Large Memory: Boosting DB2 for z/OS Performance

Mark Rader
DB2 for z/OS Specialist
IBM Corporation

Delivered to MDUG
November 18

2015

IBM Systems Technical University

IBM z Systems • IBM Power Systems • IBM Storage

October 5–9 | Hilton Orlando, Florida

Please Note

- IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

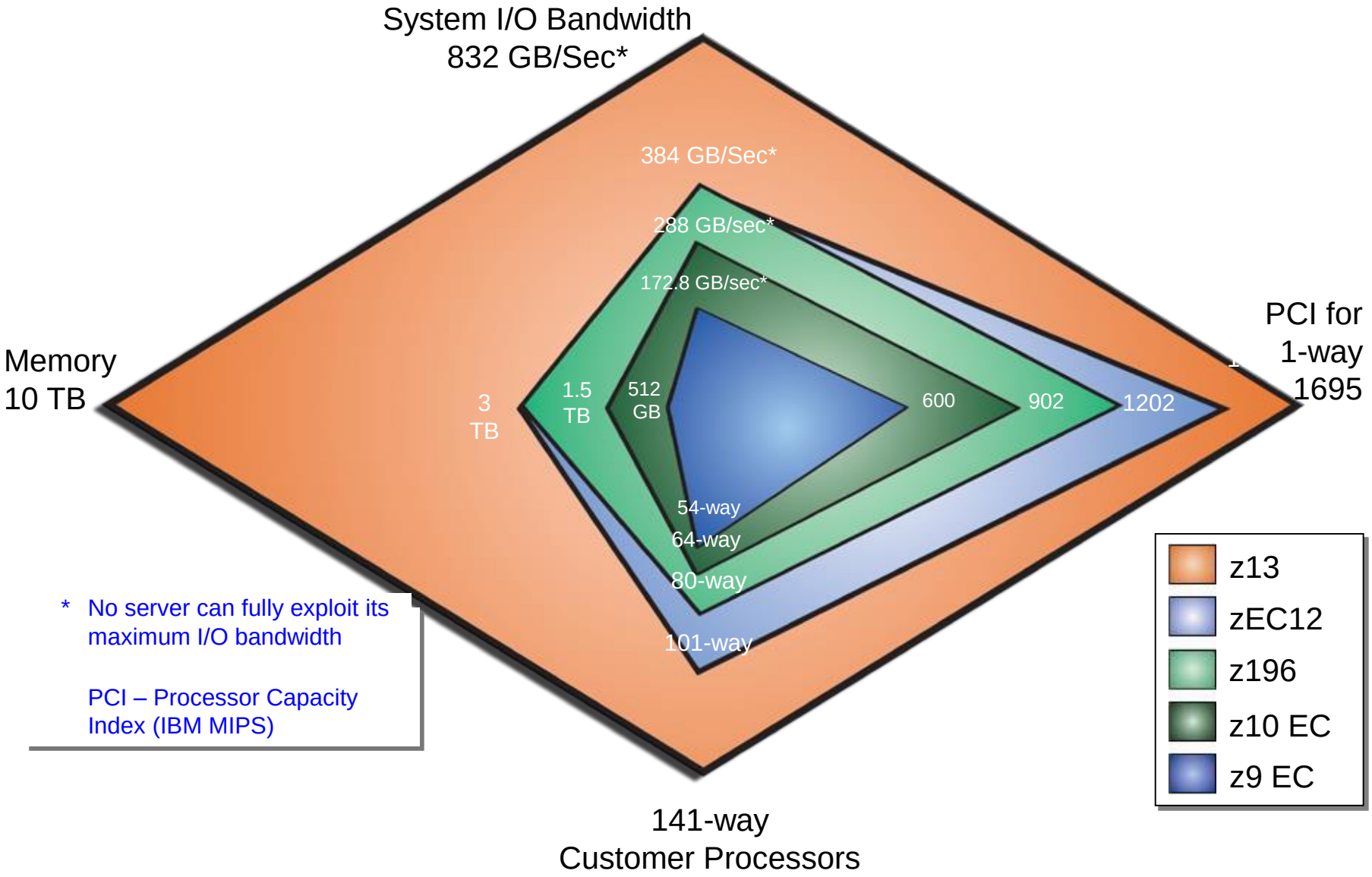
Objectives

- Exploiting memory to achieve better performance
  - More memory with z13
  - Lower TCO by reducing CPU time while achieving lower I/O response time
- What happens when z/OS starts to run out of memory, how to prevent this
  - Paging
  - Dump considerations
  - Batch window and DFSORT considerations
  - SYSPLEX sympathy sickness
  - DISCARD processing
  - DB2 and system parameters for controlling memory
  - DB2 and system APARs that you should know about it
- Monitoring REAL/AUX usage

DB2 and Large Memory

"Memory is cheap or one time charge, CPUs are expensive"
"For every I/O that you save, you avoid the software charge for the CPU that it took to otherwise do that I/O"

