

Hot Topics for DB2 10 for z/OS

John J. Campbell

DB2 for z/OS Development

Michael Dewert

DB2 for z/OS Development

Session Code: A05

Tuesday 15th October 2013 10:15am – am Platform: DB2 for z/OS





Abstract

- This session will introduce and discuss the most common issues and topics around real world customer usage of DB2 10 for z/OS. Specifically this will include:
 - Real storage control
 - Online release migration in 24*7 environment
 - Steps to investigate CPU performance
 - OPTIMIZE FOR 1 ROW
 - OPTIOWGT system parameter
 - zIIP capacity
 - CHAR, VARCHAR and CAST



Objectives

- Understand real storage monitoring and control
- Introduce and discuss performance opportunities and surprises
- How to plan for and execute online release migration
- Introduce and discuss release incompatibilities



Agenda

- Real storage control
- IFCID 225 reference
- Bufferpool - large size page frames, long term page fix
- ECSA usage
- Release migration in 24*7 environment
- Steps to investigate CPU performance
- OPTIMIZE FOR 1 ROW
- OPTIOWGT
- zIIP Capacity
- CHAR, VARCHAR, CAST
- Preventative maintenance
- Excessive SPT01/DBD01 growth



Real storage control

- Paging rate is a critical performance measure for any z/OS system
 - When shortage of REAL frames, frames are moved to AUX (DASD)
 - Having DB2 paged out is a not good thing for performance
 - Paging should be minimised
- Page fixing bufferpools is a good idea for performance
 - Avoid page fix and page free for high activity bufferpools (heavy I/O)
 - Page fixed 1M real storage page frames reduces TLB misses (saves CPU)
- But if insufficient REAL storage provisioned for the LPAR
 - LPAR begins to page and DB2 is a candidate for page stealing
 - Thread and EDM Pool storage is paged out
 - Performance problems as data is rapidly paged back in



Real storage control ...

- “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
 - Poor response 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
 - Dump capture
 - TCBs non-dispatchable
 - P-Lock negotiation delayed affecting other members
 - Locks not released in a timely manner
 - Excessive dump time caused by paging on the LPAR may cause massive sysplex-wide sympathy sickness slowdowns z



Real storage control ...

- 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 but if DUMPs are needed then it can cause a small issue to become a massive SYSPLEX failure
 - Cheapest because MLC and other charges do not factor in the amount of REAL storage
 - Vendors do not charge by the amount of REAL on the machine
- Specify z/OS WLM STORAGE CRITICAL for DB2 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



Real storage control ...

- Make sure MAXSPACE is set properly and defensively
 - 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 MVS 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 – Bufferpools) + 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



Real storage control ...

- Make sure REALSTORAGE_MANAGEMENT=AUTO (default)
 - 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 EXPOLD=0 to prevent DFSORT from taking "old" frames from other workloads
 - Set EXPRES=% {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%



IFCID 225 reference

Real/Aux Storage for DBM1 or DIST = **QW0225RL/QW0225AX**

Real/Aux 64-bit for DBM1 = **QW0225HVPagesInReal/QW0225AuxSlots**

Real/Aux 64-bit for DBM1 excluding bufferpools = **QW0225PriStg_Real/QW0225PriStg_Aux**

Real for bufferpools and associated objects = **QW0225HVPagesInReal - QW0225PriStg_Real**

Shared thread and system Real/Aux = **QW0225ShrStg_Real/QW0225ShrStg_Aux**

Shared stack Real/Aux = **QW0225ShrStkStg_Real/QW0225ShrStkStg_Aux**

64-bit Common Real/Aux = **QW0225ComStg_Real/QW0225ComStg_Aux**

Total 31-bit ECSA storage = **QW0225GC + QW0225FC + QW0225VC**



Bufferpool - large size page frames, long term page fix

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 Backed by real or LFAREA	CPU reduction from TLB hit
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



