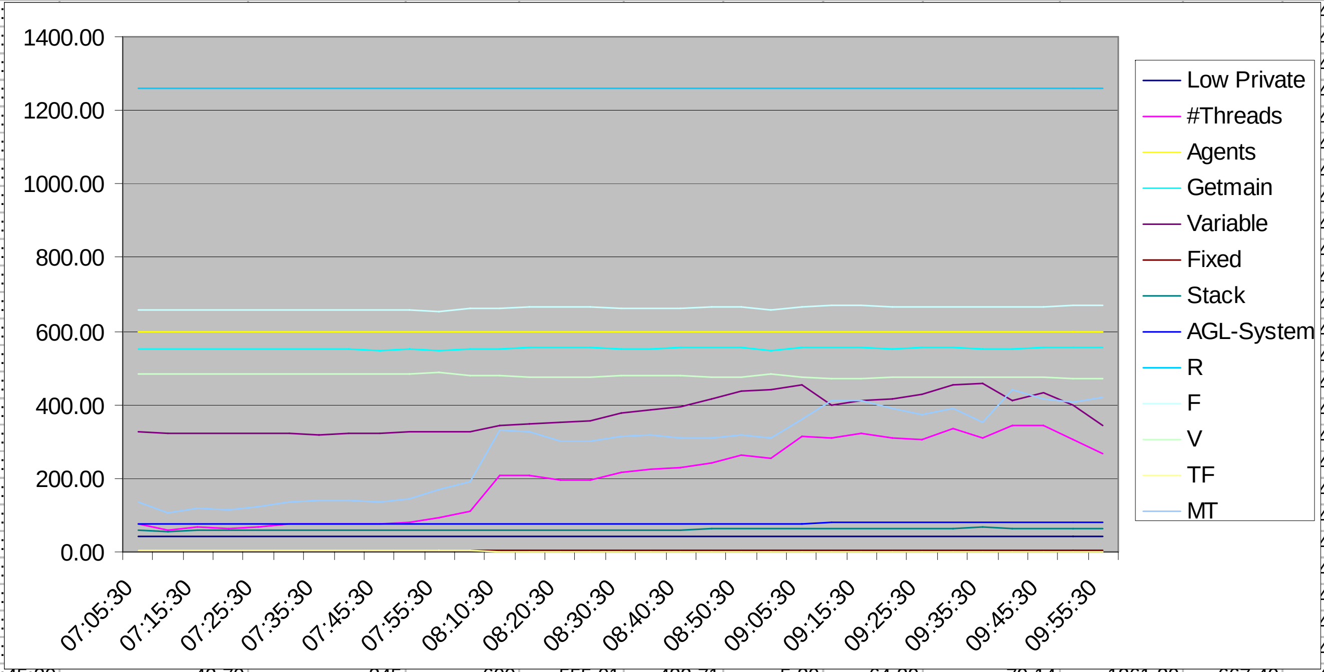
DB2PE Sample – Statistics Trace | Report

```
DBM1 AND MVS STORAGE BELOW 2 GB                     QUANTITY
--------------------------------------------  ------------------
 TOTAL DBM1 STORAGE BELOW 2 GB            (MB)            773.05
  TOTAL GETMAINED STORAGE                 (MB)            575.00
    VIRTUAL BUFFER POOLS                  (MB)            429.69
    VIRTUAL POOL CONTROL BLOCKS           (MB)             13.43
    EDM POOL                              (MB)            117.19
    COMPRESSION DICTIONARY                (MB)              2.35
    CASTOUT BUFFERS                       (MB)              9.13
    DATA SPACE LOOKASIDE BUFFER           (MB)              0.00
    HIPERPOOL CONTROL BLOCKS              (MB)              0.05
    DATA SPACE BP CONTROL BLOCKS          (MB)              0.00
  TOTAL VARIABLE STORAGE                  (MB)            139.53
   TOTAL AGENT LOCAL STORAGE              (MB)             53.94
    TOTAL AGENT SYSTEM STORAGE            (MB)             32.35
      NUMBER OF PREFETCH ENGINES                           77.00
      NUMBER OF DEFERRED WRITE ENGINES                    300.00
      NUMBER OF CASTOUT ENGINES                            73.00
      NUMBER OF GBP WRITE ENGINES                          58.00
      NUMBER OF P-LOCK/NOTIFY EXIT ENGINES                  9.00
    TOTAL AGENT NON-SYSTEM STORAGE        (MB)             21.60
      TOTAL NUMBER OF ACTIVE USER THREADS                  29.67
   RDS OP POOL                            (MB)             34.54
   RID POOL                               (MB)             16.97
   PIPE MANAGER SUB POOL                  (MB)              0.00
   LOCAL DYNAMIC STMT CACHE CNTL BLKS     (MB)              0.99
   THREAD COPIES OF CACHED SQL STMTS      (MB)              0.00
     IN USE STORAGE                       (MB)               N/A
     STATEMENTS COUNT                                        N/A
     HWM FOR ALLOCATED STATEMENTS         (MB)               N/A
       STATEMENT COUNT AT HWM                                N/A
       DATE AT HWM                                           N/A
       TIME AT HWM                                           N/A
   BUFFER & DATA MANAGER TRACE TBL        (MB)              9.41
  TOTAL FIXED STORAGE                     (MB)              3.80
  TOTAL GETMAINED STACK STORAGE           (MB)             54.71
 STORAGE CUSHION                          (MB)            112.04
```

