



## DB2 for z/OS May 24 2012 Paris

# DB2 10 Migrations Experiences

Gareth Jones

DB2 for z/OS Development

Email: [jonesgth@uk.ibm.com](mailto:jonesgth@uk.ibm.com)

## Objectives

- Share lessons learned, surprises, pitfalls
- Provide hints and tips
- Address some myths
- Provide additional planning information
- Provide usage guidelines and positioning on new enhancements

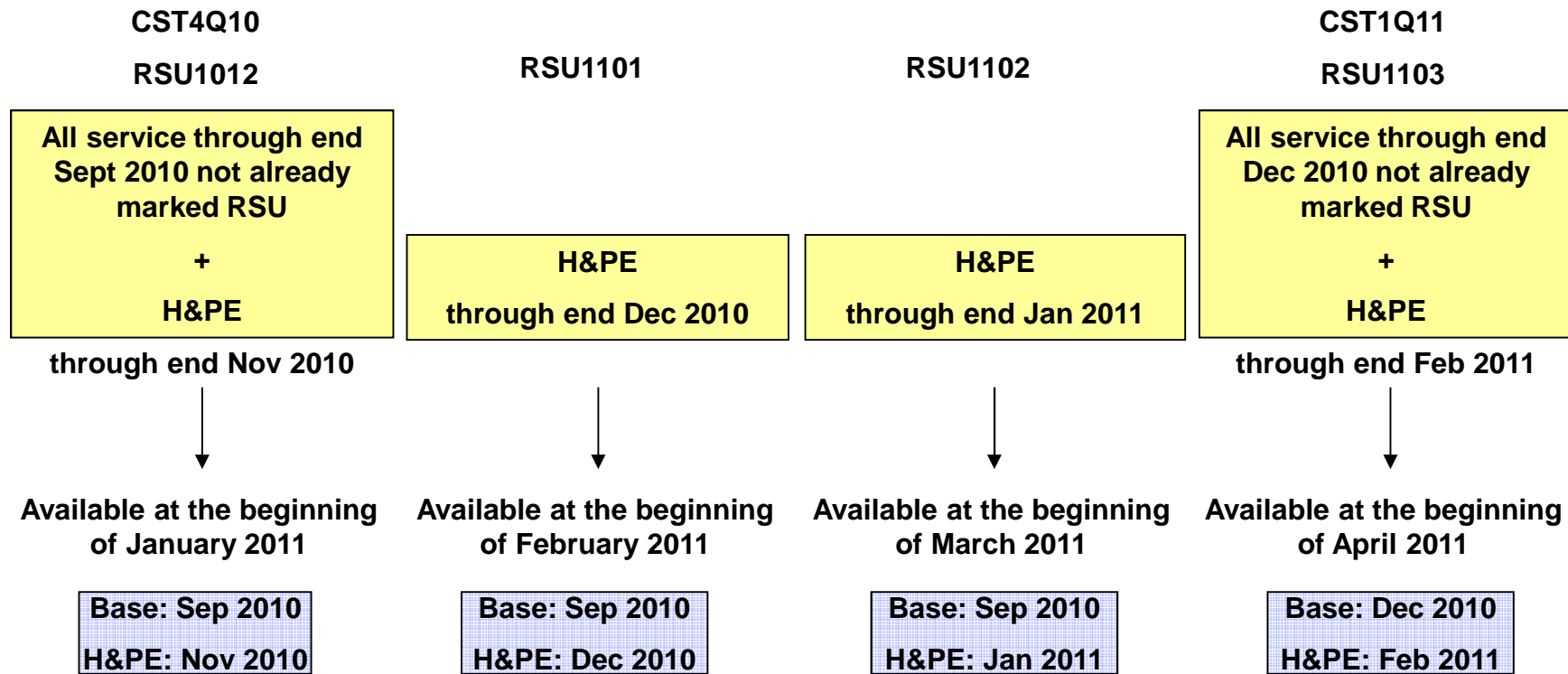
## Agenda

- Keys to customer migration success
- Performance and Scalability
- BIND, REBIND and EXPLAIN
- Availability
- Migration Planning
- Removal of DDF Private Protocol
- Security Considerations when removing DDF Private Protocol
- Other
- Summary

## Keys to customer migration success

1. Plan for continual application of preventative service
  - Need to stay more current on HIPERs at this stage in the release take up cycle
  - Apply preventative service every 3 months
    - Two “major” and two “minor” releases
    - Refresh of the base every 6 months (“major”)
    - Each base should be based on latest quarterly RSU as opposed use of PUT
    - In addition, two ‘minor’ packages covering HIPERs and PEs in between time
  - Augment by exploiting Enhanced HOLDDATA on a weekly basis before production cutover and continue thereafter
    - Identify and pull all applicable HIPERs and PE fixes
    - Expedite the most critical PTFs into production
2. Perform application regression and stress testing is the best way to keep ‘fires’ away from production
3. Build a realistic project plan
  - Avoid crash project
  - Allow contingency for ‘bumps in the road’
  - Involve applications teams early
    - Investigation of incompatible changes and fix up
    - Testing

# CST and RSU example



*H&PE = HIPER/Security/Integrity/Pervasive PTFs + PE resolution (and associated requisites and supersedes)*

## Performance and Scalability

- Many opportunities for price/performance (cost) improvements
  - Major theme of this release
  - Most welcome to our customers
- Customers intimidated by the marketing noise about improved performance
  - Expectation of their CIO
  - For some of their workloads not seeing improvements in CPU and elapsed time
  - Conversely see big improvements for certain workloads
  - Small workloads can skew expectations on savings
  - Some measurements and quotes are insanely positive
    - Should be ignored
  - How to extrapolate and estimate for production mixed workload?
    - Estimation with accuracy and high confidence not practical
    - Benchmarking effort would be required
- Very important to correctly level set customer performance expectations
- Customers should not spend any performance benefits until they see them

## Performance and Scalability ...

- Assumes no major access path regressions
- On Day 1 in production in CM without any changes (e.g., no rebind, no use of 1MB page size) there may be customers who see zero % improvement and even some will see degradation
  - Why? SPROCs disabled, puffing of run time structures for migrated packages from V8 or V9, etc
- To maximise the performance improvements must:
  - REBIND static SQL packages
  - Use PGFIX=YES bufferpools with sufficient 1MB real storage page frames to 100% fully back the requirement from PGFIX=YES bufferpools
- Seeing 0-10% improvement after REBIND and use of 1MB real storage frames
- Need to look at total CPU resource consumption picture across
  - Acctg Class 2 TCB Time (Accounting Trace)
  - DB2 System Address spaces (Statistics Trace)

## Performance and Scalability ...

- Customers should expect to see some increase in real storage consumption (10-30%)
  - Must also factor in MAXSPACE requirement for DB2 dumps (approx 16GB)
    - Avoid very long dump capture times and bad system performance
    - Critical for V10 serviceability



## Performance and Scalability ...

- The 0-10% CPU reduction is based on the DB2 portion of a given application workload
- Customer value driven on how sub-capacity workload licensing works
  - Based on 4-hour rolling average MSU utilisation
  - Highest rolling average figure for each month used to calculate software charges for all MLC products (IBM and non-IBM)
  - Provided DB2 forms a significant component of the total MSU usage during peak period, any MSU savings will translate directly to MLC savings
  - Typically this is the online day - mid morning and mid afternoon
  - So for example - this may be driven by CICS-DB2 workload where the DB2 portion of the workload only represents 40-60% of the total path length
  - So the 0-10% may represent only 0 to 6% (i.e., needs to be discounted)
  - Investigate how much CPU is used in the 4-hour period for DB2 work (SQL)

## Performance and Scalability ...

- Opportunities for additional price/performance improvements driven by DBM1 31-bit VSCR supported by additional real storage include
  - More use of persistent threads with selective use of RELEASE(DEALLOCATE)
    - High Performance DBATs
    - CICS Protected ENTRY Threads
    - CICS Unprotected ENTRY Threads with queuing
    - Typical savings 0-10%, may be more
  - Increasing MAXKEEPD to improve Local Dynamic Statement Cache hit ratio and reduce the number of short prepares
  - Sysplex/Data sharing Group consolidation
    - So for example, 8-way to 4-way
    - Reduced cost of data sharing
- Very important to correctly level set customer performance expectations
- Customers should not spend any performance benefits until they see them