IBM z13: Advanced system design optimized for digital business

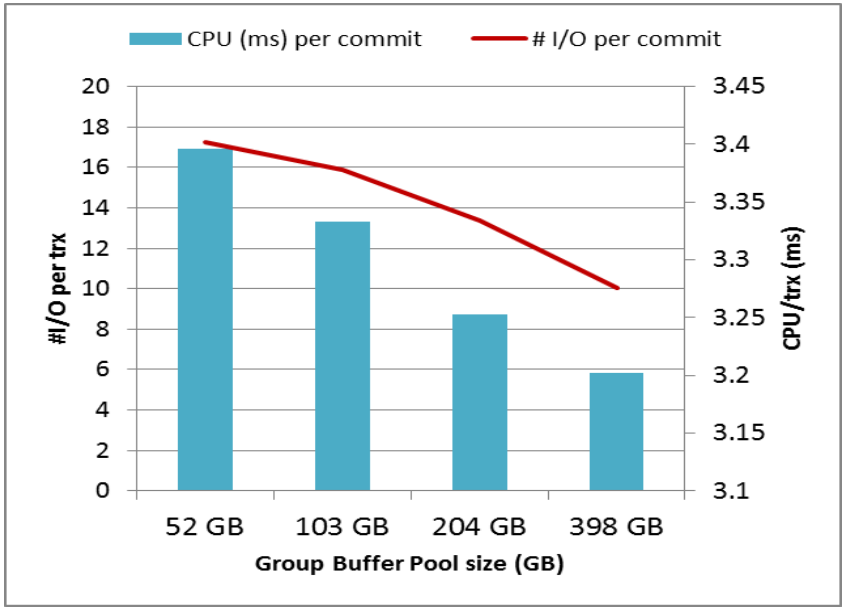


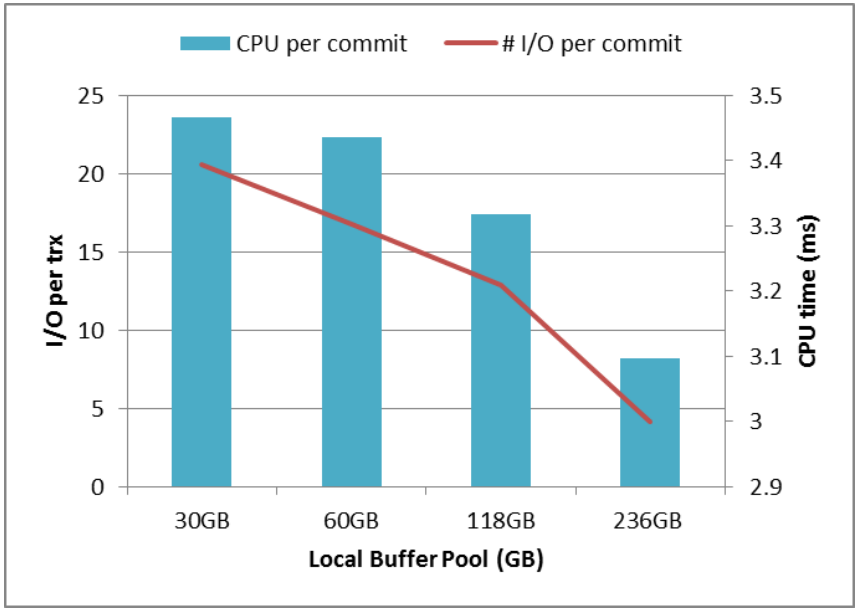
Memory is cheap

- Make sure LPAR has enough REAL storage
- REAL storage upgrade is the cheapest and easiest performance upgrade
  - REAL storage shortage not only can cause performance issues. If DB2 needs to create a dump, it can cause a small issue to become a massive SYSPLEX failure
  - Cheapest because MLC and other charges do not factor into the amount of REAL storage
  - Vendors do not charge by the amount of REAL on the CEC/CPC processor

CPU Cost Saving by Reducing DB2 Synch I/Os

- Banking (60M account) workload with 2 way data sharing :
- Reduce 11 % response time and 6 % CPU by increasing GBP from 52GB to 398GB with same LBP size (60GB) for both members
- Reduce 40% response time and 11% CPU by increasing LBP from 30GB to 236GB for both members with same reasonable GBP size (60GB)





Benefit of Larger Buffer Pools

- Larger buffer pools can potentially reduce CPU usage by reducing synch I/Os
  - z/OS team measures approx. 20-40us CPU per one synch I/O
  - The benefit depends on the size of active workload and access pattern
    - May not see any benefit if working set is very small and already fit in the buffer pools today
    - May not see any benefit if working set is too large and increment is not large enough
  - Pages have to be re-referenced – not for one time sequential. read
    - There is more value if pages are not prefetched.
- Try & validate, may not work well with customer's workload with high variations
- Available tool requires expensive set of traces and intensive analysis