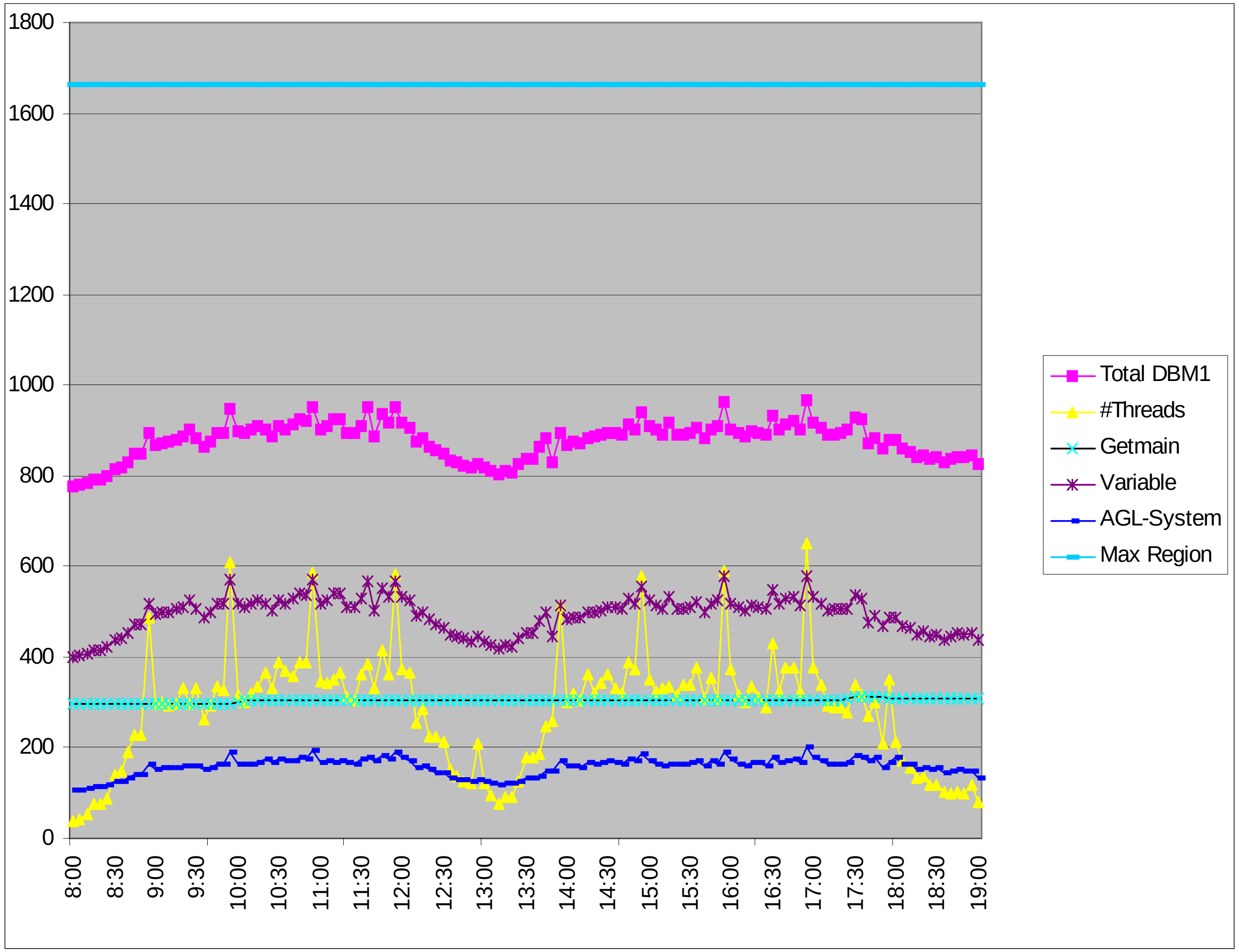
```
DBM1 AND MVS STORAGE BELOW 2 GB                     QUANTITY
--------------------------------------------  ------------------
24 BIT LOW PRIVATE                        (MB)              0.23
24 BIT HIGH PRIVATE                       (MB)              2.25
31 BIT EXTENDED LOW PRIVATE               (MB)             27.38
31 BIT EXTENDED HIGH PRIVATE              (MB)            954.23
EXTENDED REGION SIZE (MAX)                (MB)           1714.00
EXTENDED CSA SIZE                         (MB)            200.06

AVERAGE THREAD FOOTPRINT                  (MB)              3.61
MAX NUMBER OF POSSIBLE THREADS                            236.12
```

