DB2 Information Management Software

DB2 Virtual Storage Exposed

…from VIRTUALly constrained to REALly overcommitted

Adrian Burke
DB2 SWAT team SVL
agburke@us.ibm.com

Certified for IBM Information Management software

Agenda

- Storage terms and concepts
- Address spaces and DB2
- REAL and AUX impact on DB2
- LFAREA
- FlashExpress

**Not everyone has the same amount of room →*

Virtual storage concepts

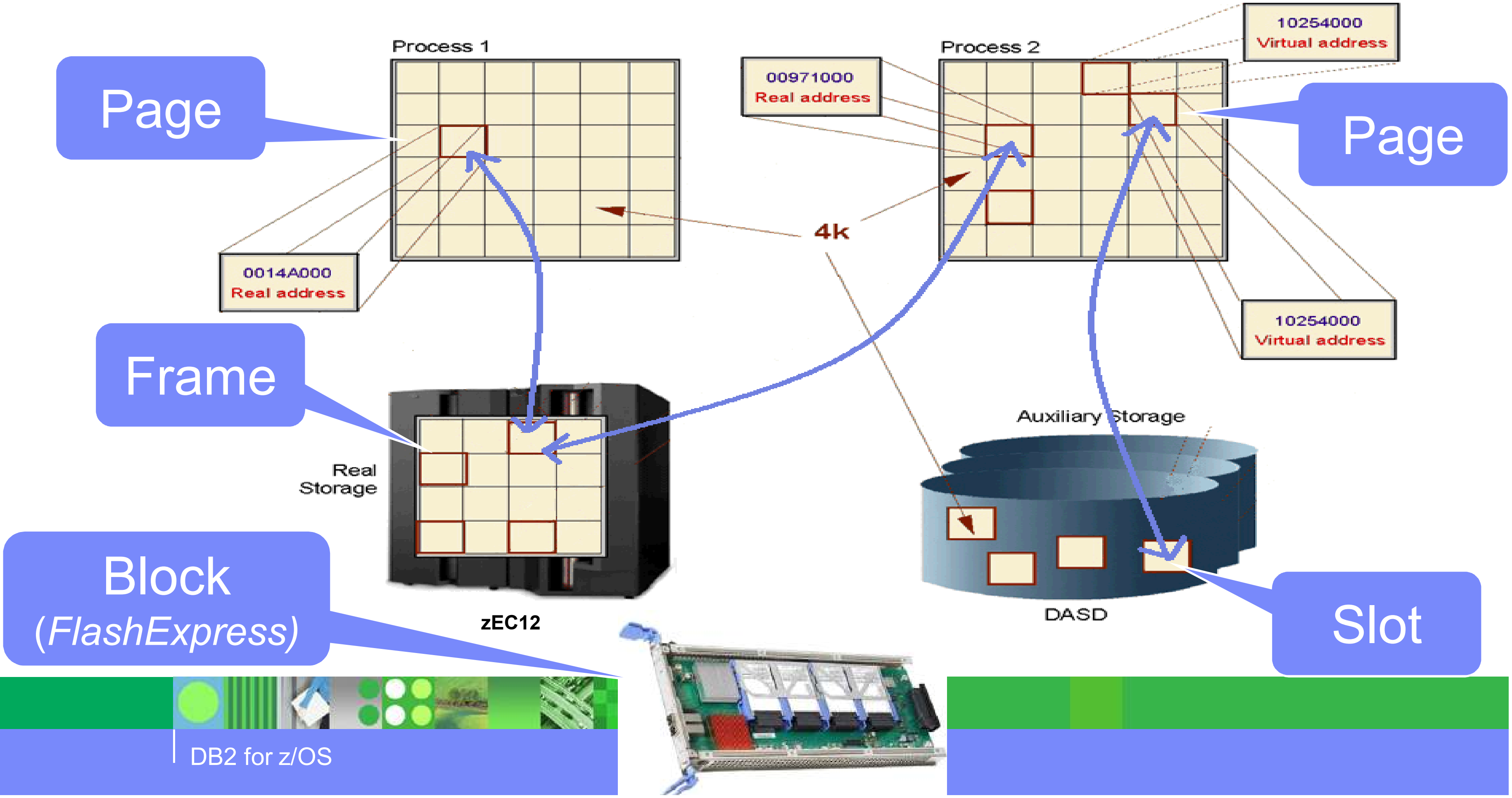
- *Virtual storage* is an “illusion” created through z/OS management of real storage and auxiliary storage through mapping tables
  - It allows each program access to more storage than exists on the box
  - REAL is what you paid for on the CEC
  - AUX is how many page data sets your DASD folks set aside to be used
- The executing portions of a program are kept in *real storage*; the rest is kept in *auxiliary storage*
- Range of addressable virtual storage available to a user or program or the operating system is an *address space*
- Each user or separately running program is represented by an *address space* (each user gets a limited amount of private storage)

How virtual storage works

- Virtual storage is divided into 2GB – 8GB *regions* composed of 1-megabyte *segments* composed of 4-kilobyte *pages*
- Transfer of pages between auxiliary storage and real storage is called *paging*
- When a requested page is not in real storage, an interruption (called a *page fault*) is signaled and the system brings it into real storage
- z/OS uses region, segment and *page tables* to keep track of pages
- Addresses are translated dynamically, a process called *Dynamic Address Translation* (DAT)
- *Frames* and *slots* are repositories for a page of information
  - A frame is a 4K piece of real storage (Central storage)
  - A slot is a 4K record in a page data set (AUX)
  - Page is a 4k,8k,16k,1MB grouping representing frames (Virtual Storage)

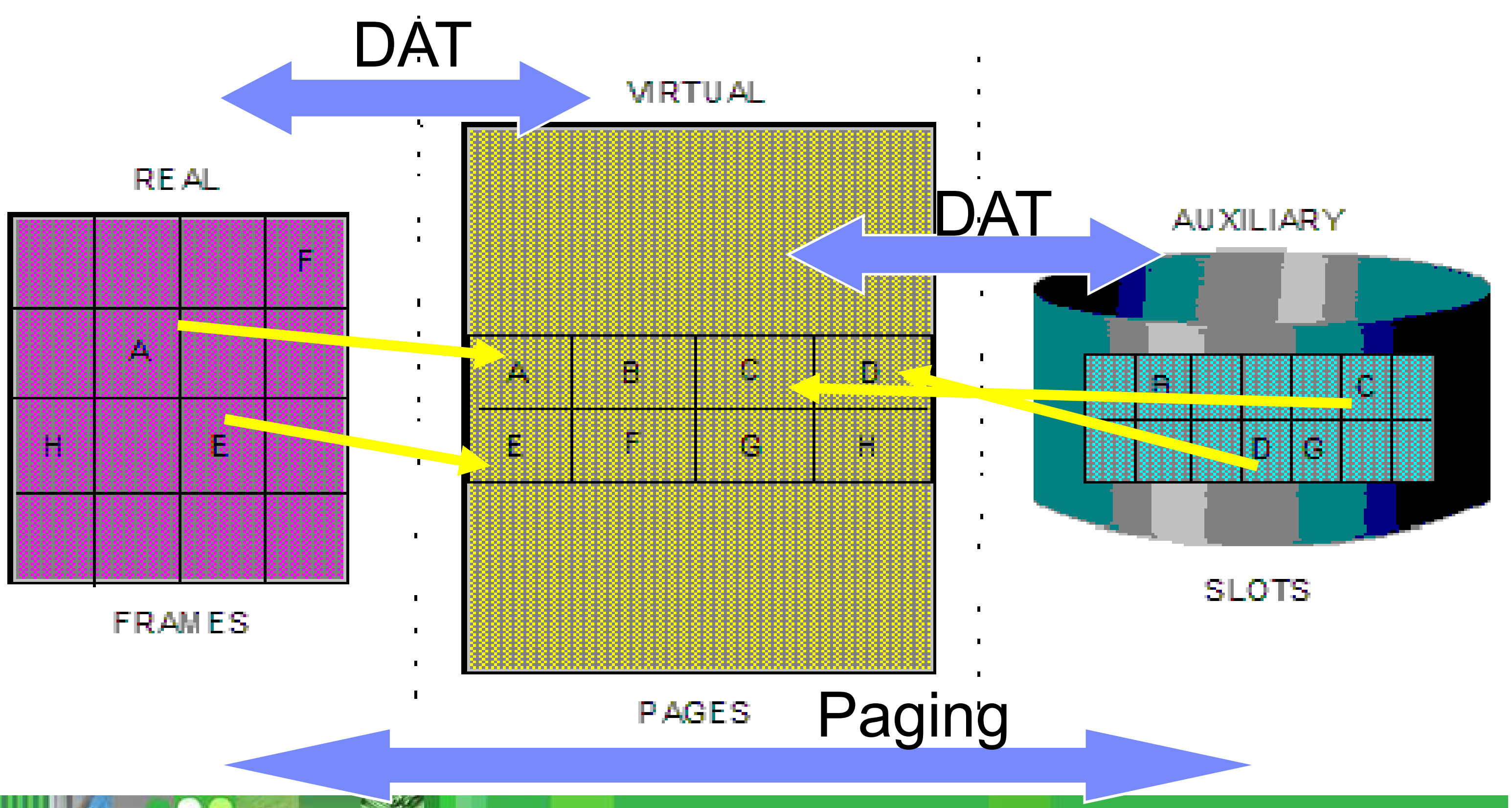
How virtual storage works (continued…)