Buffer Pool Simulation (DB2 11 PI22091)

- Simulation provides accurate benefit of increasing buffer pool size from production environment
- ALTER BUFFERPOOL now supports simulation pools
  - To simulate the case of doubling the current 20,000 buffer pools with simulated VPSEQT of 30

```
-ALTER BPOOL(BP1)  VPSIZE(20000)  SPSIZE(20000)  SPSEQT (30)
```

- For example, if you want all of the buffer pool growth to be used for random Getpages, set SPSEQT to 0. Default is SPSEQT=VPSEQT.
- Simulation is against local buffer pools, not group buffer pools. It supports local buffer pools with GBP dependent objects
- Storage cost for a simulated buffer pool is less than 2% for 4K pages
  - SPSIZE(2000) or 7.8MB of simulated buffer pool will use about 156KB of storage.

Buffer Pool Simulation – Output

- Output from simulation is written in statistics traces and DISPLAY buffer pool output
- OMPE V520 APAR to format statistics (APAR PI28338)

  DISPLAY BPOOL DETAIL

```
DSNB431I  -CEA1 SIMULATED BUFFER POOL SIZE = 20000 BUFFERS -
             ALLOCATED       =    20000
             IN-USE          =    20000   HIGH IN-USE     =    20000
                    SEQ-IN-USE      =     2229   HIGH SEQ-IN-USE =
3684
DSNB432I  -CEA1 SIMULATED BUFFER POOL ACTIVITY -
             AVOIDABLE READ I/O -
              SYNC  READ I/O (R)  =365071
              SYNC  READ I/O (S)  =5983
              ASYNC READ I/O      =21911
              SYNC  GBP READS (R) =89742
              SYNC  GBP READS (S) =184
              ASYNC GBP READS     =279
          PAGES MOVED INTO SIMULATED BUFFER POOL =13610872
              TOTAL AVOIDABLE SYNC I/O DELAY =158014 MS
```

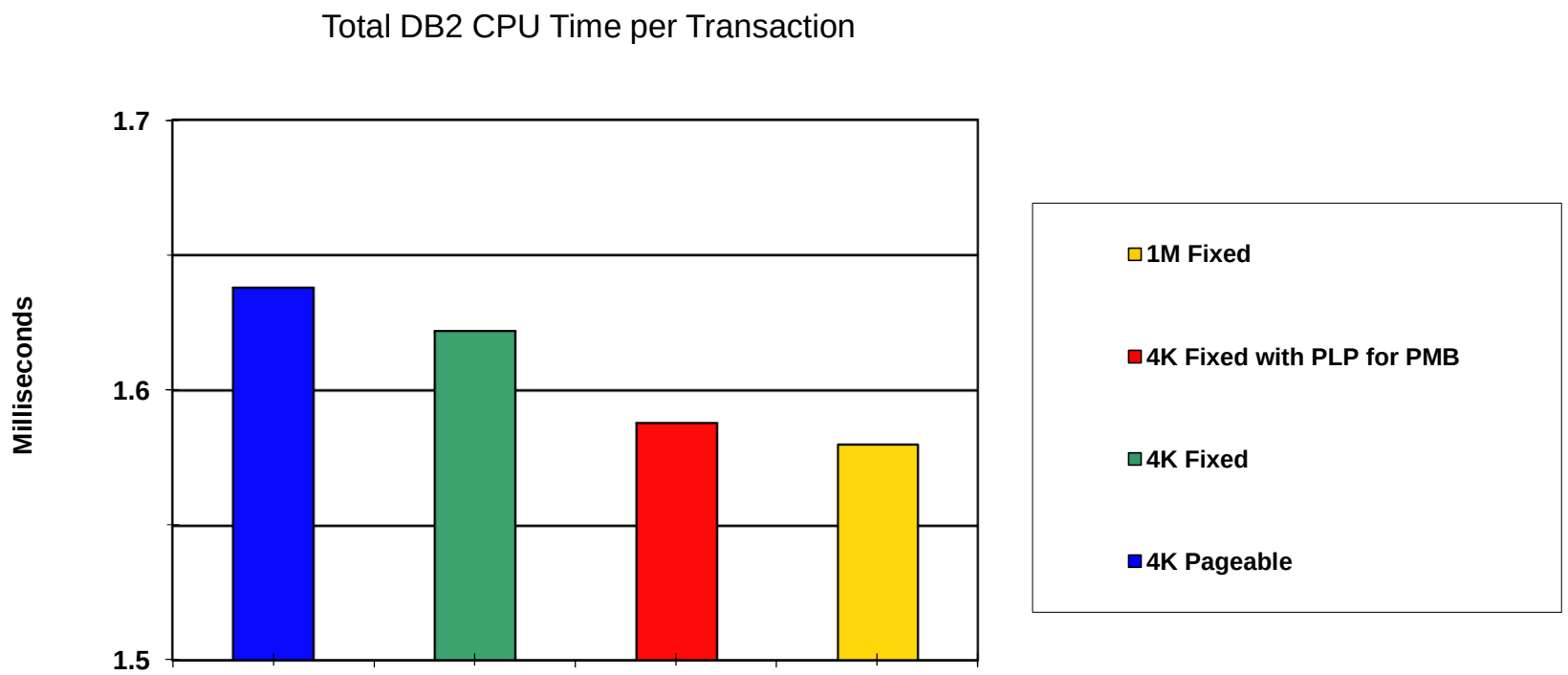
Potential DB2 Benefit from Larger Memory