Study Historical Evolutionary Trend

Basic Cushion ©	163.00								Theoritical	Fixed	Upper		
EPrivate	1304.00	Allied Threads							Max Region	Areas	Limit	Thread	Max
	31Bit Extended	+ DBATs	# System	Total	Total	Total	Total		Size		Variable	Footprint	Threads
Time	Low Private	#Threads	Agents	Getmain	Variable	Fixed	Stack	AGL-System	R	F	V	TF	MT
07:05:30	42.70	78	600	551.70	326.34	5.07	59.69	78.16	1261.30	659.16	481.84	3.51	137
07:10:30	42.70	60	600	551.08	321.43	5.07	56.69	78.16	1261.30	655.54	485.46	4.50	108
07:15:30	42.70	66	600	549.52	322.02	5.07	59.79	78.24	1261.30	657.08	483.92	4.06	119
07:20:30	42.70	63	600	550.09	320.81	5.07	59.66	78.16	1261.30	657.52	483.48	4.24	114
07:25:30	42.70	67	600	550.97	320.89	5.07	59.68	78.16	1261.30	658.42	482.58	3.99	121
07:											483.41	3.52	137
07:											483.78	3.42	141
07:											483.57	3.49	139
07:											485.07	3.57	136
07:											484.08	3.37	144
07:											486.03	2.89	168
08:											480.53	2.52	191
08:											478.81	1.45	331
08:											476.76	1.46	326
08:											475.46	1.57	303
08:											476.51	1.58	301
08:											479.06	1.52	315
08:											478.66	1.51	317
08:											477.91	1.54	310
08:											476.75	1.55	308
08:											476.94	1.51	316
08:											481.59	1.56	308
09:											474.31	1.32	360
09:											472.55	1.14	413
09:											472.04	1.14	413
09:											474.56	1.21	392
09:											473.98	1.27	375
09:											473.81	1.22	388
09:											473.86	1.34	354
09:											476.19	1.08	439
09:45:30	42.70	345	600	555.01	432.71	5.30	64.39	79.14	1261.30	667.40	473.60	1.14	417
09:50:30	42.70	306	600	557.81	397.20	5.31	64.14	79.23	1261.30	669.96	471.04	1.16	406
09:55:30	42.70	266	600	557.37	344.21	5.33	64.00	79.14	1261.30	669.40	471.60	1.13	418



Study Historical Evolutionary Trend …



Analysing Virtual Storage Used …

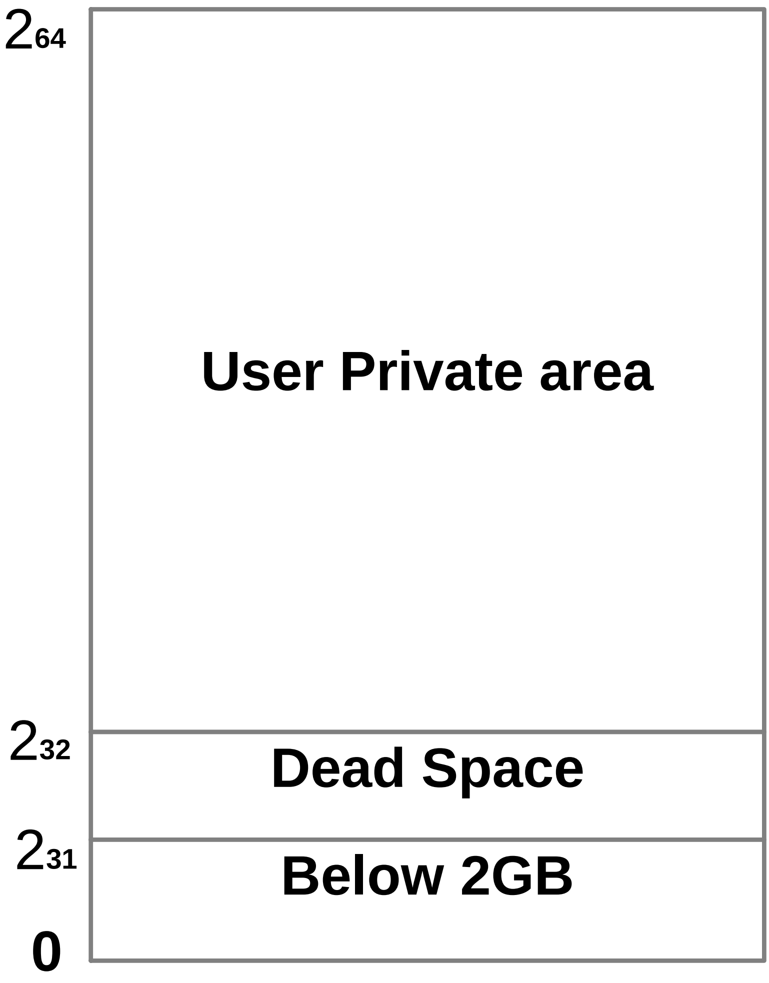
- DB2 Dump Formatter
  - To understand current storage consumption
  - See 'DB2 Diagnosis Guide & Reference' for use of DSNWDMP command
  - Storage types
    - POOL (PHB):
      - Fixed
      - Variable
    - Getmained (GMB)
    - Stack
  - Use SM=1 to get Summary (PHB level + GMBs)
    - Should always be run first
    - Gives the totals and good place to start
  - Use IPCS command VERBX VSMDATA
    - Obtain size of Subpool 229 Key 7
    - Compare against amount consumed by DB2

Analysing Virtual Storage Used …

- DB2 Dump Formatter …

```
===Stack(SKB) total:               96152K
===Fixed subpool total:             1868K
   GMB total:                     695626K
   ADMF AGL system total:          64484K
   ADMF AGL non system total:      15020K
   ADMF local total:               79504K
   Variable subpool total:        320836K
   Var+Fix+GMB subpool total:     1018330K
```

What Is The Value of V8 64-bit?



<- 16EB "beam"(?)