- We can only read/write to virtual page or AUX slot if it is in REAL memory, so we need to find it or page it in
  - You can have more than 1 virtual address that maps back to a 1 frame of real storage – hence DBM1 and DIST have 64-bit shared in V9 (slide 15)
    - Virtual address is not the same as real address



Pages, Frames, and Slots (continued)

- Dynamic Address translation (DAT)
  - How we determine where the block of storage we need is, could be in virtual or in AUX (DASD)
  - If it is out in AUX then we need a REAL 4k frame fixed in real first (page fixing buffers)
  - Then we need to page that 4k AUX slot into real memory so we can read/write it



Page Stealing

- z/OS tries to keep an adequate supply of available real storage frames on hand
- When this supply becomes low, z/OS uses *page stealing* to replenish it (LRU mechanism, unreferenced interval count)
- Pages that have not been accessed for a relatively long time are good candidates for page stealing.
- z/OS also uses various storage managers to keep track of all pages, frames, and slots in the system:
  - Auxiliary Storage Manager (ASM)
  - Real Storage Manager (RSM)
  - Virtual Storage Manager (VSM)
- If you run out of AUX storage (page data sets full) z/OS will come down

Available Frame Queue Processing

- RSM manages the queue and starts the stealing process when the number of frames falls below the threshold
- There is a ‘low’ level where stealing begins, and ‘ok’ value above which stealing stops
  - **MCCAFCTH=(*lowvalue*,*okvalue*) defaults in IEAOPTxx**
    - LOW will vary between MAX(MCCAFCTH lowvalue, 400, 0.2% of pageable storage)
    - OK will vary between MAX(MCCAFCTH okvalue, 600, 0.4% of the pageable storage)

      SRM will automatically adjust the actual threshold values based on measurements of storage usage but doesn’t let values get lower than MCCAFCTH low threshold
    - Typically no need to specify this parameter

Messages to be aware of

- **REAL – (too much fixed, not enough pageable)**
  - Informational level

    50%
    - Issue ENF 55
    - Issue IRA405I: % of real is fixed
  - Shortage level

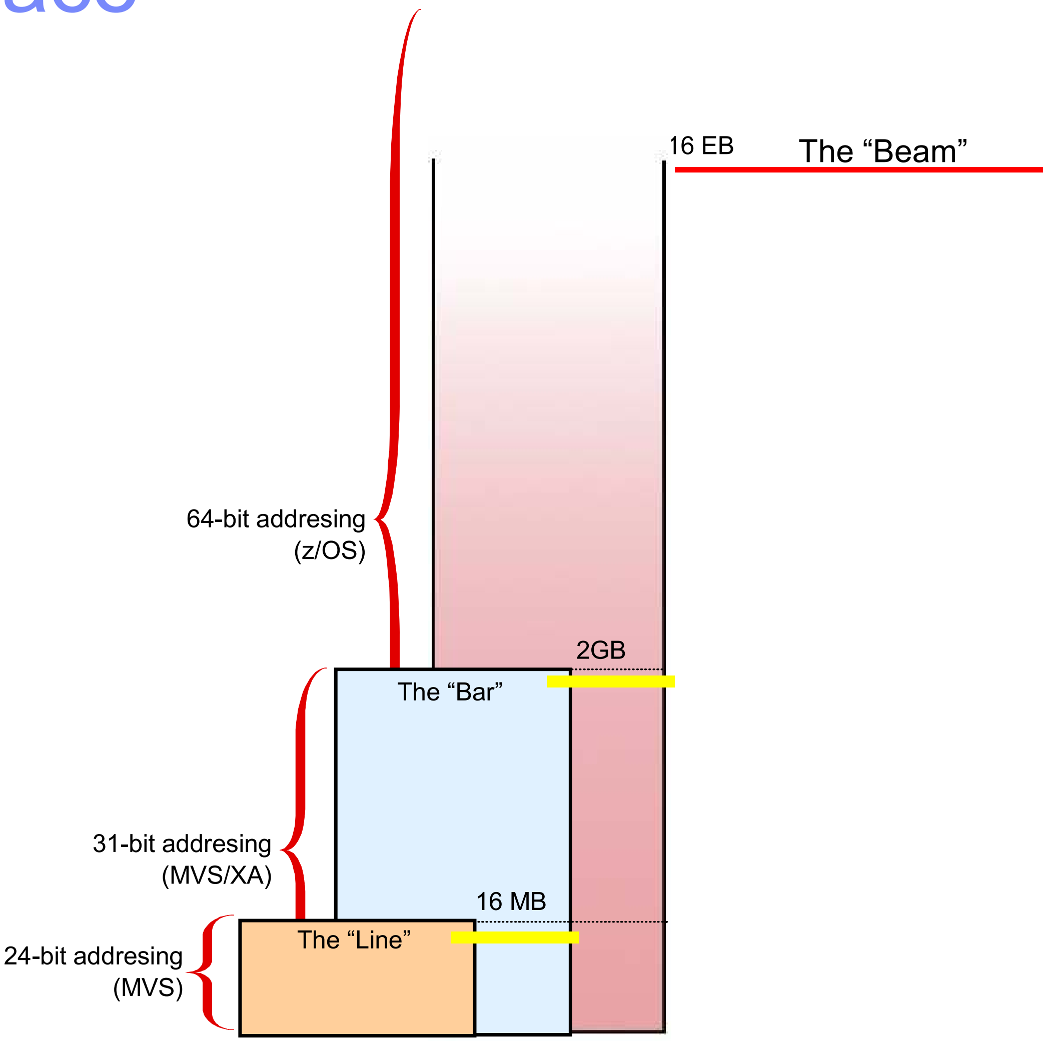
    80%
    - Issue ENF 55
    - Issue IRA400E: shortage of pageable frames and list of top 20 consumers
    - Issue IRA403E swapping culprit out (in-real, moving it higher)
    - Issue IRA410E set non-swappable AS as non-dispatchable (STORAGENSWDP=YES)
  - Critical Shortage level (everything seen in Shortage +)

    90%
    - Issue IRA401E Critical paging shortage
    - If this case continues for 15 seconds issue IRA420I and IRA421D to show largest consumers and allow operator to cancel them

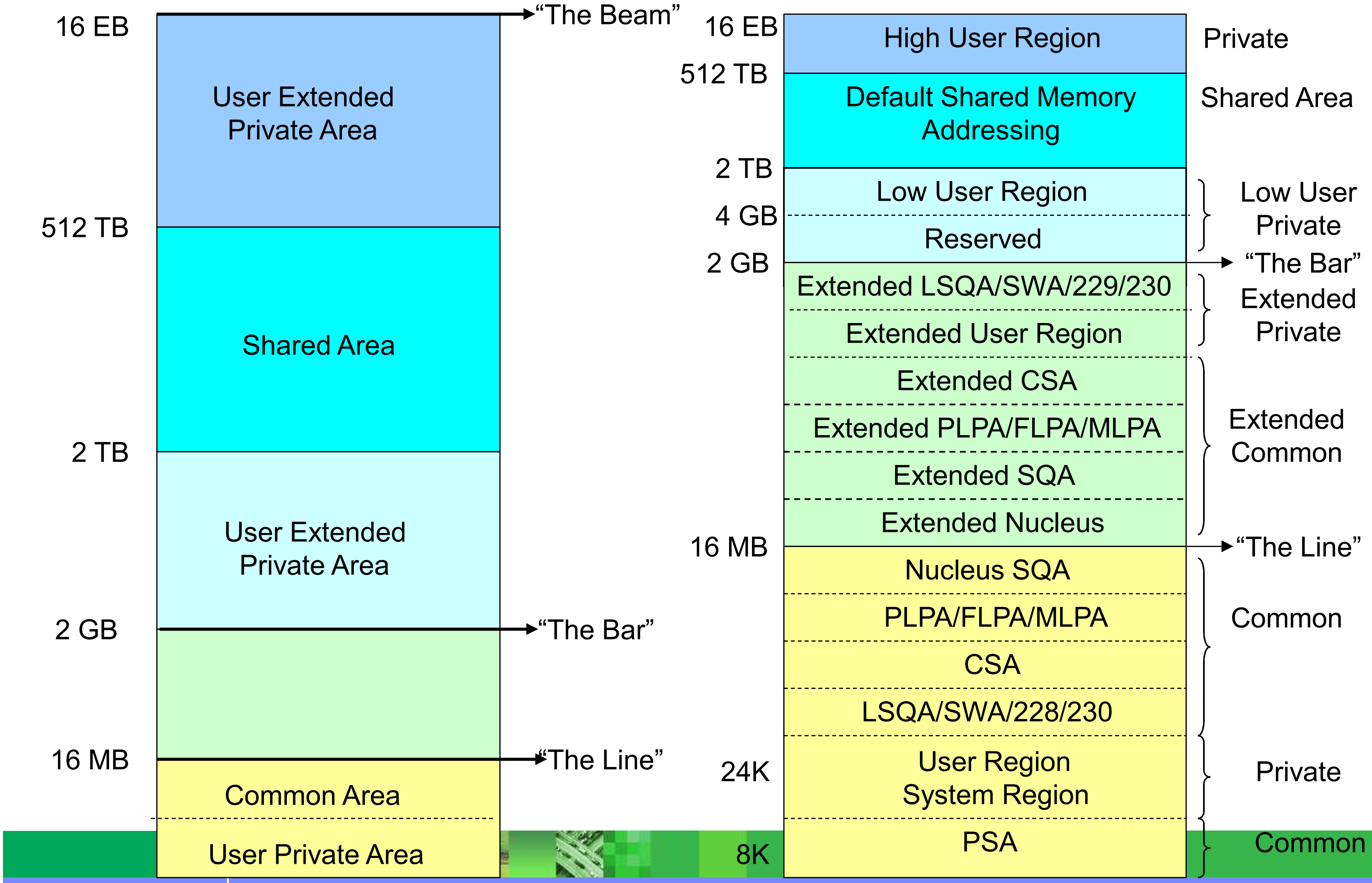
Messages to be aware of…

- **AUX**
  - Informational level — 50%
    - Issue ENF 55
    - Issue IRA205I: % of AUX is allocated
    - Also where DB2 goes into hard DISCARD mode
  - Warning level — 70%
    - Issue ENF 55
    - Issue IRA200E and IRA206I: shortage of AUX and list of top 20 consumers
    - Issue IRA203E swapping culprit out
    - Issue IRA210E set non-swappable AS as non-dispatchable (STORAGENSWDP=YES)
  - Critical Shortage level (everything seen in Shortage +) — 85%
    - Issue IRA201E Critical paging shortage
    - If this case continues for 15 seconds issue IRA220I and IRA221D to show largest consumers and allow operator to cancel them