## Performance and Scalability ...

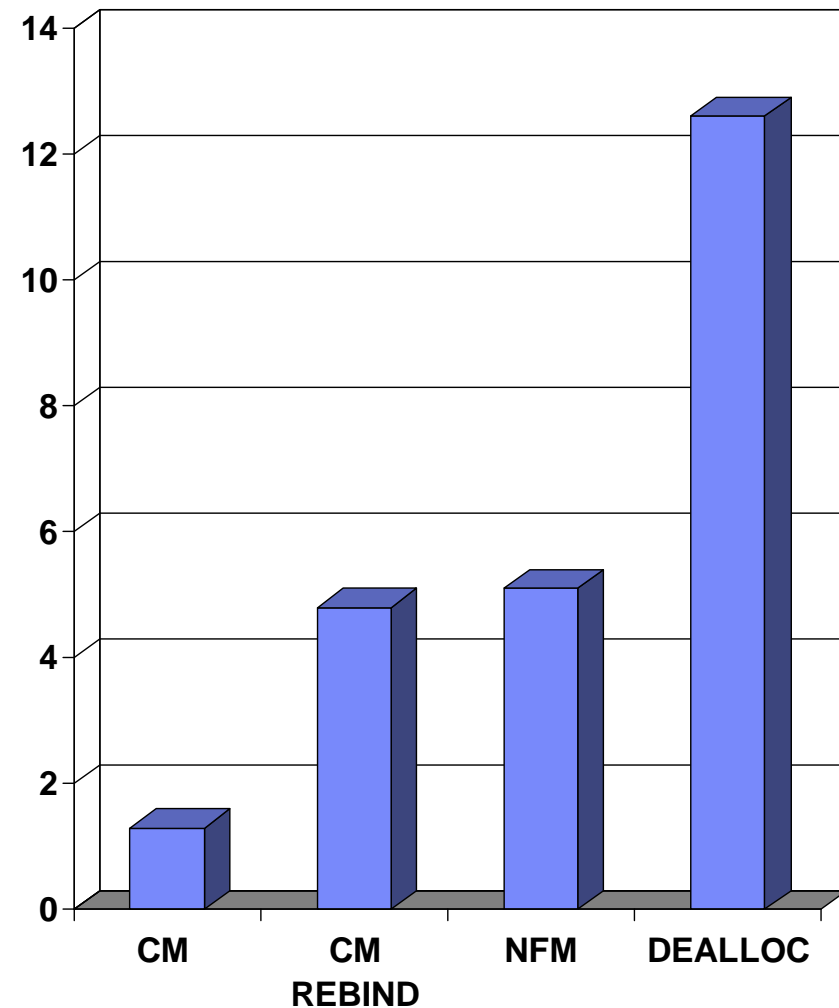
Workload	Customer Results
CICS online transactions	Approx. 7% CPU reduction in DB2 10 CM after REBIND, additional reduction when 1MB page frames are used for selective buffer pools
CICS online transactions	Approx 10% CPU reduction from DB2 9
CICS online transactions	Approx 5% CPU reduction from DB2 V8
CICS online transactions	10+% CPU increase
Distributed Concurrent Insert	50% DB2 elapsed time reduction, 15% chargeable CPU reduction after enabling high performance DBAT
Data sharing heavy concurrent insert	38% CPU reduction
Queries	Average CPU reduction 28% from V8 to DB2 10 NFM
Batch	Overall 20-25% CPU reduction after rebind packages

## Performance and Scalability ...

Workload	Customer Results
Multi row insert (data sharing)	33% CPU reduction from V9, 4x improvement from V8 due to LRSN spin reduction
Parallel Index Update	30-40% Elapsed time improvement with class 2 CPU time reduction
Inline LOB	SELECT LOB shows 80% CPU reduction
Include Index	17% CPU reduction in insert after using INCLUDE INDEX
Hash Access	<p>20-30% CPU reduction in random access</p> <p>16% CPU reduction comparing Hash Access and Index-data access.</p> <p>5% CPU reduction comparing Hash against Index only access</p> <p>Further improvements delivered late in the beta program.</p>

## Performance and Scalability ...

- Measurements of IBM Relational Warehouse Workload (IRWW) with data sharing
  - Base: DB2 9 NFM REBIND with PLANMGMT EXTENDED
  - DB2 9 NFM → DB2 10 CM without REBIND showed 1.3% CPU reduction
  - DB2 10 CM REBIND with same access path showed 4.8% CPU reduction
  - DB2 10 NFM brought 5.1% CPU reduction
  - DB2 10 CM or NFM with RELEASE DEALLOCATE 12.6% CPU reduction from DB2 9



## Performance and Scalability ...

- Query performance enhancements
  - No REBIND required for
    - Index list prefetch
    - INSERT index read I/O parallelism
    - Workfile spanned records
    - SQLPL performance
    - High performance DBATs
    - Inline LOBs

## Performance and Scalability ...

- Query performance enhancements ...
  - REBIND required for
    - Use of RELEASE(DEALLOCATE)
    - Early evaluation of residual predicates
    - IN-list improvements (new access method)
    - SQL pagination (new access method)
    - Query parallelism improvements
    - Index include columns
    - More aggressive view/table expression merge
    - Predicate evaluation enhancements
    - RID list overflow improvements
  - Execute RUNSTATS before REBIND
    - When coming from V8, to collect improved index statistics including CLUSTERRATIOF
    - When coming from V9, if do not already include the KEYCARD option of RUNSTATS

## Performance and Scalability ...

- 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
  - 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)
    - APAR PM56845 now open to provide option for OPTIMIZE FOR 1 ROW to allow sort access plans



## Performance and Scalability ...

- Increase in DB2 system address space CPU resource consumption
  - DBM1 SRB
    - More use of prefetch
      - Row level sequential detection and progressive prefetch
      - INSERT index read I/O parallelism
      - Index list prefetch when disorganised index
      - After BIND, more use of list prefetch
    - zIIP offload for prefetch and deferred write
      - Seeing 50-70% zIIP offload achieved
  - DBM1 TCB
    - Closing of high use CLOSE=YES datasets when hitting DSMAX because of stale list
    - See APAR PM56725 for this issue
  - MSTR TCB
    - Increase related to real storage monitoring which was introduced (APAR PM24723)
    - DB2 is calling a z/OS RSM service for COUNTPAGES function which serialised the frame access with spin loop
    - CPU increase especially when multiple DB2 subsystems running on the same LPAR
    - See z/OS APAR OA37821 and corresponding DB2 APAR PM49816 for this issue

## Performance and Scalability ...

- Bufferpool Page Classification
  - Sequential, dynamic, and list prefetch are all treated the same
    - Pages are marked *sequential* at the time they are prefetched
  - In V8
    - These pages were reclassified as *random* when subsequently touched via getpage
  - In V9 and V10
    - None of them will be re-classified to *random* on a getpage
- CF utilisation issue
  - Root cause is the way GBP data in the CF is deleted in V10
  - CF DELETE call option being invoked by V10 deletes both data and directory entries in one CF call
    - But the code path used with name class mask requires more internal CFCC serialisation and can take longer
      - Results in significantly more re-drives of the DELETE requests observed with V10
      - Increased volume of re-drives causes the significant increase in CF utilisation
  - See APAR PM51467 for this issue

## Performance and Scalability ...

- DB2 10 and z196 synergy
  - Taking the general case, performance improvement from V9 to V10 observed on z10 processor should be in same range on z196 processor as long as they are measured on the same number of processors
    - Expectation is still in the 5-10% range
  - Apart from MIPs improvement, z196 provides
    - Higher cache hit ratio thus better scalability as number of processors per LPAR increases (more than 16 processors per LPAR)
  - V10 performance on z196
    - Scales better with more processors per LPAR than z10
    - Can run with higher number of concurrent threads
  - IBM measurement shows 20% ITR improvement from V9 (with a few benchmark specials) compared to V10 on z196 80-way with IRWW-like workload
    - Measurement is extreme case
    - Will only apply to very high end customers
    - Not a general message
  - Why does V10 run better on z196
    - Latch contention reductions, 1MB real storage page frame size, general path length

## Performance and Scalability ...

- Use of 1MB size real storage page frames on z10 and z196
  - Long term bufferpool page fix was introduced in V8 to reduce CPU
    - Many customers reluctant to use PGFIX=YES
    - Potential for real storage shortage because running too close to the edge of the amount of real storage provisioned
    - Customers do understand the value, but it only applies for a few hours each day
    - But page fix is a long term decision
    - In most cases requires DB2 recycle to change attribute
    - 75% cost reduction on real storage on z196 (USD1.5K vs. USD6K)

## Performance and Scalability ...

- Use of 1MB size real storage page frames on z10 and z196 ...
  - 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
  - Involves partitioning real storage into 4KB and 1MB size page frames
    - Specified by LFAREA xx% in IEASYSnn parmlib member and only changeable by IPL
    - 1MB size page frames are non-pageable
    - 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
  - Do not use 1MB size real storage frames until running smoothly on V10
  - Make sure any critical z/OS maintenance is applied before using 1MB size real storage page frames