- **DB2 local and group buffer pools**
  - Reduction of elapsed time and CPU time by avoiding I/Os
  - PGSTEAL(NONE) in DB2 10 = In memory data base
  - CPU reduction from PGFIX=YES and large page frames
- **Thread reuse with IMS or CICS applications**
  - Reduction of CPU time by avoiding thread allocation and deallocation
- **Thread reuse and RELEASE(DEALLOCATE)**
  - Reduction of CPU time by avoiding package allocation and parent locks
  - DDF High performance DBATs support with DB2 10
  - Ability to break-in persistent thread with DB2 11
- **Global dynamic statement cache**
  - EDMSTMTC up to 4G with DB2 11, default 110MB
  - Reduction of CPU time by avoiding full prepare
- **Local statement cache**
  - MAXKEEPD up to 200K statements with DB2 11, default 5000
  - Reduction of CPU time by avoiding short prepare
- **In-memory data cache for sparse index**
  - MXDTCACH up to 512MB per thread, default 20MB
  - Reduction of CPU and elapsed time with potentially better access path selection with DB2 11
- **RID Pool (MAXRBLK)**
  - Lower CPU time if RID lists don't spill into work files

Large Page Frames - IBM Measurements

All of buffer pools are backed by real storage

- zEC12 16 CPs, 5000-6000 tps (mid to complex transactions) with 70 GB local buffer pools
  - 120GB real storage with 75GB LFAREA configured for 1MB measurements
- 1MB frames with PGFIX=YES (long term page fix) is the best performer
- 4KB frames using PGFIX=YES and zEC12 Flash Express exploitation (1MB Pageable PMBs) is good alternative
  - Note : 70GB buffer pools are used, 8-10 sync I/O, 370 getpages per transaction



1MB Frames and LFAREA(IEASYSxx)

- Meant for memory rich environment
- Identify the candidate buffer pools
  - DB2 10 : Existing PGFIX = YES pools
  - DB2 10 and 11 : Buffer pools with high getpage intensity
- Estimating LFAREA

  LFAREA = 1.04 x (sum of VPSIZE from candidate buffer pools) + 20MB

  (+ OUTBUFF size for DB2 11)
  - 20MB to accommodate z/OS usage
  - With Java use, additional java heap size needs to be considered
- Related z/OS APARs
  - APAR OA34024 – Documentation on how to select the right LFAREA size
    - Using the DISPLAY VIRTSTOR,LFAREA system command
  - APAR OA41968 – Fixed 1M pages were not be used to satisfy 4K page requests
    - Added the support for INCLUDE1MAFC in the LFAREA parameter (cause the system to take the available fixed 1M frames into account when making paging decisions)

Pageable 1MB page frames

- May be used with PGFIX(NO) buffer pools starting with DB2 10
- 1M size pageable large frames arrived in DB2 10 and used for buffer pool control blocks, not for the buffers
- Requires Flash Express
  - Flash Express is good for paging, but normally your buffer pools should never be paged out
- Not good to use if the system is paging unless the entire buffer pool becomes dormant
  - z/OS converts 1M size large Frames into 256 x 4K size small frames
  - DB2 is not allowed to use these frames since DB2 is non swappable and uses preferred storage
  - z/OS expects to recombine the small frames back into large frames later

2GB frames

- 2GB frame improvements were made in z13, but measurements not yet complete to quantify the benefit
- Watch this space

In-Memory Database

- In-memory DBMS have existed for over a decade
- Concepts apply for both row and column store formats
- DB2 for z/OS incorporates extensive in-memory technology and operates almost exclusively on in-memory data

  Keeps frequently accessed data in memory (buffer pools)
  - Avoids disk I/O: > 90% of data accessed in memory without I/O
  - Prefetch mechanisms avoid I/O waits
  - Option to pin a table in memory

  Writes all data changes (INSERT, UPDATE, DELETE) to memory
  - Persistently writes log records to disk by commit time
    → Same behavior as In-Memory Databases
- PGSTEAL(NONE)
  - Buffer pool option for In-memory objects
- Extremely efficient memory usage across the cluster

PGSTEAL(NONE)