The address space concept



Address Space Storage - z/OS View



16 EB

"The Beam"

User Extended Private Area

512 TB

Shared Area

2 TB

User Extended Private Area

2 GB

"The Bar"

16 MB

"The Line"

Common Area

User Private Area

16 EB

High User Region

Private

512 TB

Default Shared Memory Addressing

Shared Area

2 TB

Low User Region

4 GB

Reserved

Low User Private

2 GB

"The Bar"

Extended LSQA/SWA/229/230

Extended User Region

Extended Private

Extended CSA

Extended PLPA/FLPA/MLPA

Extended SQA

Extended Nucleus

Extended Common

16 MB

"The Line"

Nucleus SQA

PLPA/FLPA/MLPA

CSA

LSQA/SWA/228/230

Common

24K

User Region
System Region

Private

8K

PSA

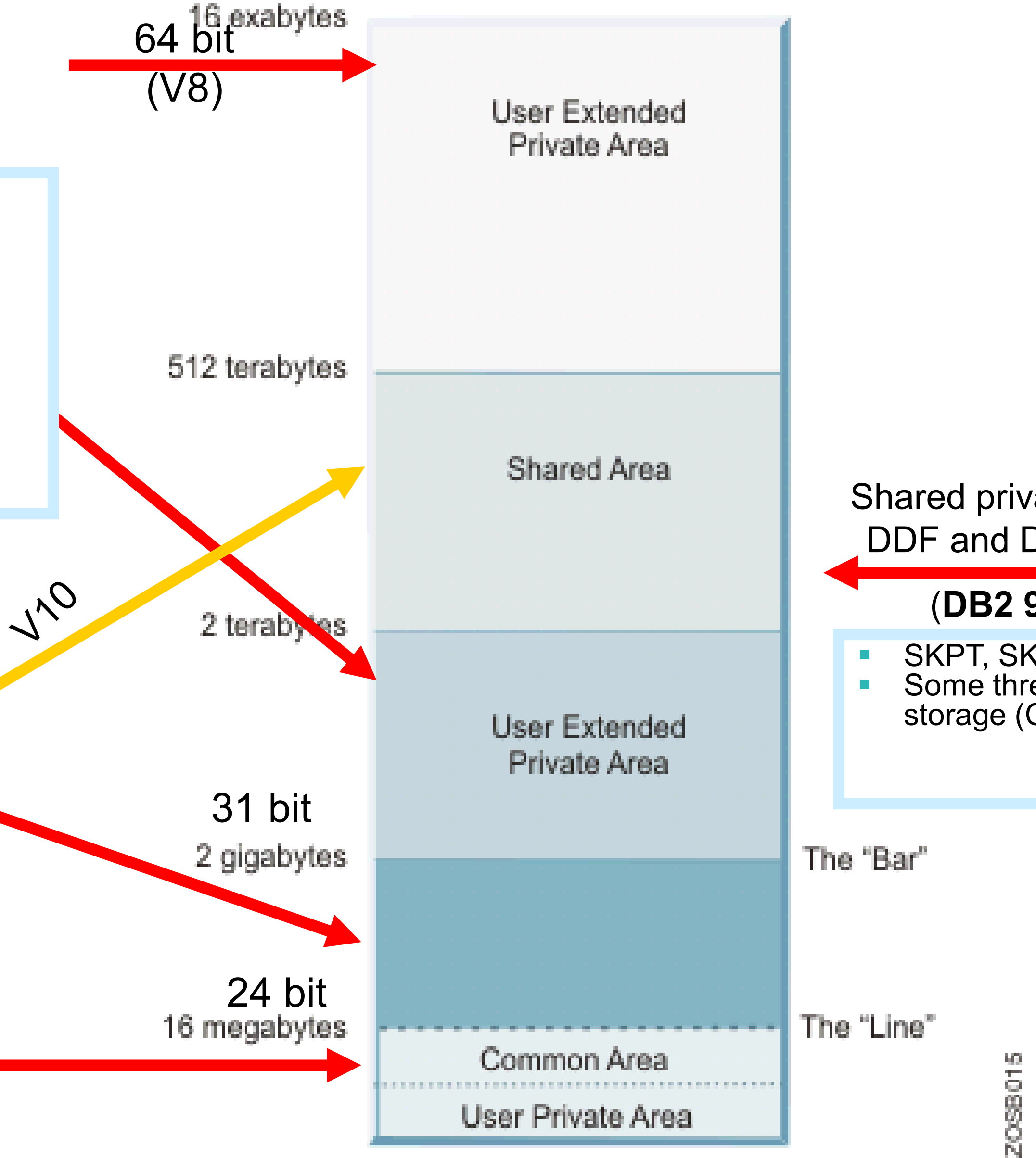
Common

Addressability V9

- EDM above bar:
  - –Statement cache
  - –DBD
- Also above bar (fixed):
  - –RID and SORT pool
  - –Buffer pools
  - –Compression dictionary
- Code and working storage

- EDM below bar:
  - –CTs, PTs,
  - –SKCTs, SKPTs
  - –blocks for cached plans
  - –Variable – stmt cache

- VSAM data set control blocks
  - – About 4k per data set
- Working storage



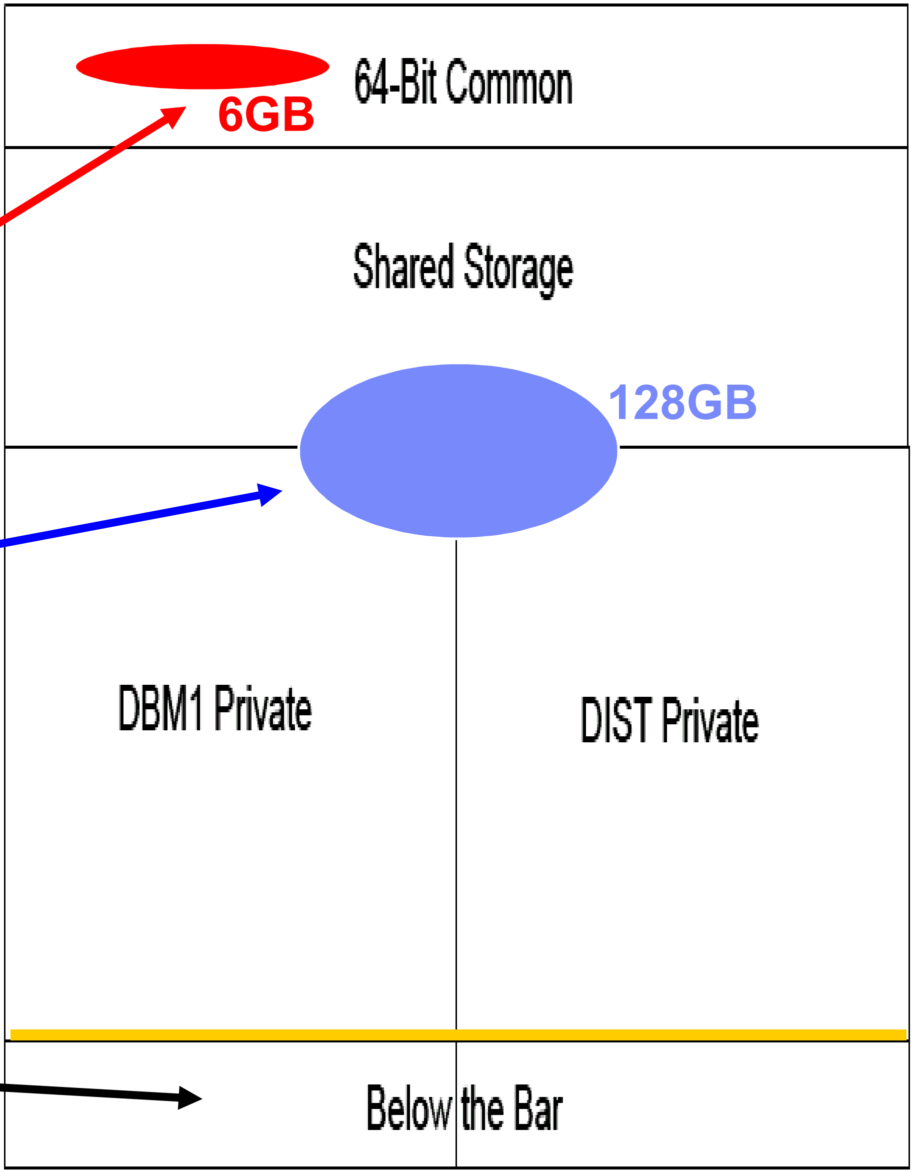
Shared private for
DDF and DBM1

(**DB2 9**)