- Buffer pools, BM control blocks
- Castout buffers
- RID pool
- EDM DBD cache (OBDs)
- Global dynamic stmt cache
- Sort pool
- Trace tables (Global, Lock, BB)
- Accounting blocks
- Compression dictionaries
- IRLM locks

<- 2GB "bar"

- Most thread storage. Local DSC
- Other EDM and RID component

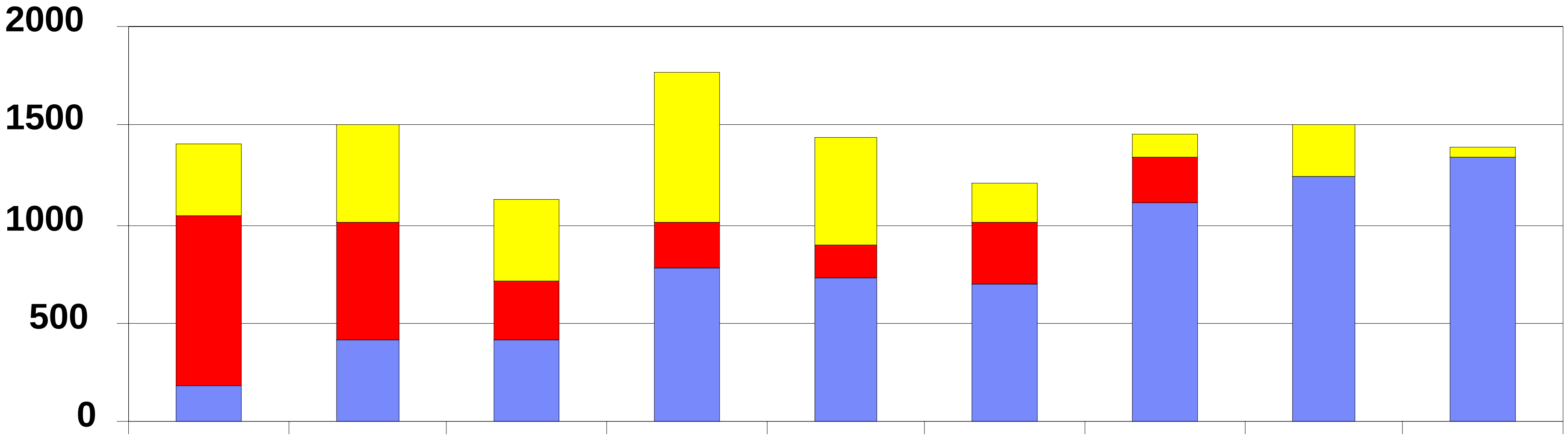
How Much Will You Gain From 64-bit Virtual?

- At first sight, re-engineering of DBM1 to exploit 64-bit virtual should make a very significant difference in terms of providing significant VSCR
- Many customers will accrue significant VSCR, but not true in all cases
- Consider V7 installation achieving significant VSCR through VSTOR diet
    - Maximum use of Dataspace Bufferpools
    - Using dataspace extension to EDM Pool for Global Dynamic Statement Cache
    - Significant cut back in use of ESA Compression
  - and
    - Use of local (thread) dynamic statement caching
      - BIND option KEEPDYNAMIC YES and zparm MAXKEEPD>0
    - Running many threads (allied, DBAT)
  - What is the 'net' benefit in V8
    - Dataspace Lookaside Pool is eliminated
    - Buffer Manager control blocks going above 2GB bar
    - Some other items going above the 2GB bar
      - Compression dictionaries, certain EDM/RID Pool components, Thread Sort Pool…

64-bit and Thread Storage

- Most of the thread storage stays below the 2GB bar
  - Agent Local, Stack Storage, Local Dynamic Statement Cache
  - Expect some regression
- Average estimate of DBM1 VSTOR increase (regression) below "the bar" in V8 versus V7 – preliminary and subject to change as more customer data is obtained
  - Thread storage: +30 to 40% for static or system, +50 to 100% for dynamic
  - Stack storage: +100%
  - Dynamic statement cache: +10 to 100%
  - EDM pool: roughly the same
  - RID pool: -90%
  - Others: -100%
- Bigger thread and stack storage in V8 for
  - Long names, keys, statements, other new functions, reduce CPU regression
  - A portion of RDS Op pool for dynamic SQL
- How much more VSCR, if any, in DBM1 address space below "the bar" depends on % of storage used for threads, stacks, and local DSC versus others