ECSA usage

- Customers claiming
 - Drop in free ECSA storage which could cause LPAR failure and critical service impact
 - Orphaned ECSA storage in DB2 on behalf of CICS regions that have “owner gone (OG)” or been recycled is a major contributing factor
- Storage marked “owner gone” is a well known reporting problem
 - If CSA pool is expanded for say an IRLM RLPL when a batch job is running then z/OS reports the home address space that did the getmain
 - The block is subsequently returned to the pool and the batch job ends
 - “Owner gone” is not orphaned storage and it should not be cleaned up as it may bring DB2 crashing down very quickly
- In DB2 10, DB2 records the storage against our own ASIDs
 - Reporting problem is now fixed
 - But there is a tiny increase in the ECSA storage requirement in DB2 10
- If there is an ECSA leak then track it and take two dumps
 - DB2 has SM=1 global report in the dump
 - Shows how much ECSA is used and for what



Online migration in 24*7 environment

- Technically possible to run DSNTIJTC and DSNTIJEN alongside well-behaved online workloads
 - Jobs use SQL DDL with frequent commit and REORG SHRLEVEL(REFERENCE)
 - Designed to fail gracefully leaving DB2 catalog fully operational
 - After problem determination is complete, the respective job can be corrected and resubmitted
 - The respective job will restart from where it left off



Online migration in 24*7 environment ...

- But some 'rules of the game' and you must be prepared to play
 - DSNTIJTC and DSNTIJEN jobs should be scheduled during a relative quiet period
 - If non data sharing
 - Must stop all application workload when DSNTIJTC job is running
 - If data sharing
 - Must route work away from the DB2 member where DSNTIJTC job is running
 - Must temporarily change workload balancing and sysplex routing scheme
 - Should synthetically stop all of the following workload types from running
 - SQL DDL, Grants & Revokes, BIND/REBIND, utilities, monitors
 - All essential business critical workloads that are running should commit frequently
 - Must be prepared to watch and intervene if needed
 - Strong recommendation to perform Pre-Migration Catalog Migration Testing
 - Must be prepared for DSNTIJTC and/or DSNTIJEN jobs to possibly fail or for some business transactions to fail



Online migration in 24*7 environment ...

- Some critical maintenance
 - APAR PM66287
 - Ensure RTS and AUTOBIND do not interfere with Online CATMAINT
 - APAR PM62572
 - Undetected lock contention failure during the switch phase of the ENFM REORG step
 - APAR PM58575
 - Autobind triggers deadlock with RTS
 - APAR PM81189
 - Associated with the CGTT set of problems to be corrected
- If not prepared to play by the 'rules of the game' then take the outage
 - Quiesce all applications
 - Run DSNTIJTC or DSNTIJEN job with DB2 started ACCESS(MAINT)



Steps to investigate CPU performance

- Comparing CPU performance on V10 relative to V8 or V9
 - More difficult to do in real customer production environment
 - Uncertainty caused by application changes
 - Fluctuation in the daily application profile especially batch flow
 - Must try to normalise things out to ensure workloads are broadly comparable
 - Broadly similar in terms of SQL and getpage profile
 - Usually have to exclude the batch flow
 - Factor out extreme variation
 - Need to look at multiple data points



Steps to investigate CPU performance ...

- Check that you have the same pattern across releases from a DB2 perspective based on combined view of DB2 Statistics and Accounting Traces
- Validate that there have been no access path regression after migration or from application changes going on at the same time as the migration
- Use as a starting point look at
 - Statistics Trace
 - MSTR TCB & SRB, DBM1 TCB, SRB & IIP SRB, IRLM TCB & SRB CPU times
 - Split of CP vs. zIIP for DBM1 is likely to be very different between V9 and V10
 - Accounting
 - For each CONNTYPE
 - Class 2 CPU times on CP and zIIP, numbers of occurrences and commits/rollbacks
 - Workload indicators:
 - DML (split by type: select, insert, update, fetch, etc...),
 - Commits, rollbacks, getpages, buffer update
 - Read and write activity (#IOs. #pages)



Steps to investigate CPU performance ...