- SKPT, SKCT
- Some thread storage (CT/PT)

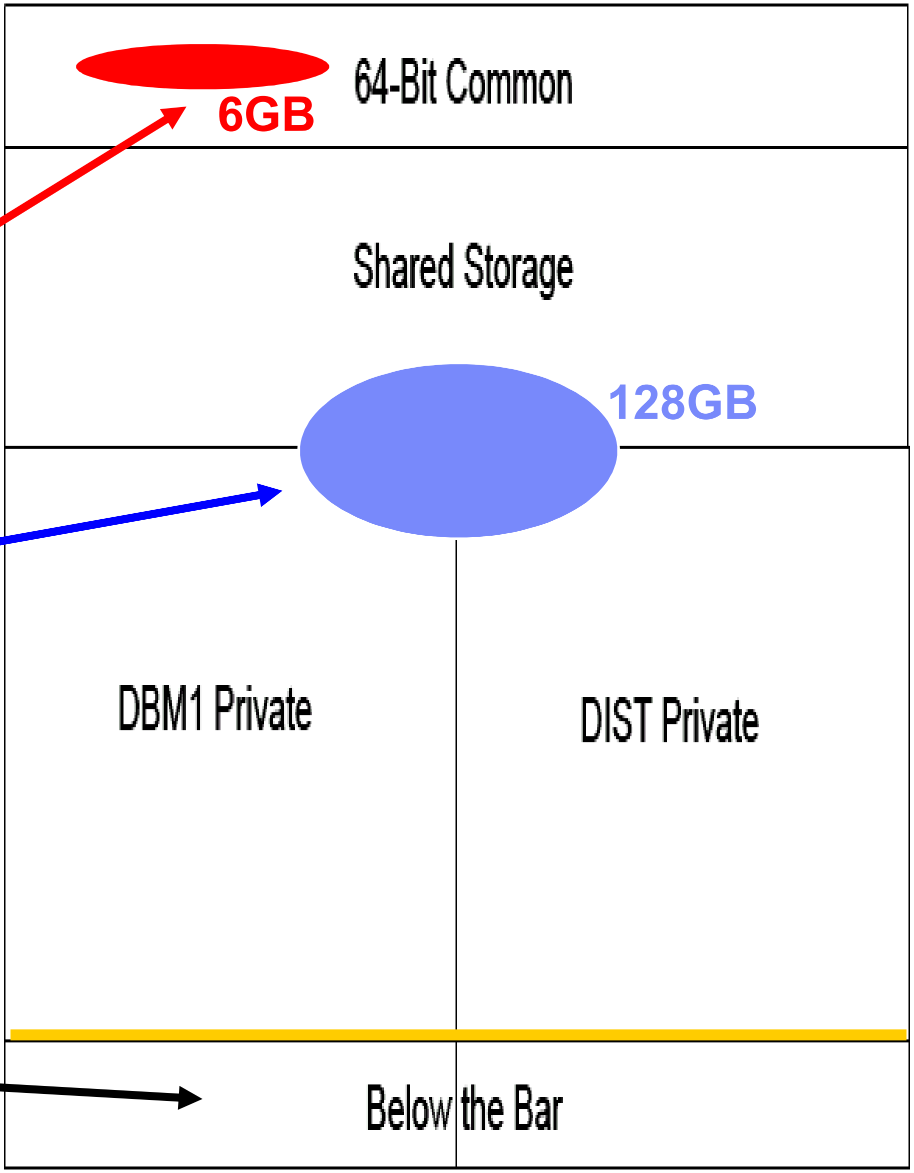
MVS Storage DB2 10

- Almost everything is above the bar:
  - –6GB Common per subsystem: z/OS 1.13 has 64GB as default
    - –If you have many subsystems on same LPAR this may not be enough
  - –128GB Shared per subsystem came in v9:: now in V10 this is where all the thread storage went
    - –In V9 only DRDA comm. area and trusted context ran here
    - –510 TB default in z/OS 1.13
- We still have 31bit stack for agents (threads)
  - –16K for system agents
  - –32k for attach
  - –xPROCS still here, and reported in IFCID 225



MVS Storage DB2 11

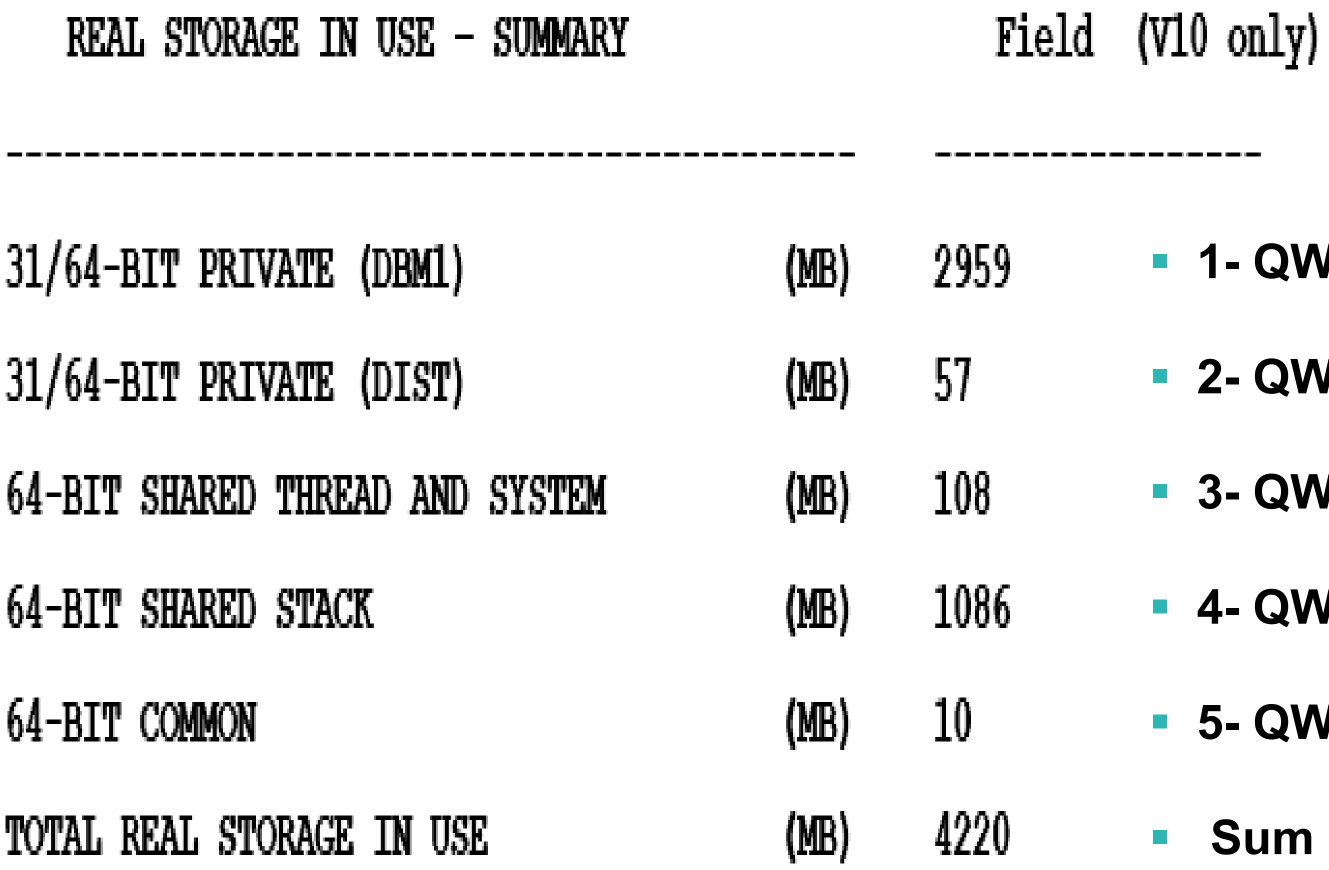
- Almost everything is above the bar:
  - –6GB Common per subsystem: z/OS 1.13 has 64GB as default
    - –If you have many subsystems on same LPAR this may not be enough
    - –OUTBUFF +15% goes here
  - –1TB Shared per subsystem:: xProc storage shared among threads now
    - –510 TB default in z/OS 1.13
    - –xPROCs moved above bar
  - –31-bit low private is now eligible to be backed by 1 MB frames



What affects the storage?