DBM1 below 2GB available, used by V7, used by V8 High-end peak period data from



Customer Example: Value of V8 relative to V7

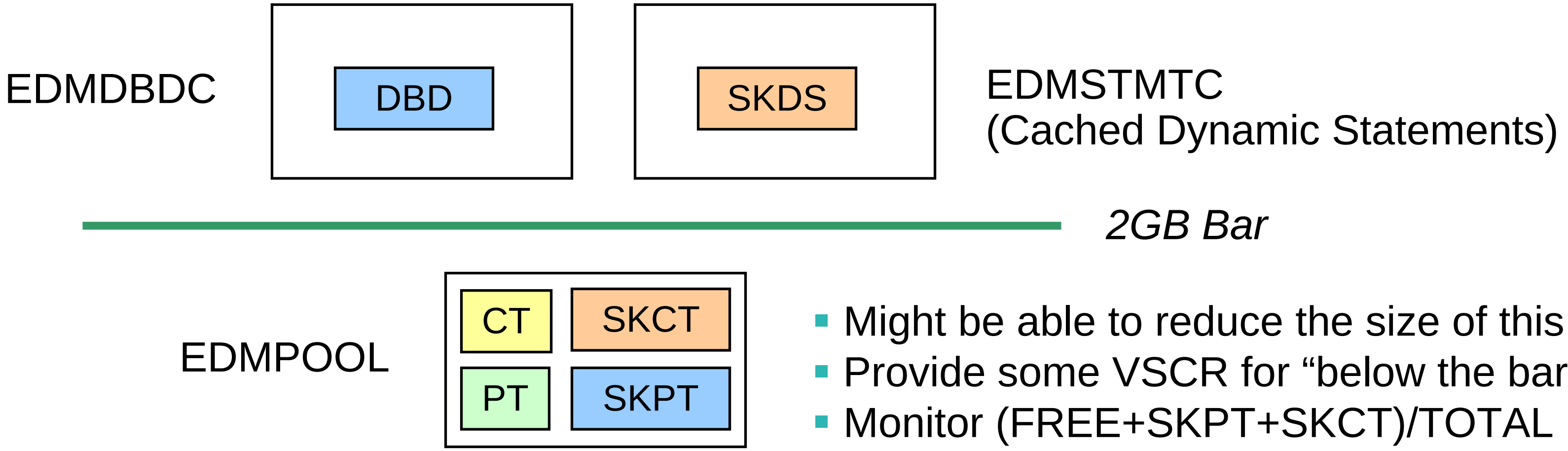
	V7 measured (MB)	V8 estimated (MB)		Notes:
Virtual buffer pool	15	0		0
Buffer pool control blocks	95	0		0
Dataspace lookaside buffer	48	0		0
EDM pool	88	88		Roughly the same
Compression dictionary	54	0		0
Castout buffers	28	0		0
System thread storage	89	134		+50%
User thread storage	114	217		+30 to 40% static or system, +50 to 100% dynamic
RDS OP pool	8	0		0
RID pool	78	8		-90%
Pipe Manager subpool	1	1		Same
DSC control blocks	70	0		0 with V8 APAR PQ96772
Local DSC	8	12		+10 to 100%
BM/DM trace table	18	9		-50%
Fixed storage	3	3		Same
Stack storage	58	116		+100%
Total	775	588		
% VSCR		24		

64-bit Virtual Key Messages

- Will not absolutely eliminate VSTOR constraint below “the bar” in DBM1 AS
- Can provide valuable VSCR for many installations, but not all installations
- Mileage will vary by installation
- Will be able to exploit all available processor storage on latest processor models (currently 256GB, current DB2 limit of 1TB)
  - XXL bufferpools: good real memory vs IO reduction trade off
  - Expand use of ESA Compression
  - Possibly support an additional number of active threads (allied, DBAT)
  - Larger thread Sort Pool
  - May be able to set zparms CONTSTOR=NO and MINSTOR=NO
- Must have sufficient real storage to fully back any increased usage
  - Performance opportunity if you do
  - Danger if you do not
- Installations must continue to plan for, monitor and tune VSTOR usage below “the bar”
- Get current service for storage, monitor Info APAR II10817

64-bit and EDM Pool

- V7 option to have EDM data space for Global Dynamic Statement Cache
- No such option in V8
- EDM Pool split into three specific pools
  - Use SET SYSPARM LOAD to increase/decrease these pools via zparm changes as needed: EDMPOOL, EDMDBDC, EDMSTMTC



- Might be able to reduce the size of this EDM pool
- Provide some VSCR for “below the bar” storage
- Monitor (FREE+SKPT+SKCT)/TOTAL pages

Real Storage Use

- Increase in real storage usage with V8
  - Up to 20%
  - Many installations monitor virtual storage usage
  - Few installations monitor real storage usage, which is a concern
- Important subsystems such as DB2 should not be paging to auxiliary storage
- Strong recommendation
  - Keep DASD paging rates very low (near zero)
  - Monitor via RMF Mon III

Achieving Virtual Storage Constraint Relief

- Goals
  - Create “head room” for some tuning and for contingency
  - Establish sensible profile for projecting future requirement based on anticipated workload growth
- Establish
  - Performance baseline
  - “Stretch” target
- Trading CPU cycles for VSCR
  - Proceed in incremental manner
- Monitor against history and study evolutionary trend over time
  - Storage utilisation
  - CPU and IO performance
- Develop Virtual Storage Budget (spreadsheet)
  - Record data
  - Better understanding of “big hitters” in terms of VSTOR consumption
  - Anticipate the effect of tuning changes (What If?)

Sample Virtual Storage Budget for DBM1 AS

Virtual Pools	646.64
Local DSC	345.93
User Threads	330.77
EDM Pool	100.00
Stack Storage	80.59
ECSA	68.80
RID Pool	60.00
EPLPA	46.50
ESQA	43.60
Hiperpool Control Blocks	35.78
Storage Cushion	25.79
System Threads	23.95
ENUC	22.10
RDS OP Pool	20.31
Virtual Pool Control Blocks	19.03
Internal Trace Pool	13.99
DB2 Code	11.32
Open Data Sets	5.91
Compression Dictionaries	1.88