- A challenge to get an 'apple-to-apple' comparison in a real production environment
- Best chance is to find a period of time with limited batch activity, and to look at the same period over several days in V8/V9 and several days running on V10
- Make sure that the CPU numbers are normalized across those intervals i.e., use CPU milliseconds per commit
- Easy to combine statistics and accounting by stacking the various components of CPU resource consumption:
 - MSTR TCB / (commits + rollbacks)
 - MSTR SRB / (commits + rollbacks)
 - DBM1 TCB / (commits + rollbacks)
 - DBM1 SRB / (commits + rollbacks)
 - DBM1 IIP SRB / (commits + rollbacks)
 - IRLM TCB / (commits + rollbacks)
 - IRLM SRB / (commits + rollbacks)
 - Average Class 2 CP CPU * occurrences / (commits + rollbacks)
 - Average Class 2 SE CPU * occurrences / (commits + rollbacks)



Steps to investigate CPU performance ...

- Need to check the workload indicators for the chosen periods
- Similarities between data points for a given version, but big variations between V8/V9 and V10
 - Sign that something has changed from an application or access path perspective
 - More granular analysis of accounting data will be required to pin point the specific plan/package



OPTIMIZE FOR 1 ROW

- Potential for access path regression when using OPTIMIZE FOR 1 ROW
 - Used by customers as a hint to discourage use of sort or list prefetch
 - Sometimes applied as an installation SQL coding standard
 - DB2 access path selection has always been cost based
 - V10 'hammer' change
 - Excludes the 'sort' access plan candidates
 - Remaining 'sort avoidance' access plans compete on cost – lowest cost wins
 - If no 'sort avoidance' access plans, then 'sort' access plans remain and compete on cost
 - Seeing increasing evidence of access path regression when multiple candidate indexes available e.g.,
 - DB2 using alternate index with lower MATCHCOLS value because there is no sort



OPTIMIZE FOR 1 ROW ...

- Solutions
 - Change application to code OPTIMIZE FOR 2 ROWS
 - Alter an existing index or create a new index that would support both sort avoidance and index matching (if predicates allow)
 - Set new system parameter OPT1ROWBLOCKSORT to control behavior of OPTIMIZE FOR 1 ROW
 - Introduced with APAR PM56845
 - Set OPT1ROWBLOCKSORT=DISABLE (default) and avoid performance regression



OPTIOWGT

- V8 default for OPTIOWGT is DISABLE
- ENABLE is the DB2 default in V9 and V10, and proven best practice to stabilise access path selection
 - Adjusts downward the weight of the I/O cost in access path selection
 - Particularly important when running on fast processors (z10, z196, ...)
- Installations should have changed OPTIOWGT to ENABLE as part of the V10 migration
 - Common mistake for customer performing skip release migration from V8 to V10
- Recommend that all installations should set OPTIOWGT=ENABLE
- PTF for APAR PM70046 will actually force OPTIOWGT=ENABLE



zIIP capacity

- In DB2 10, prefetch and deferred writes engines are now eligible for zIIP offload
 - These DB2 tasks must be dispatched very quickly
 - Any delays could result in
 - Significant elapsed time degradation for some batch and DB2 utility jobs
 - Very high count for 'prefetch engine not available' in the DB2 Statistics Trace
- Many installations running with IIPHONORPRIORITY=YES, HIPERDISPATCH=YES and ZIIPAWMT=3200 settings in IEAOPTxx parmlib member
 - zIIP processors can get help from standard processors (GCP)
 - zIIP needs to be running for 3.2 msec before it checks to see if it should ask for help from GCPs ('alternate wait management')
 - Many requests can be flowing through the zIIP during this time period. But if the zIIP has been running for the length of time specified by ZIIPAWMT, the queue of work is still growing, and all the zIIPs are busy, then the zIIP signals a GCP for help to process the work



zIIP capacity ...