- 64-bit private
  - Mostly buffer pools
  - Fixed-RID pool
  - EDM pool
- 64-bit shared thread and system
  - Used to execute SQL
  - Variable-dynamic statement caches, CT, PT, SKCT, SKPT
  - *Compare to number of system and user agents/threads, parallelism
- 64-bit stack
  - Just as in V9, used to by thread to execute SQL
  - *same
- 64-bit common
  - Distributed agents, package accounting, rollup
- All are affected by REALSTORAGE_MANAGEMENT

Total real In V10



```
  REAL STORAGE IN USE - SUMMARY                 Field  (V10 only)
--------------------------------------------  -----------------
31/64-BIT PRIVATE (DBM1)                (MB)  2959
31/64-BIT PRIVATE (DIST)                (MB)  57
64-BIT SHARED THREAD AND SYSTEM         (MB)  108
64-BIT SHARED STACK                     (MB)  1086
64-BIT COMMON                           (MB)  10
TOTAL REAL STORAGE IN USE               (MB)  4220
```

- **1- QW0225RL (DBM1)**
- **2- QW0225RL (DIST)**
- **3- QW0225SHRSTG_REAL**
- **4- QW0225ShrStkStg_Real**
- **5- QW0225ComStg_Real**
- **Sum 1-5**

- ** If using MEMU2 v10
  - Columns ‘Y’ + ‘BD’ + ‘CU’ + ‘CW’ + ‘CY’ (****Don’t forget AUX counters for each**)

REALSTORAGE_MANAGEMENT

- **OFF** Do not enter contraction mode unless the REALSTORAGE_MAX boundary is approached OR z/OS has notified us that there is a critical *aux shortage*

- **ON** Always operate in contraction mode. This may be desirable for LPAR with many DB2s or dev/test systems

- **AUTO (the default)** When significant paging is detected, contraction mode will be entered, done based on deallocated threads and number of commits, private pools contracted on similar boundaries

- Important notes:
  - Contraction mode is not exited immediately upon relief to avoid constant toggling in and out of this mode
  - Contraction mode shows <1% CPU degradation with no detectable impact to running workloads

Storage Contraction

- Contraction could be entered due to REALSTORAGE_MANAGEMENT (ENF 55) on previous slide, or REALSTORAGE_MAX (zParm) being encroached upon
  - We now report number of contractions in MEMU2 as well (DISCARDMODE64 or RSMAX_WARN)

- Contraction mode start is indicated by a DSNV516I message
  - **DSNVMON – BEGINNING STORAGE CONTRACTION MODE**

- Ending with a DSNV517I message
  - **DSNVMON – ENDING STORAGE CONTRACTION MODE**

Storage manager changes (V10 via maint.) ZPARM RSM REALSTORAGE_MANAGEMENT=xx

After PM88804

- ENF 55 signal means DISCARD KEEPREAL=NO (50% AUX left)
- RSM=OFF means No DISCARD
- RSM=AUTO with no paging means *No DISCARD* at Thread Deallocation or 120 commits
- RSM=AUTO with paging or RSM=ON means DISCARD with *KEEPREAL=NO* at Deallocation or 30 commits. STACK also DISCARDED
- REALSTORAGE_MAX means DISCARD KEEPREAL=NO at 80%

After PM99575

- ENF 55 signal means DISCARD KEEPREAL=NO (50% AUX left)
- RSM=OFF means No DISCARD
- RSM=AUTO with no paging means *DISCARD* with KEEPREAL=YES at Thread Deallocation or 120 commits
- RSM=AUTO with paging or RSM=ON means DISCARD with *KEEPREAL=YES* at Deallocation or 30 commits. STACK also DISCARDED
- REALSTORAGE_MAX means DISCARD KEEPREAL=NO at 100%

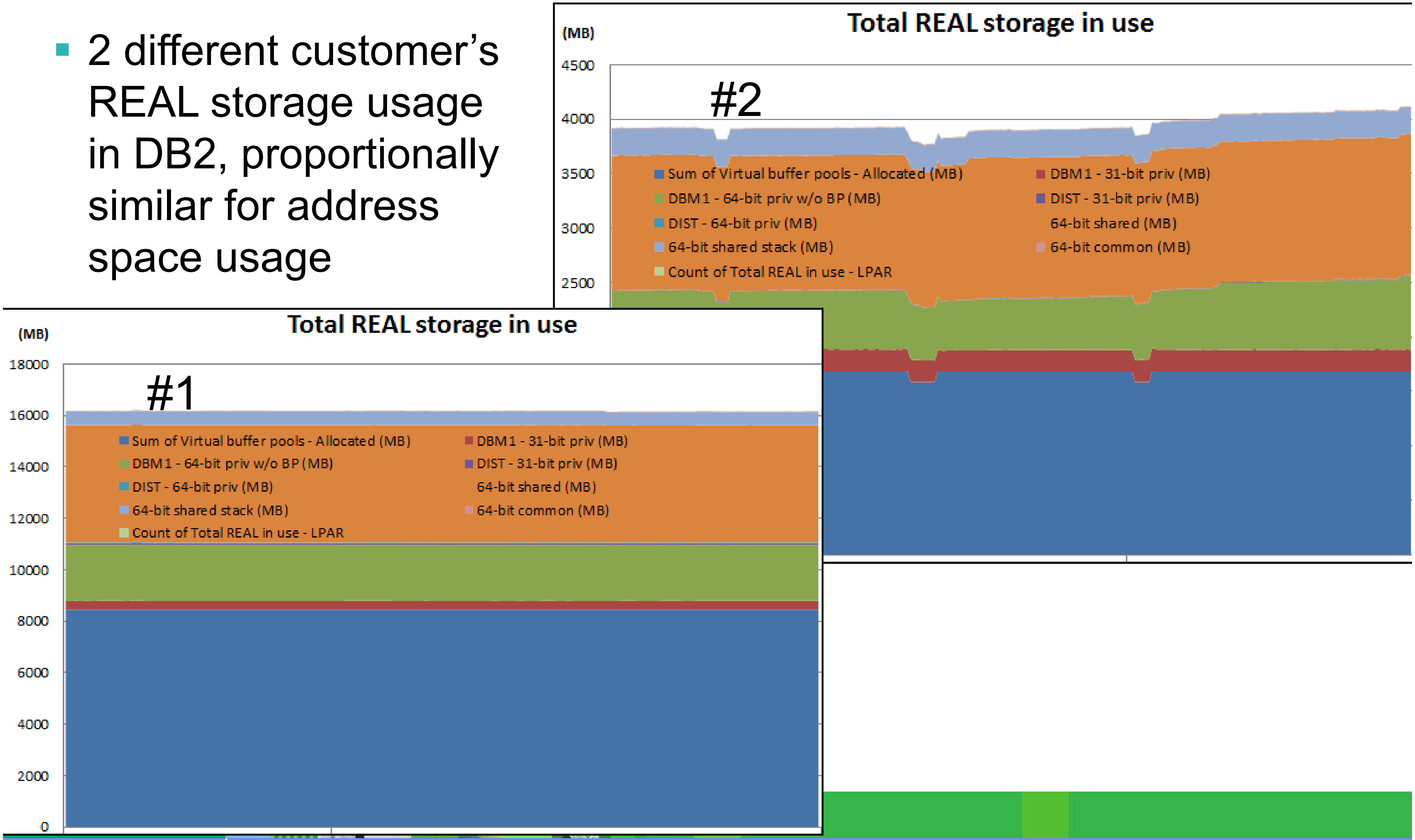
** Apply OA44207 to improve RSM efficiency

XCF Critical Paging – avoid page faults during HyperSwap

- The downside of this is a massive amount of page fixed storage to include the following:
  - 31- bit common storage (both above and below 16M)
  - Address spaces that are defined as critical for paging
  - All data spaces associated with those address spaces that are critical for paging (unless CRITICALPAGING=NO was specified on the DSPSERV CREATE)
  - Pageable link pack area (PLPA)
  - Shared pages
  - All HVCOMMON objects
  - All HVSHARED objects
- Apply z/OS APAR OA44913
  - Allows z/OS to reclaim DB2 64-bit SHARED KEEPREAL=YES frames
- In DB2 the 64-bit SHARED houses thread working storage, statement cache, SKCT/SKPT

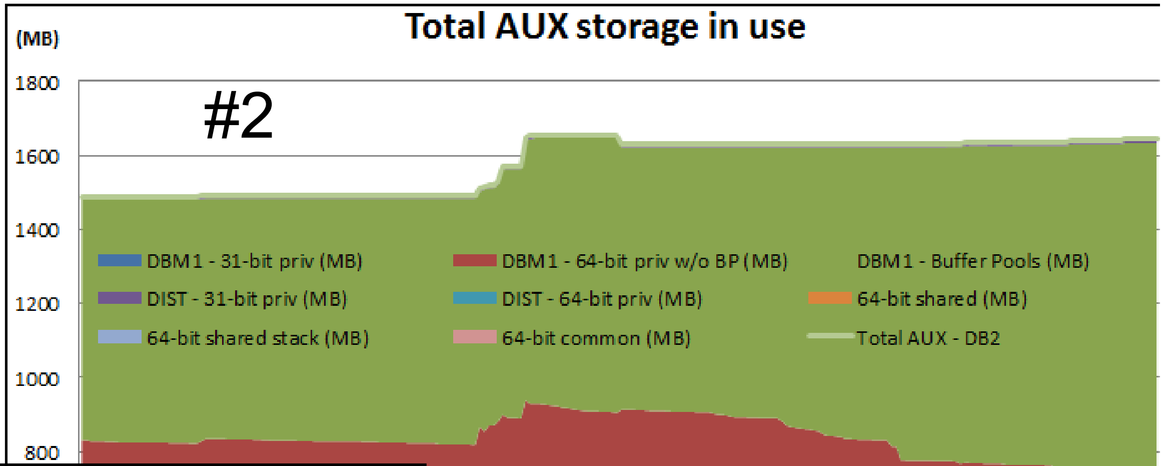
What is in REAL now?