- Meant for stable objects which can be fit in buffer pools
- Benefit
  - Eliminate I/Os by keeping the objects in memory after first access
  - Reduce page stealing overhead (no maintenance for LRU chain)
  - Disable prefetch
- How it works
  - DB2 preloads the objects (TS, partition, index space) at the first access
  - If a page needs to be stolen, DB2 uses FIFO algorithm
- Recommendations
  - Use for performance sensitive frequently accessed objects without size increase (read only or in-place update)
  - CLOSE(NO)

Local Buffer Pools vs. Group Buffer Pools

- Observations:
  - Local buffer pool - read-only pages and changed pages
  - Group buffer pool with default GBPCACHE CHANGED - changed pages only
  - For most workloads, investing in local buffer pools likely shows greater benefit provided that you have enough GBP allocated
  - As local buffer pools are increased, pay special attention to directory reclaims. Increase GBP size to limit excessive directory reclaims.
  - CPU benefit varies depending on the workload and stress level
- GBPCACHE (ALL)
  - Cache the read and changed pages
  - Less CPU saving if found in GBP instead of LBP
  - Preliminary measurements are looking promising but will require thorough evaluation
  - A large number of DB2 members will tend to favor more investment in GBP at the expense of LBP

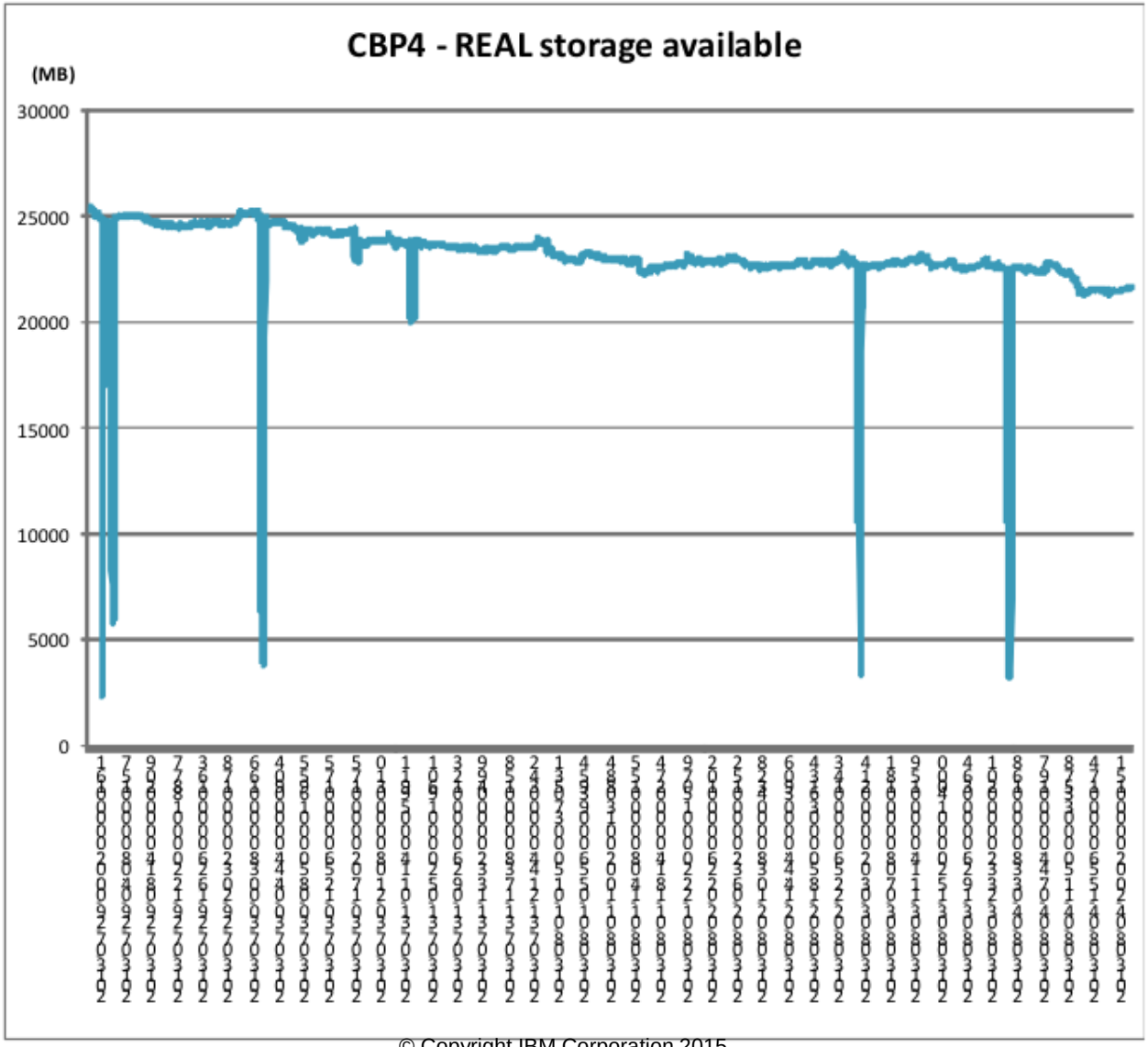
Paging

- DB2 paging is bad
  - Paging adds to direct CPU cost and hurts response time
  - Held locks while paging magnifies the problem, causing sympathy sickness on other LPARs