- With the above default settings and if the zIIP capacity is under-configured
 - DB2 prefetch engines can end up queuing for up to 3.2 msec for a zIIP before they are dispatched on a GCP
 - Of course, this could be much worse if the zIIP processors were not allowed to ask GCPs for help (IIPHONORPRIORITY=NO)
- Tuning knobs available which will impact zIIP offload for all workload
 - Disable 'alternate wait management' by setting HIPERDISPATCH=NO and reducing ZIIPAWMT value to get the zIIP to ask for help from GCP sooner
 - However using GCPs to do zIIP-eligible work will negatively impact TCO
- Correct technical solution is to add more zIIP capacity
 - zIIPs are assist processors and not intended to be run as hard as GCPs
 - zIIPs usage should be in the 30-50% CPU busy range on average (peaks higher)



CHAR(dec), VARCHAR(dec), CAST(dec as CHAR), CAST(dec as VARCHAR)

- For example

```
SELECT COL1
       , CHAR ( COL1 )
       , HEX ( CHAR ( COL1 ) )
       , SUBSTR ( CHAR ( COL1 ) , 2 , 4 ) APPL_SUBSTR
FROM DB2_TABLE ;
```

COL1	CHAR_V9	HEX_CHAR_V9	APPL_SUBSTR	COL1	CHAR_V10	HEX_CHAR_V10	APPL_SUBSTR
9	0009	40F0F0F0F96B	0009	9	9	F94040404040	9
99	0099	40F0F0F9F96B	0099	99	99	F9F940404040	99
999	0999	40F0F9F9F96B	0999	999	999	F9F9F9404040	999
9999	9999	40F9F9F9F96B	9999	9999	9999	F9F9F9F94040	9999

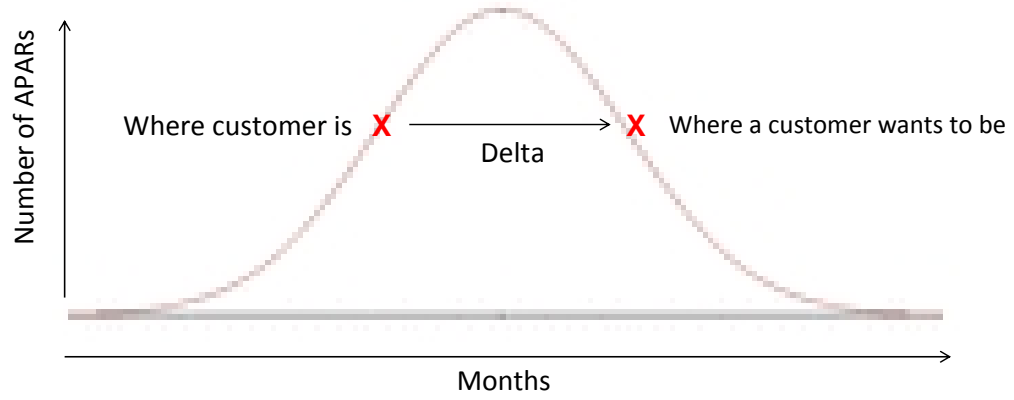


CHAR(dec), VARCHAR(dec), CAST(dec as CHAR), CAST(dec as VARCHAR) ...

- APAR PM29124 restores compatible behavior for CHAR(dec)
 - Set new system parameter BIF_COMPATIBILITY=V9 (default)
 - Get the old behavior for the CHAR(dec) function
 - But new V10 behavior for VARCHAR/CAST
 - New IFCID 366 to identify potential applications exposed
- APAR PM66095 and PM65722 introduces controls for desired behavior
 - BIF_COMPATIBILITY zparm settings
 - 'V9': will continue to provide the old behavior for the CHAR(dec)
 - 'V9_DECIMAL_VARCHAR': will get the old behavior for VARCHAR and CAST also
 - 'CURRENT': provides the new behavior for all the built in functions
 - Use 'SYSCOMPAT_V9' vs. 'SYSCURRENT' at the beginning of BIND option PATH or in setting SQL CURRENT PATH special register
 - Only affects CHAR and VARCHAR behavior
- APAR PM70455 retrofits IFCID 366 to Version 9