## Performance and Scalability ...

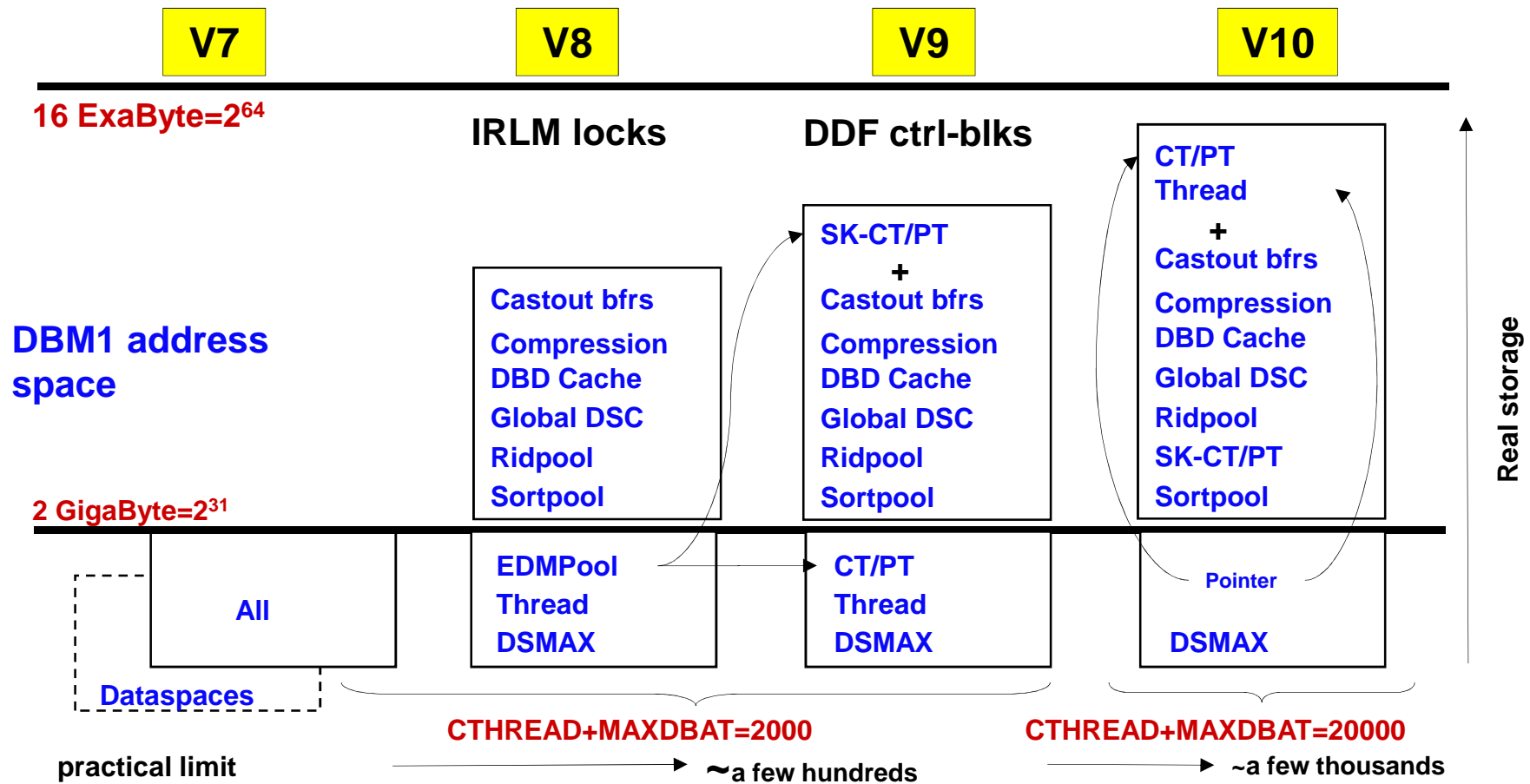
- Use of 1MB size real storage page frames on z10 and z196 ...
  - Useful commands
    - -DISPLAY BUFFERPOOL(BP1) SERVICE=4
      - Useful command to find out how many 1MB size page frames are being used
      - Especially useful when running multiple DB2 subsystems on the same LPAR
      - See DSNB999I message
    - -DISPLAY VIRTSTOR,LFAREA
      - Show total LFAREA, allocation split across 4KB and 1MB size frames, what is available
      - See IAR019I message

## Performance and Scalability ...

- Exceptions where CPU regression for very light OLTP transactions
  - Skinny packages with few simple SQL
  - Package allocation cost overrides benefit from SQL optimizations in V10
  - APAR PM31614 may solve this by improving package allocation performance
  - Good candidate for the use of persistent threads with `RELEASE(DEALLOCATE)` and will help compensate

## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time



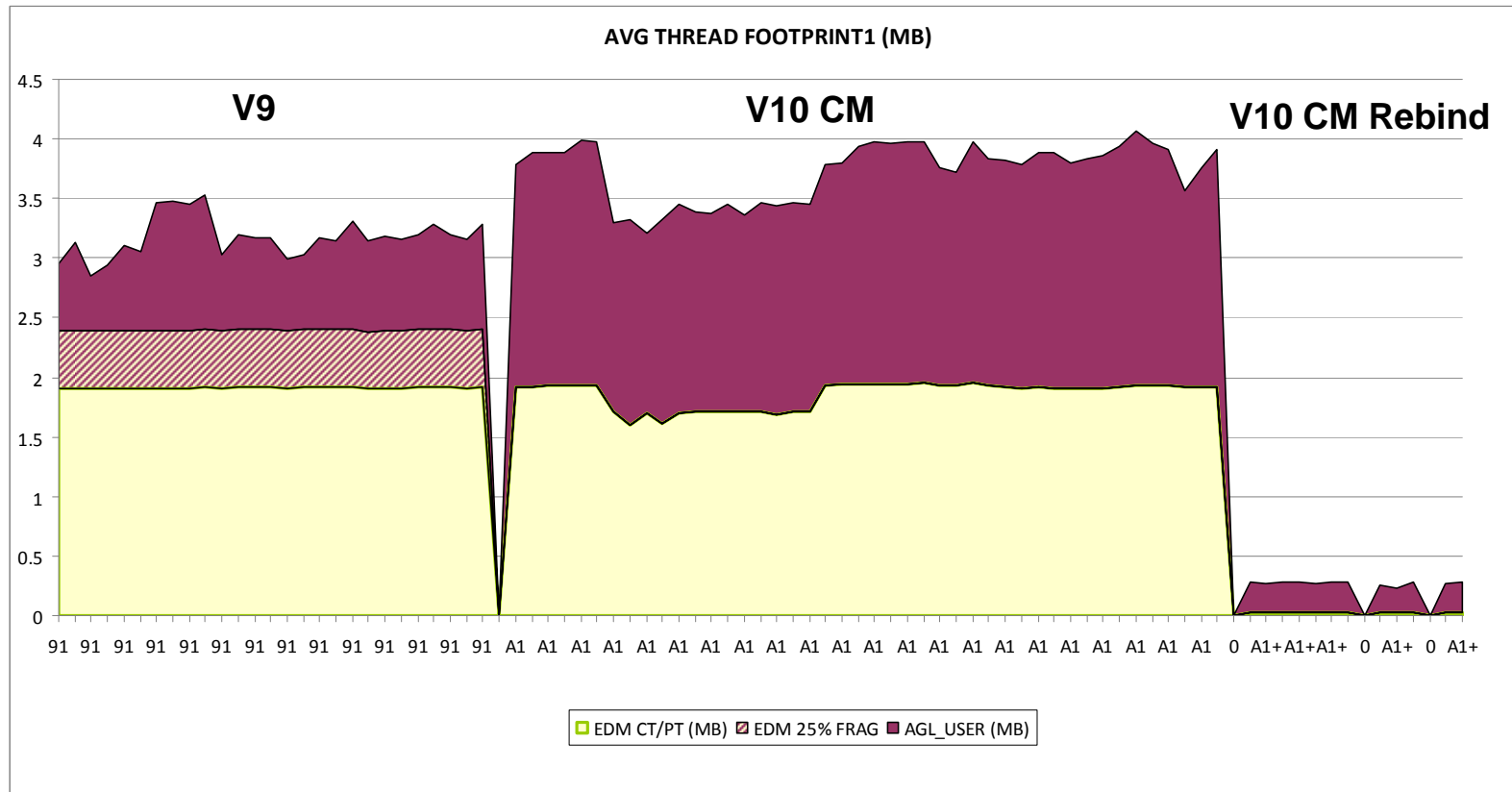


## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time
  - Available in CM
  - Requirement to REBIND static SQL packages to accrue maximum benefit
  - Very good results achieved (up to 90% VSCR)
  - Have high degree of confidence that problem addressed
    - Real world proposition: 500 -> 2500-3000 threads plus
  - Limiting factors now on vertical scalability (# number of threads, thread storage footprint)
    - Amount of real storage provisioned on the LPAR
    - Log latch (LC19) contention
    - ESQA/ECSA (31-bit) storage