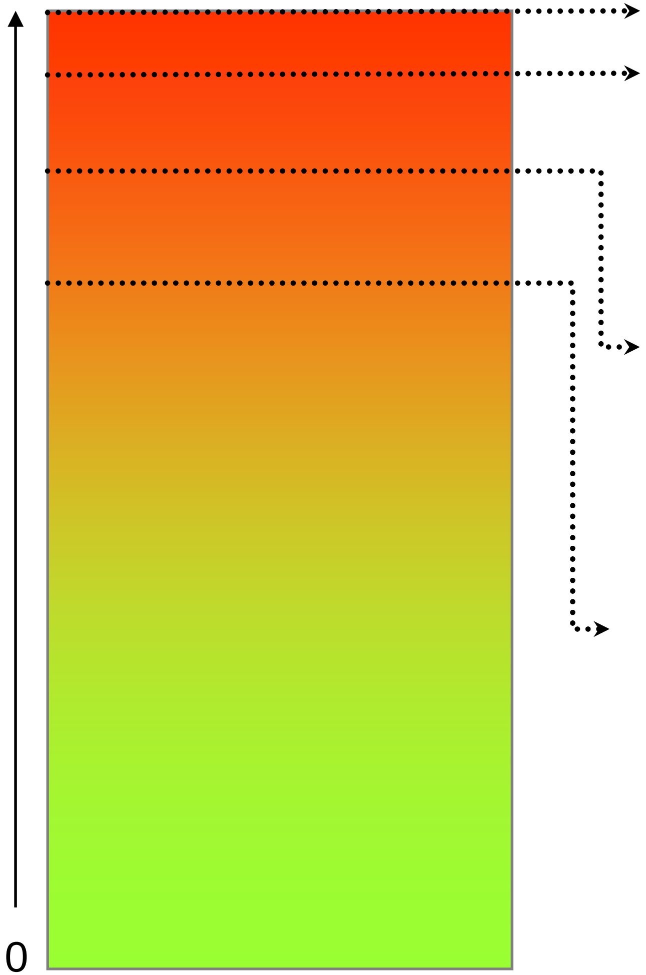
Achieving Virtual Storage Constraint Relief …

- Primary Options
  - Migrate to V8 (see earlier discussion)
  - Storage management improvements (V5->V8)
  - Exploit Dataspace Bufferpools
  - Exploit Dataspace extension to EDM Pool
  - Reduce size of Local Dynamic Statement Cache (reduce MAXKEEPD)
  - Reduce number of compressed datasets and/or number of partitions
  - Reduce use of RELEASE(DEALLOCATE)
  - Reduce size of Extended Common Areas (ECSA, EPLPA, etc)
  - Reduce number of long running persistent threads
  - Exploit Type 2 Inactive Connections for DDF work
  - Switch SMF INTERVAL recording for STCs to NODETAIL
  - Switch IRLM PC=YES
  - Implement Data Sharing or Increase width of existing Data Sharing Group

Full System Storage Contraction

- Full system storage contraction is driven by
  - *PVTCRIT* = MVS Cushion = Storage for Must Complete
    - Fixed, real value
    - Based on CTHREAD/MAXDBAT
    - (CTHREAD+MAXDBAT+1)*20K
  - *PVTMVS* = MVS Available = Storage for MVS
    - Amount set aside (reserved) for dataset opens
    - Based on DSMAX
    - Virtual number and no guarantee
    - (DSMAX*1300)+40K
  - *PVTSOS* = MVS Warning To Contract = Cushion Warning
    - Max(5% of Extended Region Size, PVTMVS)

Full System Storage Contraction …



Extended Region Size (1650M)

Storage Critical (-12.8M)

- Less than *PVTCRIT+PVTMVS* remaining
- Thread abends start to occur

Storage Cushion (-95.3M)

- Less than *PVTSOS+PVTCRIT+PVTMVS* remaining
- Contraction starts to occur
  - See DBM1 TCB Time for CPU overhead

Basic Storage Cushion (-200M)

Note: Sample values based on customer example

Thread Storage Contraction

- Turned on by zparm CONTSTOR = YES with associated CPU overhead
- Benefit should be carefully evaluated before enabling
- Compresses out part of Agent Local Non-System storage
- Does not compress
  - Agent Local System, RDS Op Pool (no longer exists in V8)
  - Getmained Stack Storage, Local Dynamic Statement Cache
- Controlled by two hidden zparms
  - SPRMSTH @ 2097152 (V7), 1048576 (V8)
  - SPRMCTH @ 50 (V7), 10 (V8)
- V7 triggers:
  - No. of Commits > SPRMCTH, or
  - Agent Local Non-System > SPRMSTH and No. of Commits > 5
- V8 trigger:
  - No. of Commits > SPRMCTH or Agent Local Non-System > SPRMSTH

Dataspace Bufferpools

- For each bufferpool, mutually exclusive with Primary Virtual Pool w/-Hiperpool
- Primary Virtual Pool moved out of DBM1 AS
- Relatively small Dataspace Lookaside Pool allocated in DBM1 AS
- Designed to be 100% backed by real (central) storage
- Delivers excellent performance when 100% backed by real storage
- Equal or better performance relative to Primary Virtual Pool with Hiperpool
- ROT: 1-10% additional CPU relative to Primary Virtual Pool without Hiperpool
- Significant performance penalty if insufficient real storage
- Not practical proposition with ESA (31-bit) LPAR with max. of only 2GB of central (real) storage
- Need ESAME (64-bit) LPAR with large processor storage (ARCHLVL=2)