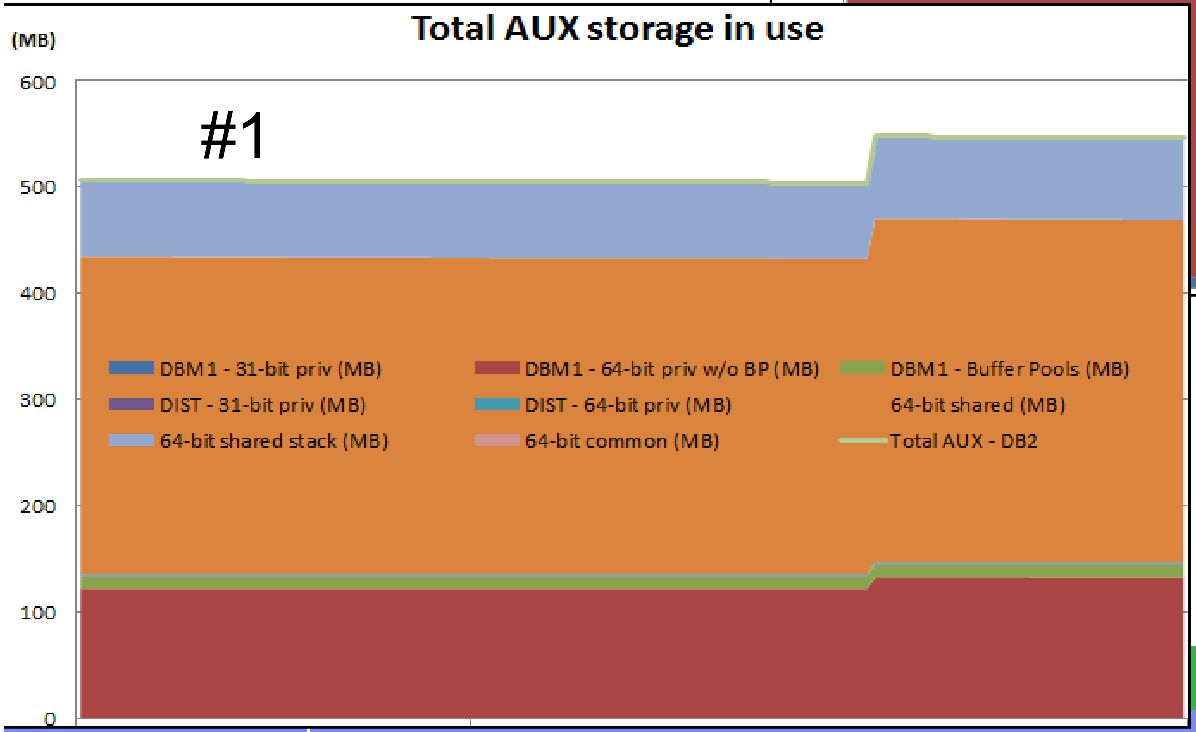
- 2 different customer's REAL storage usage in DB2, proportionally similar for address space usage



What is in AUX now?

- With CRITICALPAGING =YES HVSHARE becomes non-pageable so that leaves buffer pools and PRIVATE to be sacrificed

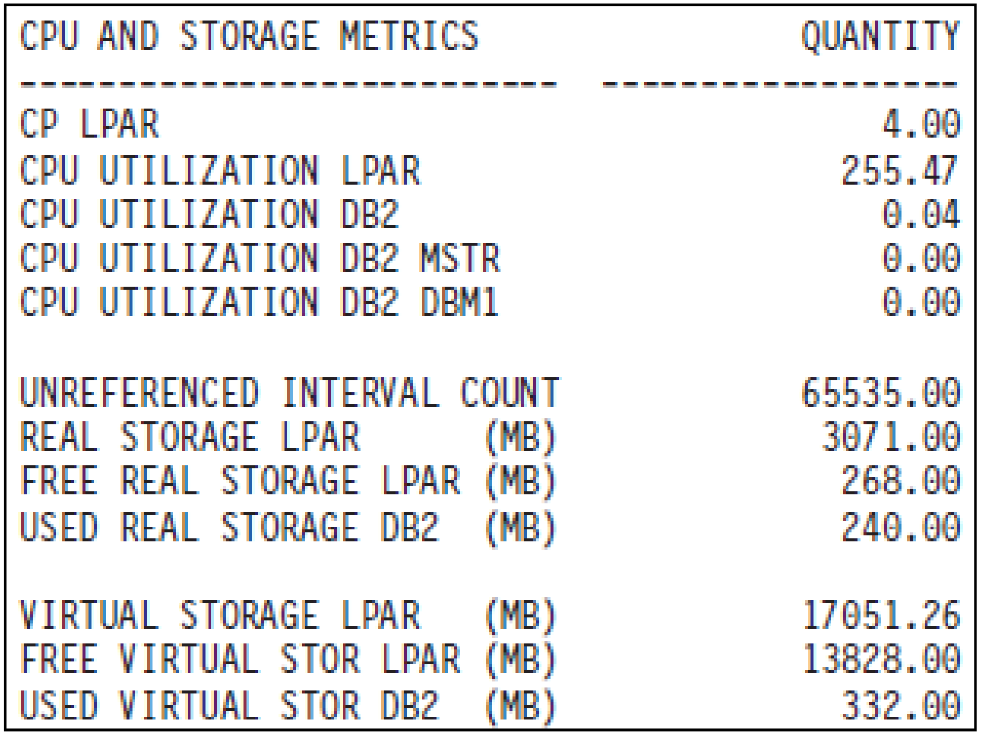




- Buffer pools are not paged out in customer #1's environment, but they are in #2 causing more I/Os (no prefetch)

z/OS Metrics

- ZOSMETRICS = YES
  - Collected by RMF and put into IFCID 001
  - Appears in stats long report



```
CPU AND STORAGE METRICS                QUANTITY
---------------------------   ------------------
CP LPAR                                    4.00
CPU UTILIZATION LPAR                     255.47
CPU UTILIZATION DB2                        0.04
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)              268.00
USED REAL STORAGE DB2   (MB)             240.00

VIRTUAL STORAGE LPAR    (MB)           17051.26
FREE VIRTUAL STOR LPAR (MB)            13828.00
USED VIRTUAL STOR DB2   (MB)             332.00
```

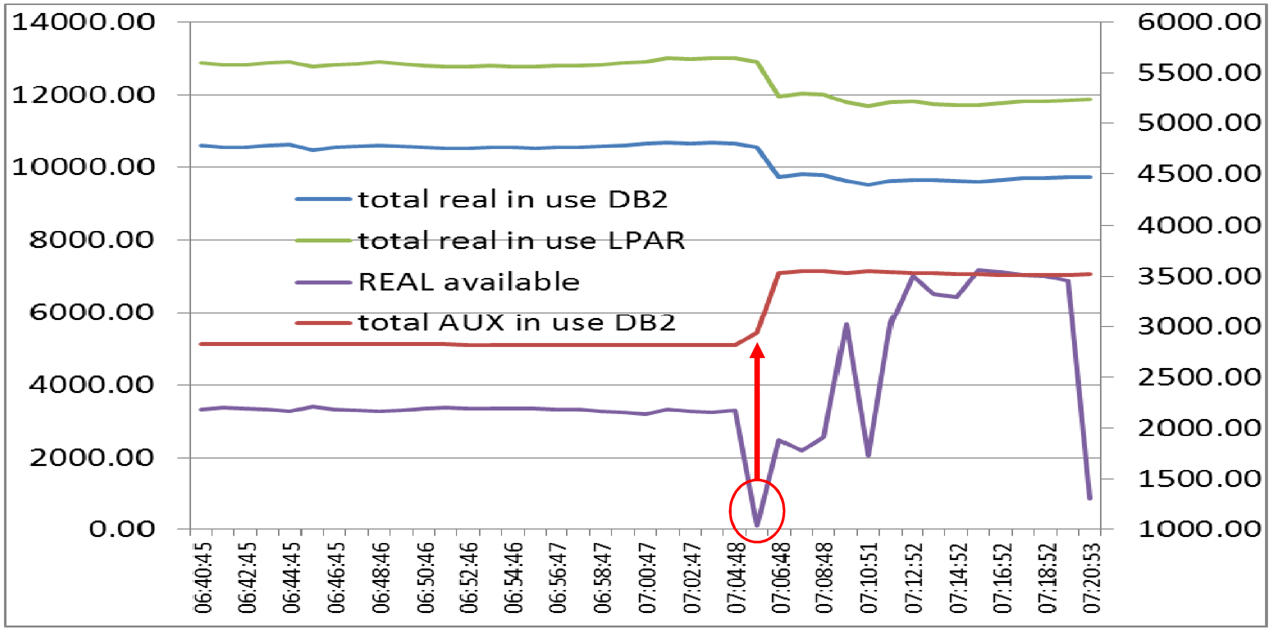
Paging Public Service Announcement… Why?

- What is a tolerable paging rate?
  - Only during dramatic workload shifts, hoping storage isn't needed anymore??
    - i.e. batch to online and vice versa?
  - This inherently means you have no 'cushion' for DUMPs, unexpected SORTing or Utility/Batch job which ran at an inopportune time….
- ….DB2's answer is never

- Most customers have undersized dev/test environment….
  - **But at ~5us synch I/O (+1us no page fixing) and ~20us for prefetch (+5us for no page fixing) you are trading CPU for REAL memory
    - DB2 path length measurement… much bigger if measured end-to-end
  - So why not right-size it to save CPU, and actually mirror the performance of what it will be in production??
- Storage $$$$ → $$ so trade your CPU cycles for Memory