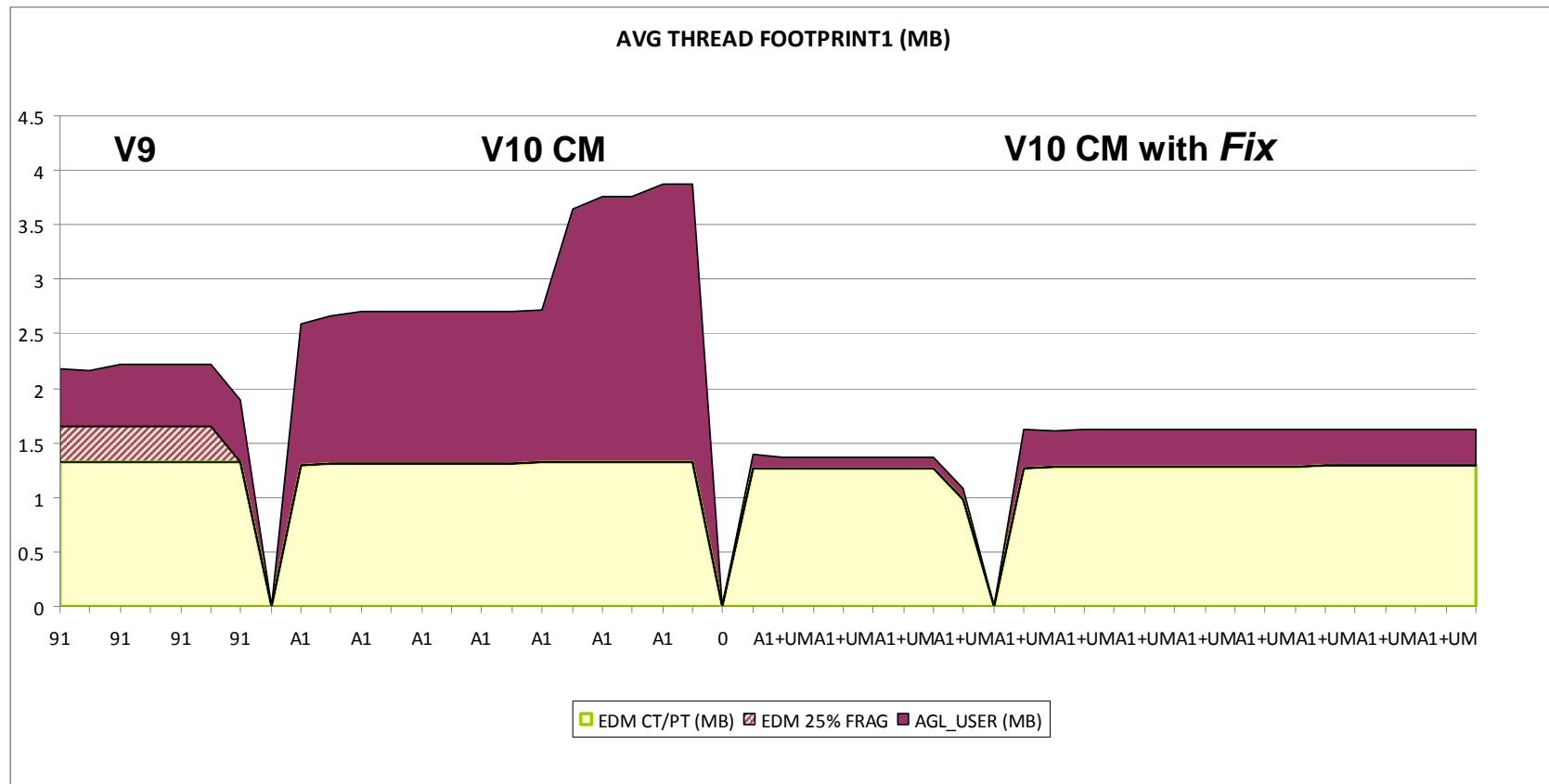
## Performance and Scalability ...

- DBM1 31-bit Thread Storage V9 vs. V10 – Initially but corrected prior to GA



## Performance and Scalability ...

- DBM1 31-bit Thread Storage V9 vs. V10 – as at GA after Fix



## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time
  - Major customer opportunities here for 31-bit VSCR and improved price/performance
    - Potential to reduce legacy OLTP transaction CPU cost through use of
      - More CICS protected ENTRY (persistent) threads
      - More use of RELEASE(DEALLOCATE) with next/existing persistent threads
    - Potential to reduce CPU for DRDA transactions by using High Performance DBAT
      - Must be using CMTSTAT=INACTIVE so that threads can be pooled and reused
      - Packages must be bound with RELEASE(DEALLOCATE) to get reuse for same connection
      - MODIFY DDF PKGREL(BNDOPT) must also be in effect
      - Do not to overuse RELEASE(DEALLOCATE) on packages
        - > Will drive up the MAXDBAT requirement
    - Potential to reduce CPU when using KEEPYNAMIC(YES) e.g., SAP
      - Increase MAXKEEPD to improve Local Dynamic Cache Hit Ratio and reduce the number of short prepares
  - Must provision additional real storage to back the requirement for each opportunity

## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time
  - More persistent threads with RELEASE(DEALLOCATE) is also trade off with BIND/REBIND and DDL concurrency
  - For RELEASE(DEALLOCATE) some locks are held beyond commit until thread termination
    - Mass delete locks (SQL DELETE without WHERE clause)
    - Gross level lock acquired on behalf of a SQL LOCK TABLE
    - Note: no longer a problem for gross level lock acquired by lock escalation
  - CICS-DB2 accounting for cost of thread create and terminate, or avoidance thereof
    - CICS uses the L8 TCB to access DB2 irrespective of whether the application is thread safe or not
    - Thread create and terminate cost will clock against the L8 TCB and will be in the CICS SMF Type 110 record
    - Note: prior to OTE did not capture the thread create in the SMF Type 110

## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time
  - High Performance DBATs (Hi-Perf DBATs) is a new type of distributed thread
    - Must be using CMTSTAT=INACTIVE so that threads can be pooled and reused
    - Packages must be bound with RELEASE(DEALLOCATE) to get reuse for same connection and -MODIFY DDF PKGREL(BNDOPT) must also be in effect
    - When a DBAT can be pooled after end of client's UOW
      - Now DBAT and client connection will remain active together
        - > Still cut an accounting record and end the enclave
      - After the Hi-Perf DBAT has been reused 200 times
        - > DBAT will be purged and client connection will then go inactive
      - All the interactions with the client will still be the same in that if the client is part of a sysplex workload balancing setup, it will still receive indications that the connection can be multiplexed amongst many client connections
      - IDTHTOIN will not apply if the if the Hi-Perf DBAT is waiting for the next client UOW
      - If Hi-Perf DBAT has not received new work for POOLINAC time
        - > DBAT will be purged and the connection will go inactive
      - If # of Hi-Perf DBATs exceed 50% of MAXDBAT threshold
        - > DBATs will be pooled at commit and package resources copied/allocated as RELEASE(COMMIT)
      - Hi-Perf DBATs can be purged to allow DDL, BIND, and utilities to break in
        - > Via -MODIFY DDF PKGREL(COMMIT)

## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time
  - High Performance DBATs (Hi-Perf DBATs) should be carefully
    - Want to have some high performance applications running on LUW application servers connected to DB2 10 for z/OS running with High Performance DBATs and others not
    - Standard ODBC and JDBC packages supplied with drivers/connect packages should be bound twice into two different package collections e.g.,
      - The CS package in collection1 will be bound with RELEASE(DEALLOCATE) so that the applications using that package will be eligible to use high performance DBATs
      - The CS package in collection2 (e.g., NULLID) would be bound with RELEASE(COMMIT) and would not use high performance DBATs
    - For JDBC applications
      - Set the currentPackageSet property in the respective datasource
    - For .NET and ODBC / CLI applications
      - Set CurrentPackageSet parameter in the db2dsdriver.cfg configuration

## Performance and Scalability ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time
  - Potential to reduce the number of DB2 subsystems in data sharing group
    - First step is to collapse multiple DB2 members running on the same LPAR
    - May then be able to reduce the number of LPARs/DB2 members
    - Consider the increase in logging rate per DB2 member
      - Possible aggravation of LC19 contention despite V10 improvement
    - Consider the increase in SMF data volume per LPAR
      - Can enable DB2 compression of SMF data to reduce SMF data volume
        - > Experience is that Accounting records compress 70-80%
        - > Tiny CPU overhead at ~1%
      - Re-consider use of accounting roll up for DDF and RRSAP workload (default)
        - > Compromises performance PD/PSI as lose information on outlying transactions
        - > Significant enhancements to package level accounting so it is now useful
    - Consider the increased DUMPSRV and MAXSPACE requirement
  - Re-emphasise the continued value of data sharing to differentiate the platform
    - Support avoidance of planned outages
    - Avoid humongous single points of failure
    - Minimum of 4-way for true continuous availability



## Performance and Scalability ...

- 64-bit virtual storage
  - Three large areas allocated at IPL time
    - Common 6GB (z/OS default)
      - Addressable by all authorized programs on the LPAR
      - IFC for accounting
    - Private 1TB
      - Buffer pools
      - XML and LOB are huge users, RTS blocks, TRACE buffers,
      - some RID blocks, IFC work buffers and few other misc system pools
    - Shared (Private) 128GB
      - Addressable by all authorized products which have registered their interest to z/OS using the unique object token created when the memory object is created
      - V9 introduced 64-bit shared private storage but it was used in limited fashion
      - Almost all the DB2 storage in V10 is now 64-bit shared private
  - DB2 is only "reserving" virtual storage, it does not mean it is being used
    - It costs nothing to reserve virtual storage i.e., addressing range
    - Having a fixed size areas is a lazy design but it makes it easier for serialization
  - Needs to be backed by real storage when it is allocated within the reference area

## Performance and Scalability ...

- 31-bit and 64-bit virtual storage contraction
  - CONTSTOR=YES and MINSTOR=YES
    - These existing system parameters drive the contraction of 31-bit storage pools and the best fit allocation of 31-bit storage respectively
    - Not applicable to 64-bit storage
    - Not as critical as before V10
    - Assuming generous DBM1 31-bit VSCR in V10, set CONSTOR=MINSTOR=NO
  - 64-bit thread pools are contracted under control of
    - Commit count
    - New Real Storage Management DISCARD function (see follow on slides)