  - Flash Express mitigates the problem, but does not solve it
- DFSORT may consume all available memory, putting you at risk for paging if workload spike occurs while DFSORT is running
- DB2 dumps will take longer if z/OS has to read pages from AUX storage, impacting the availability of DB2
  - Held locks magnifies the problem, causing sympathy sickness on other LPARs
- Do not oversize your buffer pools. If you regularly have paging, reduce the size of your buffer pools (and other DB2 storage pools)
  - Using PGFIX(YES) and large frames saves CPU time, but be careful that you do this in a way that does not introduce paging

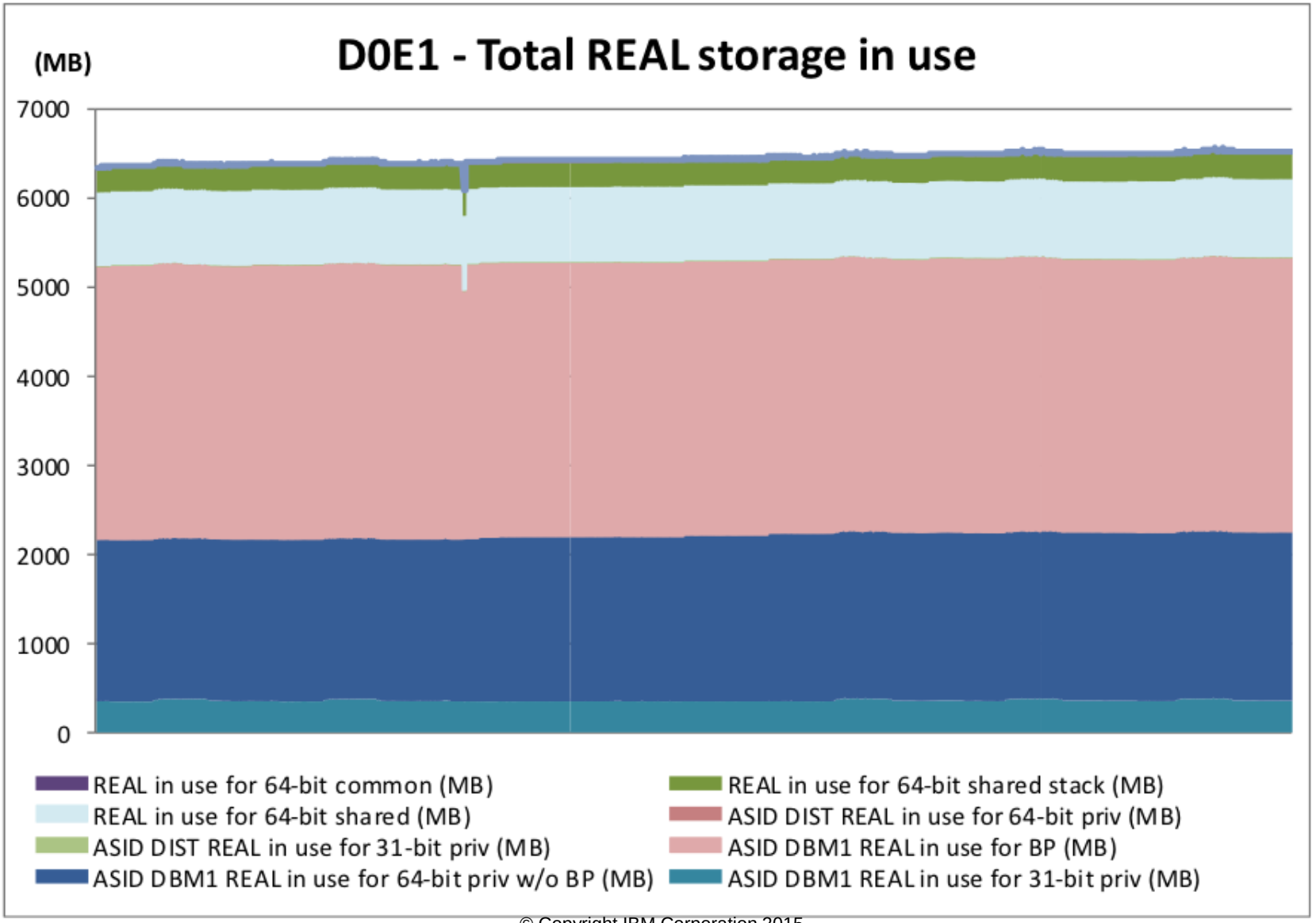
Batch overnight processing

- “I have a large LPAR (128G) and my DB2 (6G) got paged out …”
- Why is that?
  - Shift in workload with REAL frames stolen by overnight batch processing
    - Poor response times in the first few minutes of the online day
    - A lot of rapid paging going on
    - Huge increase in number of threads causing application scaling issues (lock contention, global contention)
  - REAL frames stolen by DB2 utilities
    - REORG uses REAL storage for in memory sort e.g., 64G
    - DFSORT defaults
      - EXPMAX=MAX <<<<<< Make maximum use of storage
      - EXPOLD=MAX <<<<<< Allow paging of old frames
      - EXPRES=0 <<<<<< Reserved for new work

Real storage usage and DFSORT settings …



Monitoring REAL/AUX storage usage – Sample graph #1



DB2 pages paged out and a dump happens?