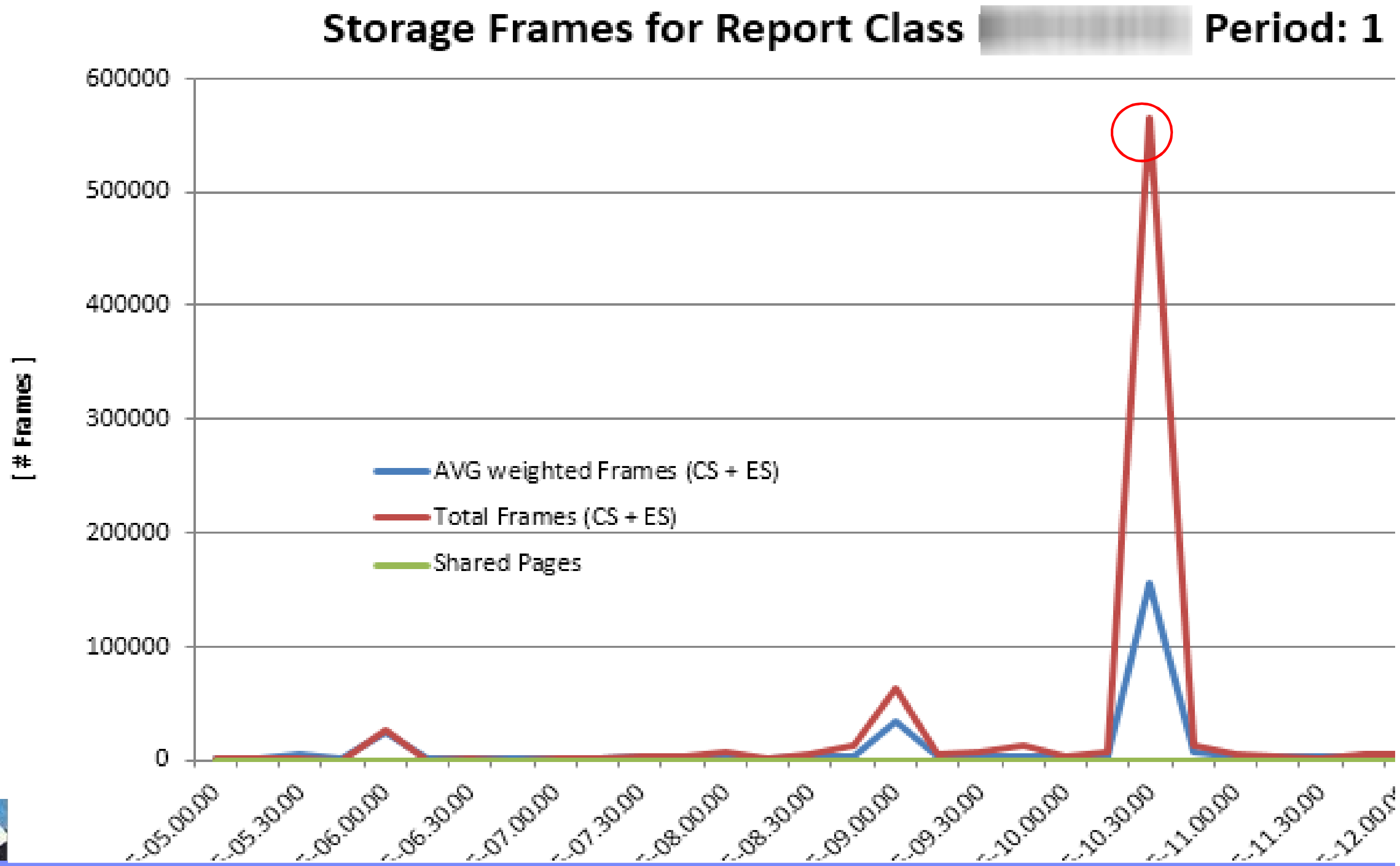
DB2 Storage

- In the graphic we can see DB2 storage goes out from REAL to AUX when the real available drops to '0' on the LPAR
- Worst case in this example to get those pages back in:
  - 700 MB – sync I/O time ~3ms = 0.003*179,200 = 537 seconds
  - If those pages were taken out of our bufferpools then we need to spend the I/Os to get the pages back in central storage – no prefetch from AUX
- Imagine a 16GB SVCDUMP occurring here!!
  - MAXSPACE = 16GB to allow for dump, should you reserve this space during peak processing hours?
  - MAXSNDSP = 15 seconds default (amount of time system non-dispatchable)
  - AUXMGMT=ON – no new dumps when AUX=50%, MAXSPACE always honored



DB2 Storage

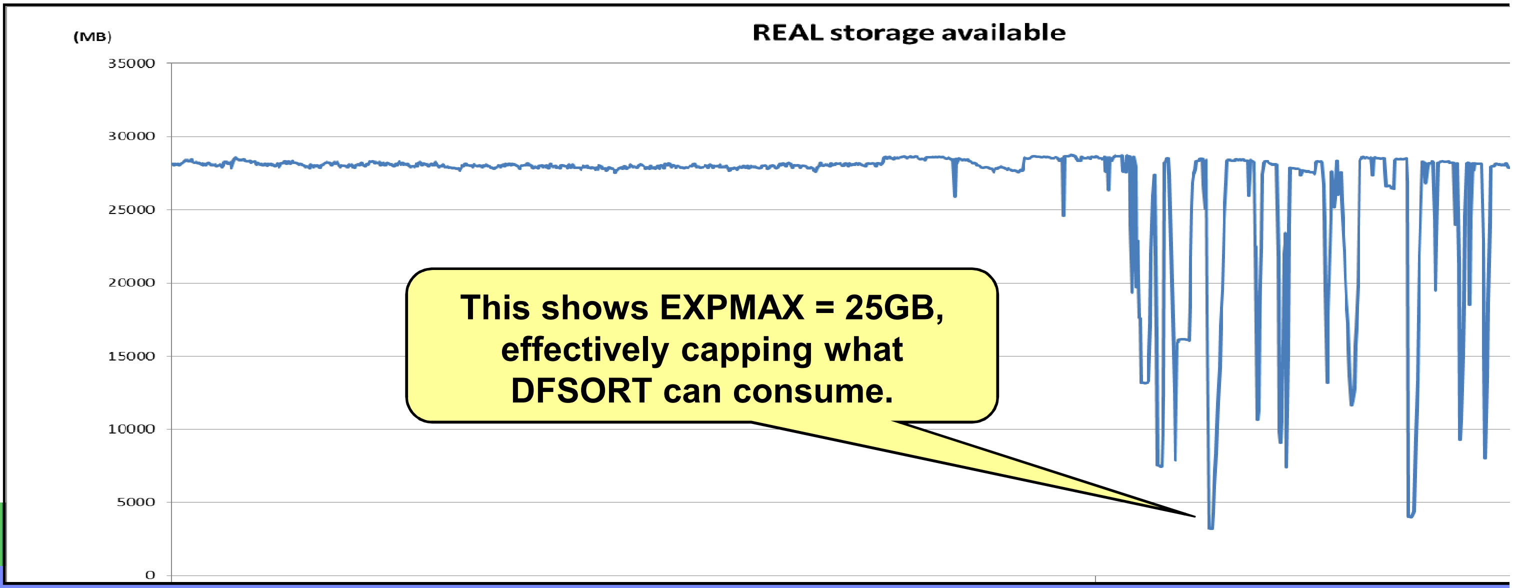
- So who caused me to get paged out??
  - If you run a WLM activity report and look at the Storage Trend graph in the reporter you can see the actual frames used by a service or report class
  - The Page In Rates would also be high for that report class as data is brought in



Storage Frames for Report Class [redacted] Period: 1

Real storage and Sort products

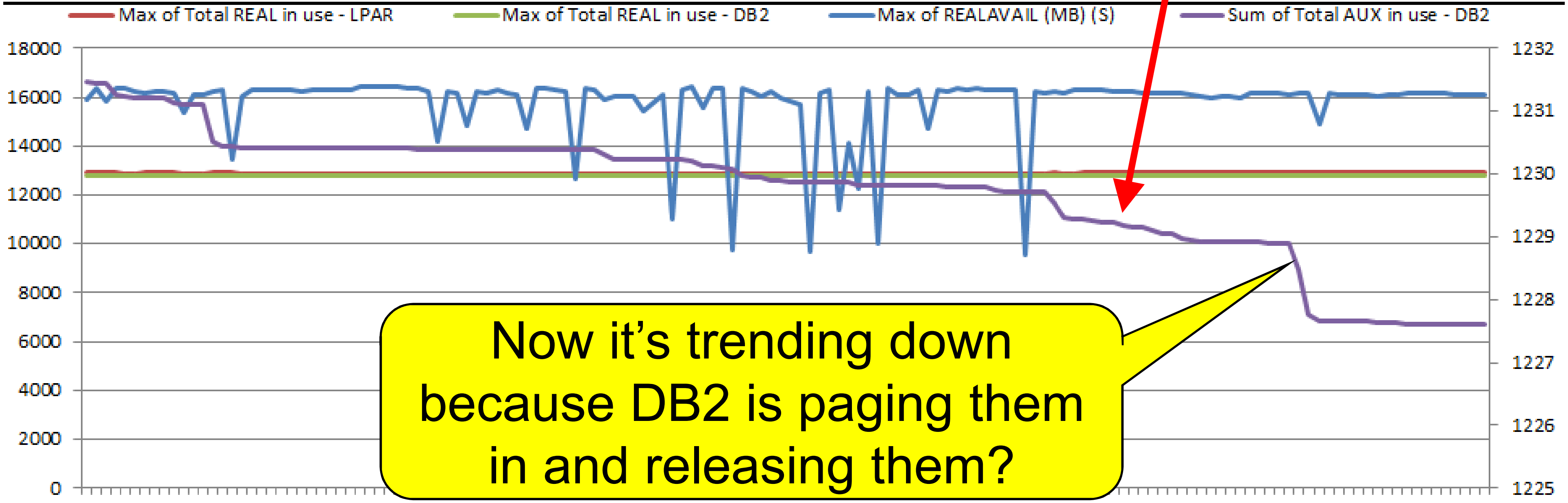
- By default DFSORT and other sort products usually take as much storage as they can get, to help performance… but what about everyone else?
- DFSORT parameters affecting storage use (II13495) → means to protect DB2
  - These can be dynamically changed for workloads using ICEPRMxx member
  - EXPOLD = % of storage allowed to be pushed to AUX → 0
  - EXPRES= % of storage to preserve, maybe in case of DUMPSPACE/ MAXSPACE → 16GB min in V10
  - EXPMAX=% of storage for memory object and hyperspace sorting, somewhat depends on EXPOLD and EXPRES → how much can you spare
- DB2Sort has PAGEMON=ON/OFF to limit central storage usage



This shows EXPMAX = 25GB, effectively capping what DFSORT can consume.