## Performance and Scalability ...

- Real storage
  - Need to carefully plan, provision and monitor real storage consumption
  - Prior to V10 a hidden zparm SPRMRSMX ('real storage kill switch') existed
    - SPRMRSMX prevents a runaway DB2 subsystem from taking the LPAR down
      - Should be used when there is more than one DB2 subsystem running on the same LPAR
      - Aim is to prevent multiple outages being caused by a single DB2 subsystem outage
      - Should be set to 1.5x to 2x normal DB2 subsystem usage
      - Kills the DB2 subsystem when SPRMRSMX value reached
    - With V10, will need to factor in 64-bit shared and common use to establish new footprint
  - Problems with introduction of V10
    - Unable to monitor the REAL and AUX storage frames used for 64-bit shared storage
      - V9 not really an issue, as limited use of 64-bit shared
      - But now V10 makes extensive use of 64-bit shared
    - LPAR level instrumentation buckets for REAL and AUX storage use
      - If more than one DB2 subsystem on the same LPAR then the numbers reported are inaccurate
      - Only able to get reliable numbers if only one subsystem like DB2 on the LPAR uses 64-bit shared
    - Lack of ENF 55 condition monitoring
      - 50% of AUX used

## Performance and Scalability ...

- Real storage ...
  - DB2 APAR PM24723 is very important
    - Monitoring issue is addressed and new extensions to IFCID 225 provided
      - Pre-req is new MVS APAR OA35885 which provides a new callable service to RSM to provide REAL and AUX used for addressing range for shared objects
    - SPRMRSMX hidden zparm now becomes an opaque parameter REALSTORAGE\_MAX
    - Introduces DISCARD mode to contract storage usage to protect against excessive paging and use of AUX
      - New zparm REALSTORAGE\_MANAGEMENT controls when DB2 frees storage frames back to z/OS
        - > ON -> Discard unused frames all the time - discard stack, thread storage, keep footprint small
        - > OFF -> Do not discard unused frames unless things are getting out of hand
        - > AUTO (default) -> Detect whether paging is imminent and reduce the frame counts to avoid system paging
      - With AUTO, DB2 monitors paging rates, switches between ON/OFF and decides when to discard frames based on
        - > 80% of SPRMRSMX reached
        - > 50% of AUX (ENF55 condition) used
        - > Hitting AVQLOW (available real storage frame)
      - New messages (DSNV516I, 517I) for when paging rate thresholds cause DB2 to free real frames
    - Strong recommendation to apply PTF for APAR PM24723 before going into business production and to run with REALSTORAGE\_MANAGEMENT=AUTO

## Performance and Scalability ...

- Monitoring Virtual and Release Storage
  - SPREADSHEETDD support in OMPE has not been enhanced to support V10
    - OMPE are working on a 'generic' spreadsheet generator
    - Outstanding requirement to also include serviceability fields
  - MEMU2 and MEMUSAGE already enhanced for V10 and available on the DB2 for z/OS Exchange community website on IBM My developerWorks
    1. From IBM My developerWorks My Home (sign in with your IBM login at <https://www.ibm.com/developerworks/mydeveloperworks/homepage>), search 'memu2' in All My developerWorks.
    2. (From DB2 for z/OS Exchange (<http://www.ibm.com/developerworks/software/exchange/db2zos>), click on 'View and download examples'. The file is tagged with 'memu2'.
    3. To access MEMU2 directly (but note that if you want to be kept informed of updates and new versions, you need to log on to developerWorks rather than download the file anonymously...)

<https://www.ibm.com/developerworks/mydeveloperworks/files/app/file/3af12254-4781-43f3-b4a8-3336e09c36df?lang=en>

- V10

<https://www.ibm.com/developerworks/mydeveloperworks/files/app/file/e2736ed5-0c73-4c59-b291-9da08255b941?lang=en>

## Performance and Scalability ...

- High INSERT performance
  - Significant improvements for UTS
    - Now support for MEMBER CLUSTER
    - Changes to space search algorithm (like classic partitioned)
  - Goal was for UTS to be equal or better than classic partitioned (PTS)
    - Not there yet, but much closer
    - Very workload dependent
    - Some good, some worse
    - Still trade off between space vs. throughput and reduced contention
    - Work still to do on UTS PBR/PBG with RLL and sequential insert

## Performance and Scalability ...

- High INSERT performance ...
  - Reduced LRSN spin for inserts to the same page
    - Works well for MRI and INSERT within loop in a data sharing environment
  - Optimization for 'pocket' sequential insert works well
    - Index manager picks the candidate RID during sequential insert (next lowest key rid)
    - Higher chance to find the space and avoiding a space search
  - Parallel index IO works very well when activated for random key inserts
    - $\geq 3$  indexes
    - Prefetch and deferred write offload to zIIP to compensate

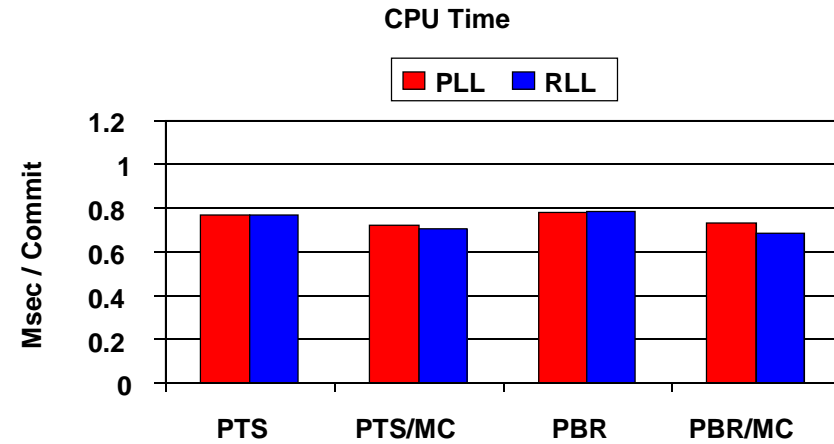
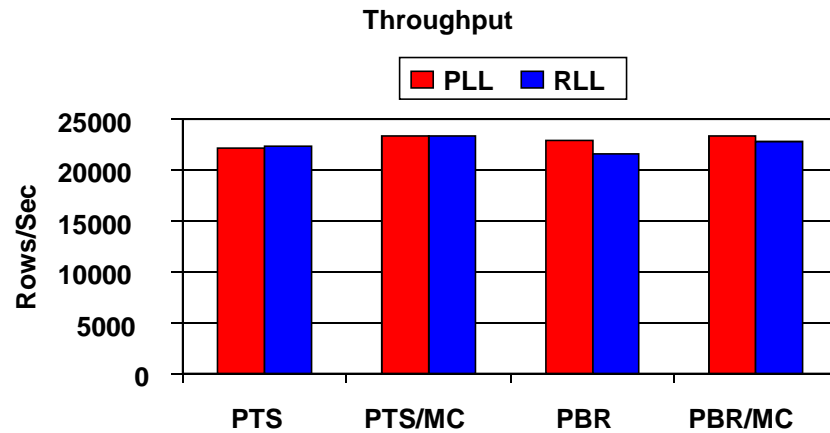
## High Insert Workload Description

- 2-way data sharing
- Database schema
  - 3 tables with total of 6 indexes (4 unique, 2 non-unique indexes, 2 secondary indexes)
  - Table space types: Classic Partitioned, Classic Segmented, UTS (PBR, PBG)
- SQL
  - INSERTs contain 5, 9 and 46 columns of integer, bigint, char, varchar, decimal and timestamp data type
- Application implemented in Java
- Sequential inserts into empty tables
  - 240 concurrent threads
  - Multi-row inserts (100)
- Random inserts into populated tables
  - 200 concurrent threads
  - Single-row inserts

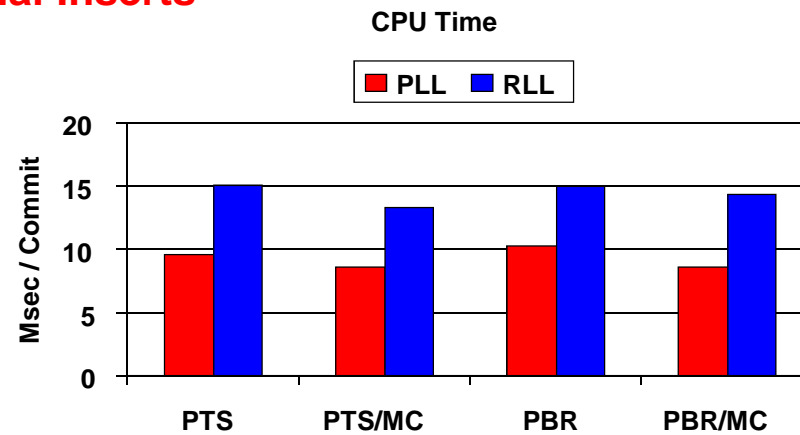
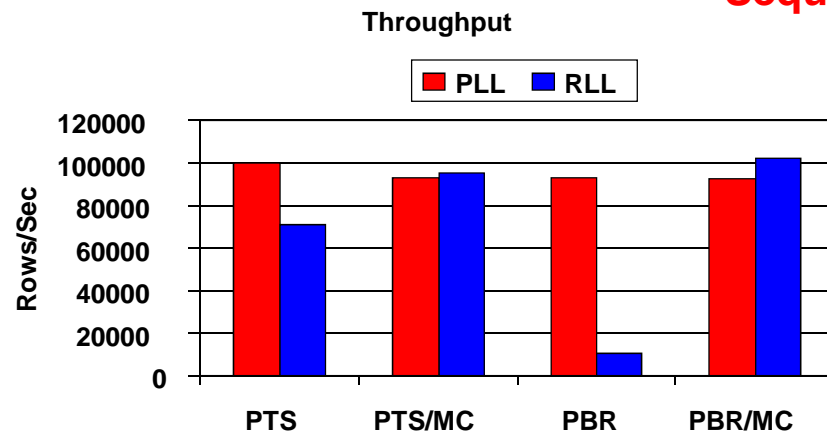