- What is my exposure?
  - Increased MASTER CPU time, z/OS tries to steal frames to meet the excessive demand caused by the dump
  - Elongated dump times
- Auxiliary storage number > 0
  - In theory each page could have to be paged in
    - z/OS can have the page in both places, REAL and AUX
    - If total size across all bufferpools is more than 800MB then the bufferpools are not dumped
  - No prefetch on AUX storage, so all synchronous I/O
  - Worst case is the number of pages * page-in I/O time
    - For example 2GB of 4K pages * 3ms = 524288 * 0.003 = 26 mins
  - Guinness world record for a dump 37 mins
  - Full Sysplex hang resulted
- At 250 MB/sec, you can write 500MB in 33 seconds
- Writing dumps to Flash Express helps

SYSPLEX sympathy sickness

- Slowdown and no apparent reason why
- Excessive dump time caused by paging on the LPAR may cause massive sympathy sickness slowdowns
- DB2 taking the dump may have TCBs non-dispatchable
- P-Lock negotiation affected
- Locks not released in a timely manner
- How can a member slow down when there is plenty of CPU/Storage on the LPAR
  - Maybe the owner of a P-lock is being dumped or is paging a lot

MAXSPACE

- Make sure MAXSPACE is set properly and defensively
  - Represents the total amount of storage for captured dumps for the entire LPAR
  - MAXSPACE value should not be set so high that paging can occur causing massive issues to the LPAR
  - If multiple DB2s on same LPAR can wildcard to the same dump, then MAXSPACE needs to be set appropriately
  - MAXSPACE=16G is a good start to cope with more than 90% of all cases
    - But there are z/OS defects around which are inflating DUMP size
    - Fixing z/OS APARs available to handle and minimise DUMP size
      - OA39596, OA40856 and OA40015
  - MAXSPACE requirement should be
    - (DBM1 – Buffer pools) + Shared memory + DIST + MSTR + IRLM + COMMON + ECSA
  - Work is underway to get the exact formula based on all the new IFCID 225 fields
    - Once the formula is properly tested, will be posted on the various websites and Info APARs

WLM STORAGE CRITICAL

- Specify z/OS WLM STORAGE CRITICAL for DB2 system address spaces
  - To safeguard the rest of DB2
  - Tells WLM to not page these address spaces
  - Keeps the thread control blocks, EDM and other needed parts of DB2 in REAL
  - Prevents the performance problem as the Online day starts and DB2 has to be rapidly paged back in

DISCARD processing

- To use KEEPREAL=YES or NO, that is the question
- When DB2 wants to free some 64-bit storage it DISCARDs the storage
  - KEEPREAL=YES tells RSM not to free the memory until available frame queue becomes low
    - DB2 statistics on memory usage treat this memory as in use (not really accurate)
  - KEEPREAL=NO tells RSM to free the memory immediately
    - A first-reference page fault will occur if and when DB2 reuses the virtual storage address
- SPIN locks used by z/OS RSM can cause performance issues when DISCARD processing is running on many CPUs at the same time
  - Possible LPAR outage in severe cases
  - The danger of SPIN locks is greatest on large LPARs

REAL_STORAGE_MANAGEMENT (After PM99575)

- When paging occurs, DB2 will use KEEPREAL=NO
- REAL_STORAGE_MANAGEMENT=OFF means KEEPREAL=YES
  - Frames will be reused when paging occurs
  - DB2 statistics are not “accurate” because frames will not be stolen back to reduce the count until paging occurs
  - Statistics will be a high water mark on most systems
  - Statistics will be fairly accurate on systems with small amounts of paging
- REAL_STORAGE_MANAGEMENT=AUTO with no paging means KEEPREAL=YES at Thread Deallocation or 120 commits
- RSM=AUTO with paging or RSM=ON means KEEPREAL=YES at Deallocation or 30 commits. STACK also DISCARDED
- REALSTORAGE_MAX means KEEPREAL=NO at 100%

CRITICALPAGING

- If you have XCF CRITICALPAGING ENABLED, you also need to apply z/OS RSM APAR OA44913
  - Without this APAR, the KEEPREAL(YES) shared frames are not stolen to replenish frame queues when paging occurs
  - Either APAR may be applied independently without the other
- NO BENEFIT FROM PM99575 if CRITICALPAGING is enabled and z/OS APAR not applied
- Do I have it?
  - D XCF,COUPLE
- To Activate
  - Update COUPLExx with: FUNCTIONS ENABLE(CRITICALPAGING)