Other VSCR Items …

- Use Dataspace extension to EDM Pool for Global Dynamic Statement Cache
- Reduce size of Local Dynamic Statement Cache
  - Trade CPU for VSCR, or
  - Increase size of Global Dynamic Statement Cache to compensate for lost CPU performance
  - Example
    - MAXKEEPD: 16000->8000 (incremental units of 2000)
    - EDMPOOL: 80MB->110MB (to compensate)

Other VSCR Items …

- Reduce number of compressed datasets that are open
  - One dictionary per pageset/partition (dataset)
  - Each dictionary can be up to 64KB
  - Concerned about aggregate demand for open compressed datasets
  - Determine budget and ruthlessly prioritise use based on benefit
  - ALTER TABLESPACE COMPRESS NO plus REORG
  - Reduce number of partitions per tablespace (disruptive)

Other VSCR Items …

- Use RELEASE DEALLOCATE selectively based on benefit
  - Overuse with persistent threads can create a VSTOR issue
    - Accumulating ever more storage for statements that are not being used
    - Storage for unused statements can be left around until deallocation
    - Also drive up demand for EDM Pool resources
    - Ineffective thread and full system storage contraction
  - Best reserved for
    - High volume and/or performance sensitive OLTP plans/packages
    - Long running batch programs that take frequent intermediate commits

Other VSCR Items …

- Reduce appetite for ECSA and give back to get larger EPVT
  - IMS users should exploit VSCR features of IMS V7 and later releases
  - Avoid excessive over allocation
- Reduce number of long running persistent DB2 threads
  - If over configured to meet throughput requirement
    - Reduce the number of such threads
  - Ruthlessly cut back on number of JDBC/SQLJ Data Sources
    - Collapse out redundant Data Sources
- Exploit Type 2 Inactive Connections for DDF work
  - Do not use KEEPDYNAMIC(YES)
  - Close open held cursors ahead of commit
  - etc

Other VSCR Items …

- Switch SMF INTERVAL recording for STCs to NODETAIL
  - Problem
    - See Information APAR II07124
    - Symptom SP230 Key storage increasing x MB per day
    - Caused by the amount of SMF Record Type 30 Subtype 4 and Subtype 5 data filling SP230
  - Solution
    - Change from DETAIL to NODETAIL for STC in SMFPRMxx
    - May have to rewrite accounting programs if they are using Subtype 4 and Subtype 5
- Switch IRLM from PC=NO to PC=YES to reduce ECSA requirement
- Implement Data Sharing or Increase width of existing Data Sharing Group
  - Redistribute and spread user workload over multiple members
  - Fewer active threads (allied, DBATs) per member

Protecting The System

- Update Virtual Storage Budget after tuning and monitoring cycles
- Plan to keep 200MB spare
  - Avoid hitting short on storage and driving Full System Storage Contraction
  - Provide some headroom for:
    - Tuning, some growth, Fast Log Apply, abnormal operating conditions
    - Estimate Maximum Number of Threads that can be supported
      - Allied
      - DBAT
- Set zparms CTHREAD and MAXDBAT to protect the system
  - Theoretical maximum: CTHREAD+MAXDBAT = 2000
  - Practical maximum much less (typical range 300-850)
  - Avoid over commiting resources
  - Better to keep the system alive
  - Deny service and queue work outside the system

Estimating Maximum Number of Threads

- Principles

Working Max = Extended Region Size - 31bit Extended Low Private - 200M (Basic Cushion)

Upper Limit Variable = Working Max - Fixed Areas

No. of Active Threads/DBATs = Upper Limit Variable divided by Usage Per Thread

- Basic Formula for estimating Number of Active Threads/DBATs

Fixed Areas = Total Getmained Storage below the 2GB bar
+ Total Getmained Stack Storage + Total Fixed Storage

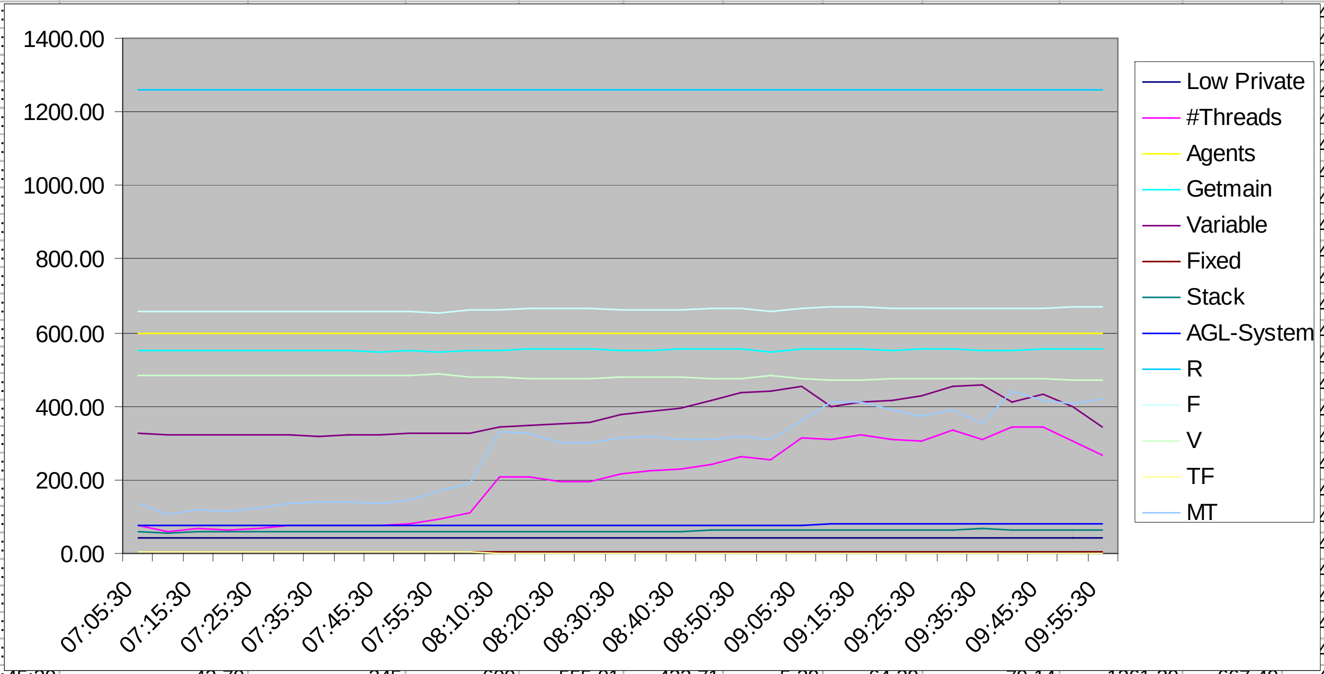
Upper Limit Variable = Working Max - Fixed Areas