# DB2 10 Range Defined Table Spaces

## Random Inserts

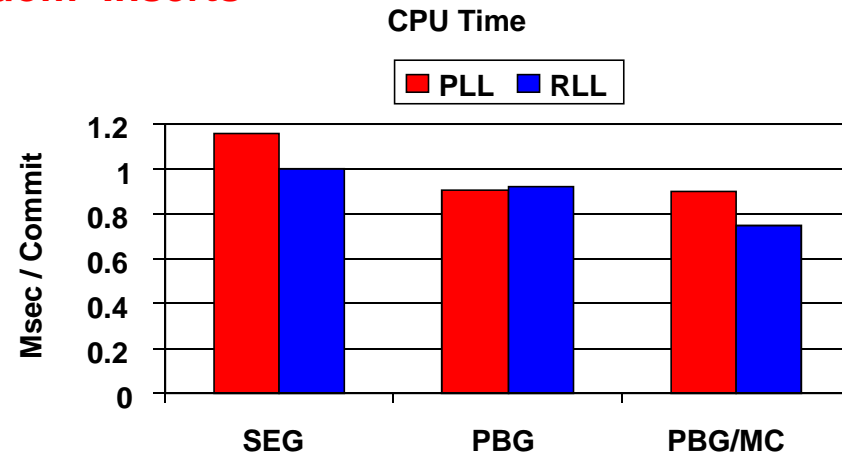
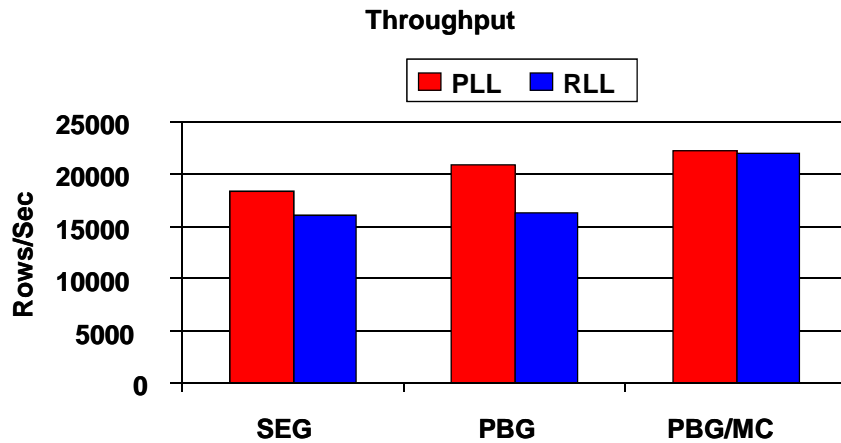


## Sequential Inserts

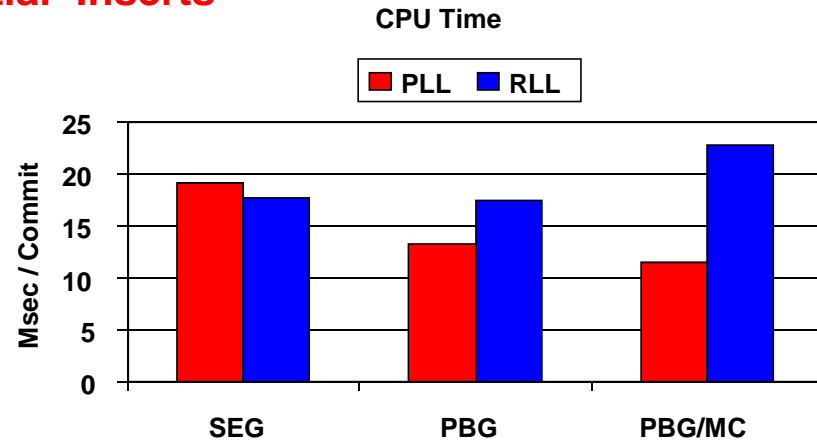
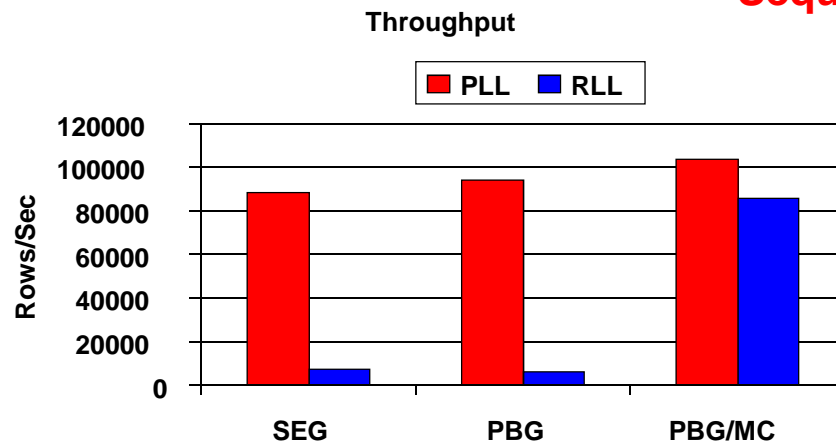


# DB2 10 Non-range Defined Table Spaces

## Random Inserts



## Sequential Inserts



## Performance and Scalability ...

- Accounting Trace Class 3 enhancement – separate counters
  - IRLM Lock/Latch waits
  - DB2 Latch waits
- Data sharing
  - Faster DB2 shut down by avoiding local buffer pool scan per GBP-dependent object
  - Avoiding scan of XXXL local Buffer pool when
    - Pageset/partition transitions into GBP-dependency
    - Pageset/partition transitions out of GBP-dependency
- Inline LOBs work very well if you hit the sweet spot
  - Potential for significant CPU and elapsed time improvement with the right inline value
  - Trade off in setting the right inline value
    - Avoiding access to auxiliary tablespace
    - Increasing base row size with fewer rows per page
    - May have to increased page size
  - Inline portion can be compressed
  - Significant space savings with small LOBs (<1000 bytes)

## Performance and Scalability ...

- Compress on INSERT
  - Compression ratios almost as good compared with running REORG later
- Active log writes
  - Prior to V10, log writes are done serially when re-writing partial CIs
  - Determined that destructive writes due to IO errors no longer occur
  - Now all log write IOs are done in parallel
  - Elapsed time improvements
- Limited value of Currently Committed locking semantics
  - SQL UPDATE not supported

## Performance and Scalability ...

- Hash access vs. Index only access and index lookaside
  - Competes against index only access and index lookaside
    - Advantage that index only access still provides for clustered data access
    - Can now have unique index with INCLUDE columns
      - Reduce number of indexes required for performance reasons
      - Improve insert, update and delete performance
  - Need to find the sweet spot
    - High NLEVELS in index ( $\geq 3$ )
    - Purely direct row access by primary key
    - Truly random access
    - Read intensive, not volatile
    - No range queries
    - Many rows per page etc
  - Space allocation of fixed hash space is key to control overflow
    - Too small will lead to rows in overflow
    - Too large will lead to random IO
    - REORG AUTOESTSPACE(YES) but still some rows in overflow
  - Degraded LOAD and REORG utility performance

## Performance and Scalability ...

- Improved index space search when index leaf page split
  - In V8/V9, Index Manager
    - First searches the space map page covering the splitting page
    - If there is no free entry, searches the space map pages starting from the first space map page to the highest allocated page
    - If all the space map pages are full, has to extend
    - If the index is huge and all the space map pages having free entries are toward the end of the index, this process can take a very long time
  - In V10, Index Manager
    - After searching the space map page covering the splitting page, and if it is full, will start searching from the page number it last remembered having the free entry in it (page A)
    - When it reaches to the highest allocated page, it starts from the beginning and search forward till it reaches to page A
    - Then it has to extend since the entire index is full
    - The page number of the space map page having free entry is stored in an in-memory control block
      - When Index Manager finds a space map page with free entry, it is updated to be the page number of that space map page
      - Index Manager updates this value when an index page is deleted or when the index is mass deleted
    - Retrofitted back to V9 via APAR PM15474

## Performance and Scalability ...

- 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

## BIND, REBIND and EXPLAIN

- Value of REBIND under V10
  - Improved performance from new run time (avoid puffing, enable SPROC)
  - Maximize DBM1 31-bit VSCR
  - Allow RID overflow to workfile
  - Reduce exposure to problems with migrated packages from earlier releases
    - INCORROUTs
    - Thread abends
- Can mitigate exposure to bad access path change introduced with REBIND which leads to degraded run time performance (regression)
  - Use access plan stability (PLANMGMT=EXTENDED|BASIC) and fallback if needed
    - PLANMGMT=EXTENDED is now the default
  - Use APREUSE and APCOMPARE



## BIND, REBIND and EXPLAIN ...

- Access Path Reuse - APREUSE and APCOMPARE
  - Introduced in V10 through APARs PM33767 (function) and PM25679 (enabled)
  - Works independently of access plan stability (PLANMGMT=EXTENDED|BASIC)
  - Provides ability to generate new SQL run time but will attempt to reuse the existing access path by internally using OPTHINTS
    - Applies to both BIND REPLACE and REBIND
  - Available in CM when migrating from V9 -> V10 and then applies to subsequent BIND/REBIND in V10
  - Only works when package previously bound on V9 or higher
    - Starting with V9, EXPLAIN information is saved away inside the package in SPT01
    - Referred to as “Explain Data Block (EDB)” which is a compact representation of PLAN\_TABLE information
  - Not guaranteed to succeed in all cases as OPTHINTS are not enforceable 100% of the time
    - Some incompatibilities between old and new release (restrictions)
    - Seeing 95-99% compatibility