Real storage controls

- Make sure REALSTORAGE_MANAGEMENT=AUTO (default)
  - Particularly when significant paging is detected, “contraction mode” will be entered to help protect the system
    - “Unbacks” virtual pages so that a REAL frame or AUX slot is not consumed for this page
  - Use automation to trap the DSNV516I (start) and DSN517I (end) messages
- As DB2 approaches the REALSTORAGE_MAX threshold
  - “Contraction mode” is also entered to help protect the system
- Control use of storage by DFSORT
  - Set EXPMAX down to limit maximum DFSORT usage
  - Set EXPOLD=0 to prevent DFSORT from taking "old" frames from other workloads
  - Set EXPRES=% or n {reserve enough for MAXSPACE}
- z/OS parameter AUXMGMT=ON
  - No new dumps are allowed when AUX storage utilization reaches 50%
  - Current dump data capture stops when AUX storage utilization reaches 68%
  - Once the limit is exceeded, new dumps will not be processed until the AUX storage utilization drops below 35%

Monitoring REAL/AUX storage usage – Mapping for reference

IFCID FIELD	OMPE FIELD	OMPE PDB COLUMN NAME	MEMU2 Description
QW0225RL	QW0225RL	REAL_STORAGE_FRAME	DBM1 REAL in use for 31 and 64-bit priv (MB)
QW0225AX	QW0225AX	AUX_STORAGE_SLOT	DBM1 AUX in use for 31 and 64-bit priv (MB)
QW0225HVPagesInReal	SW225VPR	A2GB_REAL_FRAME	DBM1 REAL in use for 64-bit priv (MB)
QW0225HVAuxSlots	SW225VAS	A2GB_AUX_SLOT	DBM1 AUX in use for 64-bit priv (MB)
QW0225PriStg_Real	SW225PSR	A2GB_REAL_FRAME_TS	DBM1 REAL in use for 64-bit priv w/o BP (MB)
QW0225PriStg_Aux	SW225PSA	A2GB_AUX_SLOT_TS	DBM1 AUX in use for 64-bit priv w/o BP (MB)
QW0225RL	QW0225RL	DIST_REAL_FRAME	DIST REAL in use for 31 and 64-bit priv (MB)
QW0225AX	QW0225AX	DIST_AUX_SLOT	DIST AUX in use for 31 and 64-bit priv (MB)
QW0225HVPagesInReal	SW225VPR	A2GB_DIST_REAL_FRM	DIST REAL in use for 64-bit priv (MB)
QW0225HVAuxSlots	SW225VAS	A2GB_DIST_AUX_SLOT	DIST AUX in use for 64-bit priv (MB)
QW0225ShrStg_Real	SW225SSR	A2GB_SHR_REALF_TS	REAL in use for 64-bit shared (MB)
QW0225ShrStg_Aux	SW225SSA	A2GB_SHR_AUXS_TS	AUX in use for 64-bit shared (MB)
QW0225ShrStkStg_Real	SW225KSR	A2GB_SHR_REALF_STK	REAL in use for 64-bit shared stack (MB)
QW0225ShrStkStg_Aux	SW225KSA	A2GB_SHR_AUXS_STK	AUX in use for 64-bit shared stack (MB)
QW0225ComStg_Real	SW225CSR	A2GB_COMMON_REALF	REAL in use for 64-bit common (MB)
QW0225ComStg_Aux	SW225CSA	A2GB_COMMON_AUXS	AUX in use for 64-bit common (MB)
QW0225_REALAVAIL	S225RLAV	QW0225_REALAVAIL	REALAVAIL (MB) (S)

Note: All REAL/AUX storage fields in IFCID 225 and OMPE performance database are expressed in 4KB frames or slots – they should be converted to MB (conversion is already done in MEMU2)

Summary

- Keep the LPAR well provisioned with REAL storage
  - Avoid Paging to AUX or Flash Express
  - Use buffer pool simulation to predict benefits of a larger buffer pool
- Do not oversize your buffer pools and use PGFIX(YES)
- Apply the DB2 APARs
- Apply the z/OS APAR if CRITICALPAGING is enabled
- Use REAL_STORAGE_MANAGEMENT=AUTO
- Use 1MB page frames if you have a memory rich environment
  - Do not over commit the LFAREA if the system may page and the large frames may be broken down and put back together again
- Watch out for MAXSPACE and large dump sizes that may cause the system to page
- Don’t let DFSORT consume all of your available memory
- Use ‘common currency’ for monitoring REAL and AUX usage

Continue growing your IBM skills

ibm.com/training

provides a comprehensive portfolio of skills and career accelerators that are designed to meet all your training needs.

If you can't find the **training that is right for you** with our Global Training Providers, we can help.

Contact IBM Training at dpmc@us.ibm.com

Global Skills Initiative