Usage Per Thread = (Total Variable Storage - Total Agent System Storage)
divided by (Allied Threads + HWM Active DBATs)

No. of Active Threads = Upper Limit Variable divided by Usage Per Threads

Estimating Maximum Number of Threads …

Basic Cushion ©	163.00								Theoritical	Fixed	Upper		
EPrivate	1304.00	Allied Threads							Max Region	Areas	Limit	Thread	Max
	31Bit Extended	+ DBATs	# System	Total	Total	Total	Total		Size		Variable	Footprint	Threads
Time	Low Private	#Threads	Agents	Getmain	Variable	Fixed	Stack	AGL-System	R	F	V	TF	MT
07:05:30	42.70	78	600	551.70	326.34	5.07	59.69	78.16	1261.30	659.16	481.84	3.51	137
07:10:30	42.70	60	600	551.08	321.43	5.07	56.69	78.16	1261.30	655.54	485.46	4.50	108
07:15:30	42.70	66	600	549.52	322.02	5.07	59.79	78.24	1261.30	657.08	483.92	4.06	119
07:20:30	42.70	63	600	550.09	320.81	5.07	59.66	78.16	1261.30	657.52	483.48	4.24	114
07:25:30	42.70	67	600	550.97	320.89	5.07	59.68	78.16	1261.30	658.42	482.58	3.99	121
07:											483.41	3.52	137
07:											483.78	3.42	141
07:											483.57	3.49	139
07:											485.07	3.57	136
07:											484.08	3.37	144
07:											486.03	2.89	168
08:											480.53	2.52	191
08:											478.81	1.45	331
08:											476.76	1.46	326
08:											475.46	1.57	303
08:											476.51	1.58	301
08:											479.06	1.52	315
08:											478.66	1.51	317
08:											477.91	1.54	310
08:											476.75	1.55	308
08:											476.94	1.51	316
08:											481.59	1.56	308
09:											474.31	1.32	360
09:											472.55	1.14	413
09:											472.04	1.14	413
09:											474.56	1.21	392
09:											473.98	1.27	375
09:											473.81	1.22	388
09:											473.86	1.34	354
09:											476.19	1.08	439
09:45:30	42.70	345	600	555.01	432.71	5.30	64.39	79.14	1261.30	667.40	473.60	1.14	417
09:50:30	42.70	306	600	557.81	397.20	5.31	64.14	79.23	1261.30	669.96	471.04	1.16	406
09:55:30	42.70	266	600	557.37	344.21	5.33	64.00	79.14	1261.30	669.40	471.60	1.13	418



Miscellaneous

- REGION size on DBM1
  - Way of limiting low private storage
  - DB2 allocates SP229 and does not allocate low private storage
  - Installation can pretty much set it to whatever they like
  - It will be ignored by DB2
  - Most installations set REGION to 0K or 0M which means unlimited
  - Caveat
    - Another product may use low private in DBM1 address space
    - Could be affected by the setting of the REGION parameter
    - If so, setting REGION to 64M should be enough for DB2 code …

Miscellaneous …

- “Total No. of Active threads” reported in DB2PE
  - QW0225AT (Allied) plus
  - QDSTHWAT (HWM Active DBATs)
- Monitor DB2 Storage Information APAR on a weekly basis and apply as preventative service
  - II10817

Summary

- Raise awareness of potential issues related to virtual storage consumption in DBM1
- V8 64-bit Support does not eliminate the problem
- Understand how DB2 allocates virtual storage in DBM1 address space
- Encourage proactive behaviour with regular monitoring of virtual storage usage
- Strongly recommend regular monitoring of DB2 storage Info APAR and stay current on service related to storage
- Provide advice on how to build and use Virtual Storage Budget
- Provide practical advice on some tuning options to achieve VSCR
- Set realistic values for zparm throttles based on Virtual Storage Budget to avoid over commitment, queue work outside, deny service and protect system availability