## BIND, REBIND and EXPLAIN ...

- Access Path Reuse - APREUSE and APCOMPARE ...
  - Conservative approach where minimal access path changes are required
    - Step 0 (Optional)
      - Use REBIND ... **EXPLAIN(ONLY) + APREUSE(ERROR)**
      - Perform an impact analysis before actual REBINDs
    - Step 1: REBIND PACKAGE (\*)
      - Use PLANMGMT(EXTENDED) ... backup of V9 access paths, just in case  
+ EXPLAIN(YES)  
**+ APREUSE(ERROR)**
    - Step 2: For packages that failed Step 1 (i.e., leftovers)
      - 2a: Leave them as is ... they will be at the old level
      - OR
      - 2b. REBIND with PLANMGMT(EXTENDED) + APREUSE(NO)
        - > This step exposes yourself to access path changes
        - > But you have a backup

## BIND, REBIND and EXPLAIN ...

- RUNSTATS/REBIND recommendations based upon on scenario
  - V8 preparation
    - If RUNSTATS will be difficult on large number of objects immediately after migration to V9/10, then REORG and/or RUNSTATS (V8) immediately prior to migration can reduce RUNSTATS need on V9/10 - as RUNSTATS INDEX under V10 can be sufficient to capture new CR/DRF
  - V9 migration
    - RUNSTATS objects as soon as possible after migration
      - Target dynamic applications first as these are exposed to new access paths immediately
    - Delay static REBINDs until associated objects have RUNSTATS run
  - V8->V10 migration
    - RUNSTATS objects as soon as possible after migration
      - Target dynamic applications first as these are exposed to new access paths immediately
      - Equal priority - target static parallelism packages to REBIND to avoid incremental bind at each execution
    - Delay non-parallelism REBINDs until associated objects have RUNSTATS run
  - V9->V10 migration
    - REBIND static parallelism packages as soon as possible to avoid incremental bind at each execution
    - Delay non-parallelism REBINDs until associated objects have RUNSTATS run
    - BIND/REBIND options APREUSE/APCOMPARE are available on V10 for packages bound on V9

## BIND, REBIND and EXPLAIN ...

- RUNSTATS/REBIND recommendations based upon on scenario ...
  - V8/9 co-existence
    - Set STATCLUS=STANDARD while in co-existence with V8
    - Set ABIND=COEXIST while in co-existence with V8
    - Avoid REBIND while in co-existence with V8
    - Follow V9 migration steps after all members are V9, including resetting the following zparms
      - Set STATCLUS=ENHANCED
      - Set ABIND=YES
  - V8/10 co-existence
    - Set STATCLUS=STANDARD while in co-existence with V8
    - Set ABIND=COEXIST while in co-existence with V8
    - What to do with static parallel queries?
      - Accept incremental bind whenever executed on V10 member
      - OR, REBIND with DEGREE('1') to disable parallelism while in co-existence.
    - Follow V8-V10 migration steps after all members are V10, including resetting the following zparms
      - Set STATCLUS=ENHANCED
      - Set ABIND=YES

## BIND, REBIND and EXPLAIN ...

- RUNSTATS/REBIND recommendations based upon on scenario ...
  - V9/10 co-existence
    - Set ABIND=COEXIST while in co-existence with V8
    - What to do with static parallel queries?
      - Accept incremental bind whenever executed on V10 member
      - OR, REBIND with DEGREE('1') to disable parallelism while in co-existence.
    - Follow V8-V10 migration steps after all members are V10, including resetting the following zparms
      - Set ABIND=YES

## BIND, REBIND and EXPLAIN ...

- Single thread BIND/REBIND performance
  - Degraded CPU and elapsed time performance on entry to CM
    - PLANMGMT=EXTENDED is now default
    - New indexes defined for post ENFM when hash links are eliminated
    - Change in access path (index access) on entry to CM
    - No concurrency improvement until after Catalog restructure in ENFM
- Concurrent BIND/REBIND performance
  - Problems addressed
    - Performance problems related to DELETE/INSERT process
    - Space growth in SPT01 for both LOB space and base table
  - Now working well
    - Inefficient space search for out of line LOB in data sharing (APAR PM24721)
    - Inline LOB with compression for SPT01 to address SPT01 growth (APAR PM27073)
    - More efficient space reuse for base table and UTS (APAR PM27973)
  - Recommendations
    - Customers need to change existing procedures to go parallel
    - But cannot do this until post ENFM
    - Benefit from reducing application down time to implement new application releases

## BIND, REBIND and EXPLAIN ...

### ■ EXPLAIN tables

- Format and CCSID from previous releases is deprecated in V10
  - Cannot use pre V8 format
    - SQLCODE -20008
  - V8 or V9 format
    - Warning SQLCODE +20520 regardless of CCSID EBCDIC or UNICODE
  - Must not use CCSID EBCDIC with V10 format
    - EXPLAIN fails with RC=8 DSNT408I SQLCODE = -878
    - BIND with EXPLAIN fails with RC=8 DSNX200I
- Recommendations
  - Use CCSID UNICODE in all supported releases (V8, V9, V10) due to problems with character truncation and conversion etc
  - Use the V10 extended column format with CCSID UNICODE when
    - Applications access EXPLAIN tables and can only tolerate SQLCODE 0 or +100
  - V10 column format is supported under V8 and V9 with the SPE fallback APAR PK85956 applied with the exception of
    - DSN\_STATEMENT\_CACHE\_TABLE due to the BIGINT columns
- APAR PK85068 can help migrate V8 or V9 format to the new V10 format with CCSID UNICODE

## Availability

- Online Schema Evolution ('Deferred Alter')
  - Migrate from classic table space types (simple, segmented, partitioned) to UTS PBG/PBR
    - One way ticket only
  - UTS is pre-requisite for Cloned Table, Hash, Inline LOB, Currently Committed
  - Once migrated to UTS PBG/PBR can change attributes in both directions
    - DSSIZE, index page size, MEMBER CLUSTER, Hash Access, ...
  - Benefits
    - Streamed line way to move to UTS
    - Reduce administrative time and cost
    - Cuts down on errors
    - Reduce outages
  - Issue that PIT recovery to point before successful materializing REORG not possible
    - Incorrect results from REORG
    - Application change rollback



## Availability ...

- Restart Light enhancement
  - LBACKOUT will now be honoured
    - LBACKOUT=YES|AUTO will cause postponed abort (PA) URs to be created
    - Restart will complete
    - DB2 will shut down
    - Retained locks will be kept on behalf of PA URs
  - Controlled via new system parameter
  - Also retrofitted back to V9 via APAR
- Online REORG with FORCE
  - Only running threads which are blocking are cancelled
  - Threads which are suspended / inactive will cause REORG to still fail
- Online REORG LOB with DISCARD
  - Cannot handle LOB columns greater than 32KB

## Migration and Planning

- Migration process very similar to V8 and V9
  - Works well with few problems with migration fallback
- Migration from either DB2 for z/OS V8 NFM or DB2 9 for z/OS NFM
- These migration fallback sequences are not valid
  - V8 NFM > V10 CM8 > V8 NFM > **V9 CM**
  - V8 NFM > V9 CM > V8 NFM > **V10 CM8**
- Fallback Toleration SPE
  - APAR PK56922
- Early Code
  - For V8/V9 APAR PK87280 (superseeds APAR PK61766)
- Information APARs
  - II14474: V8 to V10
  - II14477: V9 to V10

## Migration and Planning ...

V8 premigration checklist:

[http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z\\_premigr8checklist.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z_premigr8checklist.htm)

V8 migration checklist:

[http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z\\_migr8checklist.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z_migr8checklist.htm)

V9 premigration checklist:

[http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z\\_premigr9checklist.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z_premigr9checklist.htm)

V9 migration checklist:

[http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z\\_migr9checklist.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z_migr9checklist.htm)

## Migration and Planning ...

- Use of V10 Early Code with V8
  - It will take an IPL to originally install the V10 Early Code
  - V8 Early Code does not understand –REFRESH
  - However, subsequent maintenance to the V10 Early Code can be accomplished with a -REFRESH command
- If coming from V8
  - BSDS must be reformatted for larger active / archive tracking
- IPL amounts for need to be adjusted based on number of DB2 members
  - 64-bit Private (1TB)
  - 64-bit Shared (128GB)
  - 64-bit Common (6GB)

## Migration and Planning ...

- DB2 Connect
  - Minimum level
    - V9.5 FP7
    - V9.7 FP3A and for new functions
  - Start with the latest levels based on CST/RSU and stabilise

## Migration and Planning ...

- DBRMs bound directly into plans no longer supported
  - If found in V10, will trigger auto bind into packages
  - For V8 and V9
    - APARs PK62876/PK79925 adds new syntax to convert from DBRMs to packages
      - REBIND PLAN option COLLID
      - Could result in access path change
    - APARs PM01821 (Version) and PM30382 (Location from \* to blank) should be on
      - Best to migrate DBRMs to packages before migrating to V10
- Old plans and packages bound prior to V6 will require REBIND
- Catalog and Directory must be SMS managed (EF, EA) ahead of CM
- PDSEs required for SDSNLOAD, SDSNLOD2, ADSNLOAD
- DSNHDECP NEWFUN=V10|V9|V8