More on AUX storage

- AUX storage has been referred to as ‘double accounting’
- Once the page data sets (AUX slots) are utilized by an address space they remain assigned to that address space
- UNTIL ‘DB2’ is bounced, ***or we page them in and release the AUX slot***… it looks like we are using it..
  - We have a requirement out to z/OS to address this



Now it’s trending down because DB2 is paging them in and releasing them?

Long-Term Page Fix for BPs with Frequent I/Os

- DB2 BPs have always been strongly recommended to be backed up 100% by real storage
  - To avoid paging which occurs even if only one buffer is short of real storage because of LRU buffer steal algorithm
    - ***<u>ROT: In a steady-state</u>**: PAGE-IN for READ / WRITE <1% of pages read / written*
- Given 100% real storage, might as well page fix each buffer just once to avoid the repetitive cost of page fix and free for each and every I/O
  - New option: ALTER BPOOL(name) PGFIX(YES|NO)
    - Requires the BP to go through reallocation before it becomes operative
      - Means a DB2 restart for BP0
  - Up to 8% reduction in overall IRWW transaction CPU time
    - About 1,000 instructions for fix/free

Long-Term Page Fix for BPs with Frequent I/Os

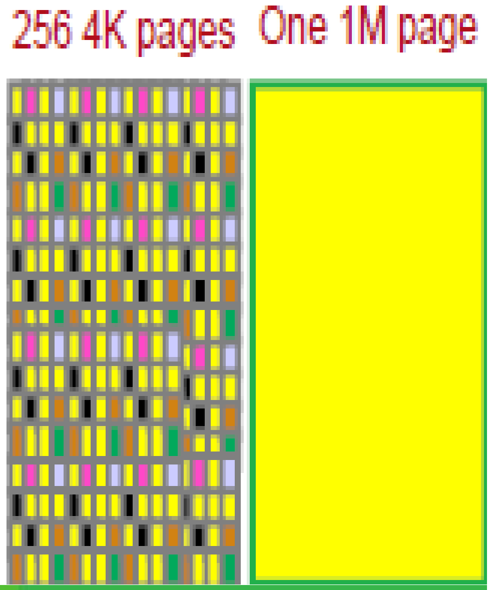
- Recommended for BPs with high I/O intensity
  - I/O intensity = [pages read + pages written] / [number of buffers]
  - Relative values across all BPs

BPID	***VPSIZE***	***Read Sync***	***Read SPF***	***Read LPF***	***Read DPF***	***Read - Total***	***Written***	***I/O Intensity***
BP0	40000	2960	0	0	6174	9134	107	0.2
BP1	110000	12411	5185	0	1855	19451	6719	0.2
BP2	110000	40482	19833	11256	9380	80951	5763	0.8
BP3	75000	23972	6065	0	14828	44865	7136	0.7
BP4	80000	22873	45933	3926	50261	122993	3713	1.6
BP5	200000	0	0	0	0	0	0	0.0
BP8K0	32000	9	11	0	11	31	169	0.0
BP32K	2000	693	873	0	6415	7981	38	4.0

In this example: Best candidates would be BP32K, BP4, BP2, BP3
No benefit for BP5 (data in-memory)

1MB Frames

- TLB – translation lookaside buffer is a ‘fast-path’ through translation between virtual and REAL
  - coverage today represents a much smaller fraction of an application’s working set size leading to a larger number of TLB misses
- Applications can suffer a significant performance penalty resulting from an increased number of TLB misses
- Solution:
  - Increase TLB coverage without proportionally enlarging the TLB size by using large pages
- Large Pages allow for a single TLB entry to fulfill many more address translations
- Large Pages will provide exploiters with better TLB coverage



1MB Large Frames for BPs with highest getpages

- First exploited by DB2
  - Page fixed buffer pools in DB2 10
    - As opposed to I/O intensity, use large frames for pools with highest getpage counts
  - DB2 11 brings
    - 31-bit low private system storage
    - non-page fixed BP control blocks (PMBs) if it is FlashExpress compatible HW
    - OUTBUFF backed with 1MB frames
  - JAVA heap storage
- To allocate 1MB frames IEASYSxx in PARMLIB governs z/OS storage allocations
  - LFAREA=(1M=(*target*[%],*minimum*[%])
  - 2G=(*target*[%],*minimum*[%]))
    - We try to meet the targets, but if minimums cannot be met a console message is issued

Buffer Pool enhancements…

Frame size	Page fix	Supported DB2	H/W Requirement	Benefit
4K	NO	All	N/A	Most flexible configuration
4K	YES	All	N/A	CPU reduction during I/O
1M	NO	DB2 10 with APAR, or DB2 11	zEC12 and Flash Express compatible Backed by real or LFAREA	CPU reduction from TLB hit Control Blocks only
1M	YES	DB2 10 above	z10 above LFAREA 1M=xx	CPU reduction during I/O, CPU reduction from TLB hit
2G	YES	DB2 11	zEC12 LFAREA 2G=xx	CPU reduction during I/O, CPU reduction from TLB hit

*** If any HW/SW requirements unmet, 4K frames used**

What about LFAREA?

- Useful commands
  - -DISPLAY BUFFERPOOL(BP1) SERVICE=4
    - Useful command to find out how many 1MB size page frames are being used

DSNB999I =D2V1 4K PAGES 0 DSNB999I =D2V1 1M PAGES 1476

  - -DISPLAY VIRTSTOR,LFAREA

```
IAR019I 14.37.22 DISPLAY VIRTSTOR 735
SOURCE = WE
TOTAL LFAREA = 64M
LFAREA AVAILABLE = 32M
LFAREA ALLOCATED (1M) = 16M
LFAREA ALLOCATED (4K) = 4M
LFAREA ALLOCATED (PAGEABLE1M) = 12M
MAX LFAREA ALLOCATED (1M) = 62M
MAX LFAREA ALLOCATED (4K) = 5M
MAX LFAREA ALLOCATED (PAGEABLE1M) = 14M
```

NOT good, this means we broke down some of the 1MB frames

We reserve 1/8th of real on LAPR for pageable frames

  - Show total LFAREA, allocation split across 4KB and 1MB size frames, what is available

How do I size LFAREA? (V10)

- **Do not oversize LFAREA**
  - LFAREA used ~ sum of page fixed buffer pools and JAVA heap (verbosegc traces)
  - Can't do anything about it until an IPL, if too small just means there is potential savings you could be missing out on
    - **- IRA127I 100% OF THE LARGE FRAME AREA IS ALLOCATED** = using it all
  - Just information as of OA39941; previously was an 'E' or action message
  - If for any reason RSM denies DB2 request for 1 MB frame, uses 4k instead

- **Decomposing/coalescing 1MB frames into 4k frames is very CPU intensive because we are always trying to build the 1MB frames up to meet the LFAREA demand**

  - LFAREA ALLOCATED (4K) = 5M → Not a good sign

  - This indicates you do not have enough REAL storage needed by 4k frames, so add more real or make LFAREA smaller

BP Summary

- Potential for reduced for CPU through less TLB misses
  - CPU reduction based on customer experience 0 to 6%
  - Buffer pools must be defined as PGFIX=YES to use 1MB size page frames
  - Must have sufficient total real storage to fully back the total DB2 requirement
- **Page fixing has highest benefit for pools with high I/O intensity**
- **Backing with larger frames has highest benefit for pools with high get page counts**
- Involves partitioning real storage into 4KB and 1MB size page frames
  - Specified by LFAREA xx% or n in IEASYSnn parmlib member
  - Only changeable by IPL so pad it out!!
- If 1MB size page frames are overcommitted, DB2 will use 4KB size page frames
  - Recommendation to add 5-10% to the size to allow for some growth and tuning
  - Must have both enough 4KB and enough 1MB size page frames
- **Apply PI12512 (V10) – need minimum of 6656 buffers for pool to be backed with 1 MB frames

FlashExpress

- FlashExpress is simply solid state disc for z196 and up machines
- Generally speaking for performance…
  - Access to real memory ~ 1,000 machine cycles
  - Access to FlashExpress ~ 100,000..
  - Access to AUX storage ~ 1,000,000..
- So is it a substitute for REAL?? - No
  - Never intended for buffer pools or working storage
    - Prefetch is disabled!!
  - Better than AUX for backing MAXSPACE (DUMPSERV)
    - 16GB recommended for V10 and up
- Performance numbers and a better description:
  - http://public.dhe.ibm.com/common/ssi/ecm/en/zss03073usen/ZSS03073USEN.PDF
  - 5x reduction in dump time, 3x reduction in non-dispatchable time compared to going out to AUX
- Do not need it for pageable 1MB frames – but if paged out becomes 4k
  - Non-PGFIX buffer pool control blocks
  - DB2 31-bit low private code

References

- Techdoc for V10 and V11 MEMU2 with spreadsheet sample
  - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS5279
- Subsystem and Transaction Monitoring and Tuning with DB2 11 for z/OS
  - http://www.redbooks.ibm.com/redpieces/abstracts/sg248182.html?Open

The End!