Preventative maintenance – ‘Bell Curve’





Preventative maintenance – some observations

- Preventative maintenance strategy for any given customer needs to adapt to changing circumstances
 - Needs to be dynamic and flexible
 - Consider being more aggressive in maintenance strategy
 - If the number of problems experienced are fixed by PTFs that are not applied
 - If an early adopter of a new version or some new function
 - Consider being less aggressive and more conservative if continually hitting PEs
- Some customers are traditionally conservative and on the trailing edge on current version and preventative maintenance
 - Version upgrade triggered based on end of currency date of prior release
 - Cycle of preventative maintenance upgrades is also based on being on the trailing edge of current version
 - Reliance on PEs being found by other customers
 - No pro-active checking for HIPERs and PEs
 - Often resistant to and inflexible on
 - Application of critical HIPERs during and after roll-out period
 - Investigation of PEs (always back off which takes the base back in time)



Preventative maintenance – some observations

- Some customers are traditionally conservative and trailing edge on releases and preventative maintenance ...
 - Current process model may have served them well in terms of maintaining system availability and stability
 - Some of these same customers made the decision to migrate early to DB2
 - The decision in some cases was not a balanced decision
 - Predicated on
 - Accruing CPU cost savings from DB2 10 earlier
 - Cost savings from performing only one version rather than two
 - No additional risk mitigation actions taken as a result of moving up the adoption curve of new version
 - Continued with the old behaviour model on applying preventative service
 - No adequate plans for more frequent application of preventative service
 - No enhanced pre-production QA testing
 - Without regular preventative service upgrades the base could be many months old
 - Current run rate is ~30 HIPERs per month
 - HOLD actions
 - Infrequent preventative maintenance piles up the research and actions to be carried out



Preventative maintenance - recommendations

- As a priority implement a continuous scheduled program for applying regular DB2 preventative service using CST/RSU method
 - Paramount importance to maintain system availability and stability
 - Plan on 3-4 major preventative maintenance upgrades a year over next 12+ months to catch up and continue until world wide customer production adoption of DB2 10 becomes trailing edge
 - Pull and review Enhanced HOLDDATA on at least a weekly basis
 - Pro-active checking of all HIPERs and PEs looking for critical problems e.g.,
 - Data loss, NCORROUT, overlays, crashes, bad restart/recovery, etc
 - Introduce management process and procedure for expediting the apply of the most critical HIPERs after 1-2 weeks in Test
 - During the rollout of new preventative service package on way to Production
 - Production thereafter
 - Enhance QA testing to provide better coverage and 'keep fires away' from production
 - Application of preventative maintenance can become less aggressive as DB2 10 becomes trailing edge
 - Perform at least 2 major preventative maintenance upgrades per year



APARs for excessive SPT01/DBD01 growth – Base and LOB tablespaces

- PM64226
 - LOB table space enhancement
- PM74659
 - SPT01 space reuse enhancement
- PM75921
 - Candidate page lookup enhancement
 - More exhaustively search for an actual position for insertion within the table
- PM79266
 - Rollback fix for reuse pseudo delete space
- PM81485
 - AE fix apar of PM7465
 - More enforcement in roll back process to prevent other INSERT UR to reuse the free space from rollback of insert
- PM77611
 - Basing chain of PM79266



Summary

- Understand real storage monitoring and control
- Introduce and discuss performance opportunities and surprises
- How to plan for and execute online release migration
- Introduce and discuss release incompatibilities

25 IDUG
Leading the DB2 User
Community since 1988

IDUG DB2 Tech Conference
Barcelona, Spain - October 2013

Questions



33



**THANK
YOU**

John J. Campbell

DB2 for z/OS Development

campbelj@uk.ibm.com

Michael Dewert

DB2 for z/OS Development

dewert@de.ibm.com

Session A05

Hot Topics for DB2 10 for z/OS