## Removal of DDF Private Protocol

- Must absolutely eliminate all use of DDF Private Protocol before migrating
  - No longer supported In V10
  - Any local packages miss tagged with DDF Private Protocol will be tolerated
  - Otherwise package must exist in both local and remote sites
  - A lot of packages and plans are bound with DBPROTOCOL(PRIVATE) because this was the default (zparm DBPROTCL) when introduced in DB2 V6
    - DSNT226I is issued if DBPROTOCOL(PRIVATE) is used during REBIND

## Removal of DDF Private Protocol ...

- Must absolutely eliminate all use of DDF Private Protocol before migrating
  - APAR PK92339 in V8 introduces new zparm PRIVATE\_PROTOCOL=YES|NO
    - Must be in sync with zparm setting of DBPROTCL
    - DSNT225I message to indicate private protocol failure for REBIND
    - IFCID 157 can be used to identify packages which are using private protocol
    - Beware APAR PM17665 removes authorization behaviour for private protocol i.e., .
      - Execute privilege on package is required for the user running the program under DRDA
      - APAR PM37300 will provide relief (see later slides)
  - APAR PK64045 (PE, PM14816) in V8 new zparm DRDA\_RESOLVE\_ALIAS
    - Private protocol always performs alias resolution
      - 3-part name is sent to the server
    - If DRDA\_RESOLVE\_ALIAS is set to YES then also DRDA connections get this resolution e.g.,
      - For remote BIND with COPY option



## Security considerations when removing DDF Private Protocol

- This section also applies to customers using DRDA exclusively
- There are fundamental differences on how authorization is performed based on the distributed protocol used
- Private Protocol (DB2 for z/OS requester)
  - Supports static SQL statements only
  - Plan owner must have authorization to execute all SQL executed on the DB2 server
  - Plan owner authenticated on DB2 requester and not on the DB2 server
- DRDA Protocol
  - Supports both static and dynamic SQL statements
  - Primary auth ID and associated secondary auth IDs must have authorization to execute package and dynamic SQL on the DB2 server
  - Primary auth ID authenticated and secondary auth IDs are associated on DB2 server
- Prior to V10, Private Protocol and DRDA Protocol can be used by same application
  - Private Protocol security semantics was used due to possible inconsistent behavior which is dependent on how programs are coded and executed

## Security considerations when removing DDF Private Protocol ...

- APAR PK92339 introduced new zparm PRIVATE\_PROTOCOL=YES|NO
  - To prevent future introduction of PP then set PRIVATE\_PROTOCOL=NO
- The result of migrating to V10 or the introduction of APAR PM17665 under V8 or V9, when running with PRIVATE\_PROTOCOL=NO introduces the authorization changes at the DB2 for z/OS server for DRDA connections coming from DB2 for z/OS requester
  - PP security semantics are no longer used as default for access from a DB2 for z/OS requester
  - Plan owner value is ignored and connecting userid must be granted authority to execute the package at the remote site
  - Otherwise the connection will fail with SQLCODE -551
- As a result of customer complaints, APAR PM37300 introduces PRIVATE\_PROTOCOL=AUTH which allows an installation to
  - Disable PP but keep the plan owner authorization check (the "private protocol semantics")
- Migration to V10 or the application of PTF for APAR PM17665 does affect you even if you have everything already bound as DRDA

## Security considerations when removing DDF Private Protocol ...

- In summary
  - Before disabling private protocol, ensure all appropriate grants are performed
    - Grant execute privilege to any user who plans to run a package or stored procedure package from a DB2 for z/OS requester, just like other DRDA clients
  - DB2 V8 and V9 can disable private protocol but still maintain private protocol authorization checks by
    - Setting system parameter PRIVATE\_PROTOCOL=AUTH
  - DB2 10 does not support private protocol but can allow private protocol authorization checks for use of DRDA protocol for DB2 for z/OS requesters by
    - Setting system parameter PRIVATE\_PROTOCOL=AUTH

## Other

- Ability to create classic partitioned table space (PTS)
  - Classic PTS deprecated in V10
  - By default will be created as UTS PBR
  - UTS will only support table based controlled partition syntax
  - Options to be able to officially create classic PTS
    - Specify SEGSIZE 0 on CREATE TABLESPACE
    - Set new zparm DPSEGSZ=0 (default 32)
- Fast Log Apply storage
  - System parameter LOGAPSTG is eliminated
  - Total FLA storage is now 510MB
- Old COBOL and PL/1
  - V7 lookalike pre-compiler (DSNHPC7) for older COBOL and PL/I is still provided
- DDL Concurrency after Catalog restructure
  - Some help provided but concurrency issues not absolutely solved
  - Still deadlocks with parallel heavy DDL against different databases

## Other ...

- SPT01 compression is back
  - Via system parameter COMPRESS\_SPT01=YES (default is NO)
- Statistics Interval
  - Only the frequency of IFCIDs 105, 106, 199 are controlled via STATIME
  - IFCIDs 2, 202, 217, 225, 230 are always cut at a fixed 1 minute interval
  - Remember to normalise the data when comparing V8/V9 vs. V10

## Other ...

### ■ RUNSTATS

#### – Page Sampling

- Performance improvement can be phenomenal
- Potential issues with accuracy because error rates increase as the sample size decreases
- No sampling done on indexes

#### – zIIP offload

- Nearly all RUNSTATS INDEX processing is offloaded, but only 'basic' RUNSTATS TABLE processing is offloaded
- Much less for advanced/complex statistics
  - Not supported:
    - > Inline stats
    - > COLGROUP
    - > DSTATS
    - > Histogram stats

## Other ...

- Incompatible changes
  - CHAR() scalar function
    - Solution via APAR PM29124 to restore compatible behavior for CHAR

```

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	
99	0099,	40F0F0F9F96B	0099	99	99	F9F940404040	9
999	0999,	40F0F9F9F96B	0999	999	999	F9F9F9404040	99
9999	9999,	40F9F9F9F96B	9999	9999	9999	F9F9F9F94040	999

## Other ...

- Incompatible changes ...

- Create Trigger

- Impact: Trigger created in V10 not executable in V9
    - Workaround: drop and create trigger in V9
    - PMR open

```
DSNT408I  SQLCODE = -723, ERROR: AN ERROR OCCURRED IN A TRIGGERED SQL  
STATEMENT IN TRIGGER AWAT.OCA_TEST_TRIGGER. INFORMATION RETURNED: SQLCODE -  
904, SQLSTATE 57011, MESSAGE TOKENS 00E30305,00000801,AWAT.OCA_TEST_TRIGGER-  
.18E5950B04A23EEC, SECTION NUMBER 1
```



## Other ...

- Incompatible changes ...

- SQL stored procedures

```
SQLCODE = -904, ERROR: UNSUCCESSFUL EXECUTION CAUSED BY AN  
UNAVAILABLE RESOURCE. REASON 00E7009E, TYPE OF RESOURCE 00000801,  
AND RESOURCE NAME BE2TFKT.LESE_VARIANTEN.18DEB14D119115C4
```

- Impact: Deployment has to be executed on V9 member in V10
- Workaround: run ALTER PROCEDURE REGENERATE on V9 member
- APAR PM13525

## Other ...

- DB2 for z/OS Exchange on DeveloperWorks
  - Relaunch and rebranding of ' DB2 zHotline'
  - It is a DB2 Technical Newsletter for DBAs and Systems Programmers
    - Critical APAR info (more than HIPER, less than Red Alert)
    - Living undocumented incompatible change list
    - Critical issues seen in field and ways to navigate
    - Hints, tips, practical advice
  - Will be updated monthly
  - RSS Feed from DeveloperWorks into IDUG
  - Link

<https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=22586cb0-8817-4d2c-ae74-0ddcc2a409bc>

## Summary

- Very good release in terms of the opportunities for price/performance and scalability improvement
  - Significant DBM1 31-bit VSCR after rebind
  - Use long term page fixed buffer pools
    - Exploit 1MB real storage page frames on z10 and z196
  - Reduced latch contention, log manager improvements, etc
  - Opportunity for further price performance improvements
    - More use of persistent threads
      - CICS, IMS/TM, High Performance DBATs
    - More use of RELEASE(DEALLOCATE) with persistent threads
    - More use of RELEASE(DEALLOCATE) is a trade off
      - Increased storage consumption
        - > Need to plan on additional real memory
      - Reduced concurrency
        - > BIND/REBIND and DDL
    - Increase MAXKEEPD to reduce short prepares for dynamic SQL
  - Opportunity for scale up and LPAR/DB2 consolidation

## Summary ...

- Carefully plan, provision and monitor real storage consumption
- Any customer migrating from either V8 or V9 to V10 should make a solid plan, take extra care to mitigate against the risk and set themselves up for success
  - Regular full ‘major’ maintenance drops
  - Exploitation of CST/RSU recommended maintenance
  - Augment by regular use of Enhanced HOLDDATA
  - Perform application regression and stress testing to keep ‘fires’ away from production
  - Plan should allow some contingency for some ‘bumps in the road’

# Questions





## IMz 2012 WW Technical Sales Boot Camp

Silicon Valley Labs, San Jose, CA

Thank  
YOU



*Click for Session Library*