



# DB2 10 for z/OS Migration Planning and Very Early Experiences

John Campbell

[campbelj@uk.ibm.com](mailto:campbelj@uk.ibm.com)



12/06/2011

© 2011 IBM Corporation

## 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

- Introduction
- Highlights
- Performance and Scalability
- Availability
- Other
- Migration and Planning
- Security considerations when removing DDF Private Protocol
- Items planned for post-GA delivery
- Summary

## DB2 10 for z/OS Beta

- Announce: February 9th, 2010
- Shipped: March 12th, 2010
- Largest Beta Ever
  - Strong customer demand
  - 24 WW customers/cross industry
  - Extended beta started 3Q
  - 73 parties in vendor program
- Customer Focus Areas
  - Regression testing
  - Out-of-box performance
  - Additional performance
  - Scalability
  - New function

The screenshot shows the IBM website's announcement for DB2 10 for z/OS Beta. The page layout includes a top navigation bar with links for Home, Solutions, Services, Products, Support & downloads, and My IBM. A search box is located on the right side of the navigation bar. The main content area is titled "Announcing DB2 10 for z/OS Beta" and features a large banner with the text "Introducing DB2 10 for z/OS Beta". Below the banner, there is a paragraph of text describing the product's benefits, followed by a section titled "Out-of-the-box Savings by improving operational efficiencies" which details performance improvements. A sidebar on the left contains a "DB2 for z/OS Family" menu with links to Library, Technical resources, Success stories, News, How to buy, Events, Training and certification, Services, and Support. A "Related links" section is also present in the sidebar. On the right side, there is a "DB2 10 for z/OS Highlights" section with a sub-section titled "DB2 10 for z/OS Cut Costs & Improve Performance" and a list of links for announcements, data warehousing, and downloads.

# Highlights

- Good Results
  - DBM1 31-bit virtual storage constraint relief
  - Insert performance
  - Hash Access good when hitting smaller than expected sweet spot in terms of use case
  - Complex queries
  - Inline LOBs (SLOBs)
  - Latch contention reduction
  - Quality of problems and issues found
  - Improved reliability and confidence as program progressed

## Highlights ...

- Mixed Results
  - OLTP performance, mostly good, some bad
  - Single thread BIND/REBIND performance
  - DDL concurrency
  - Access path lockdown (APREUSE, APCOMPARE, ...)

## Highlights ...

- Mainly positive customer experience and feedback about the program
- Majority of customers planning to start migration to V10 in 2011
- Incremental improvement over V8 and V9 programs
- No single voice / messages across the customers
- Need to appreciate that it is hard for customers to sustain effort over 6 period based on business and technical priorities
  - People / hardware resources and time are constrained
- Significant variation in terms of customer commitment and achievement
  - Subset of customers did a very good job on regression and new function testing
    - Good give back
  - Other customers
    - Limited qualification about what they were going to do
    - Limited qualification about what they did and what they achieved
- No customers in true business production by end of program
- Need to appreciate difference between QPP/Beta vs. ESP

## Highlights ...

- 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



## Performance and Scalability

- Plan on additional **10-30% real memory** (estimate)
- Many traditional OLTP workloads saw 5-10% CPU reduction in CM mode after (some more, some less)
- Prerequisites
  - REBIND packages to generate new SQL run time
  - Use PGFIX=YES on buffer pools to exploit 1MB real storage frames available on z10 and z196 (100% backed)
- But there were some exceptions < 5% CPU savings for OLTP with very light transaction, skinny packages with few simple SQL
  - Package allocation cost overrides benefit from SQL optimizations
  - APAR PM31614 may solve this by improving package allocation performance
  - Use of persistent threads with `RELEASE(DEALLOCATE)` will compensate

# Performance and Scalability

- Query Performance enhancements
  - No REBIND required for
    - Index list prefetch
    - INSERT index 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

- High Performance DBATs (Hi-Perf DBATs) – 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
    - 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 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 ...

- Customers measurements
  - Not always consistent and repeatable
  - Wide variation on measurement noise especially elapsed time
  - In most cases not running in a dedicated environment or scale/size of production
  - Many cases running subset of production workload
  - Sometimes use of synthetic workload to study specific enhancements
  - Do not trust some of the very big numbers on CPU and especially elapsed time savings
  - Recommendation: customers should not spend the savings until they see them in production

## 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 -> investigating
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 ...

- Use of 1MB real storage page frames on z10 and z196
  - Potential for reduced for CPU through less TLB misses
  - Buffer pools must be defined as PGFIX=YES
  - Buffer pool page fix introduced in V8 to reduce CPU
  - Many customers reluctant to use PGFIX=YES because of potential for real storage
    - Running too close to the edge of the amount of real storage provisioned
    - Understand the value but only applies for 1-2 hours per 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 ...

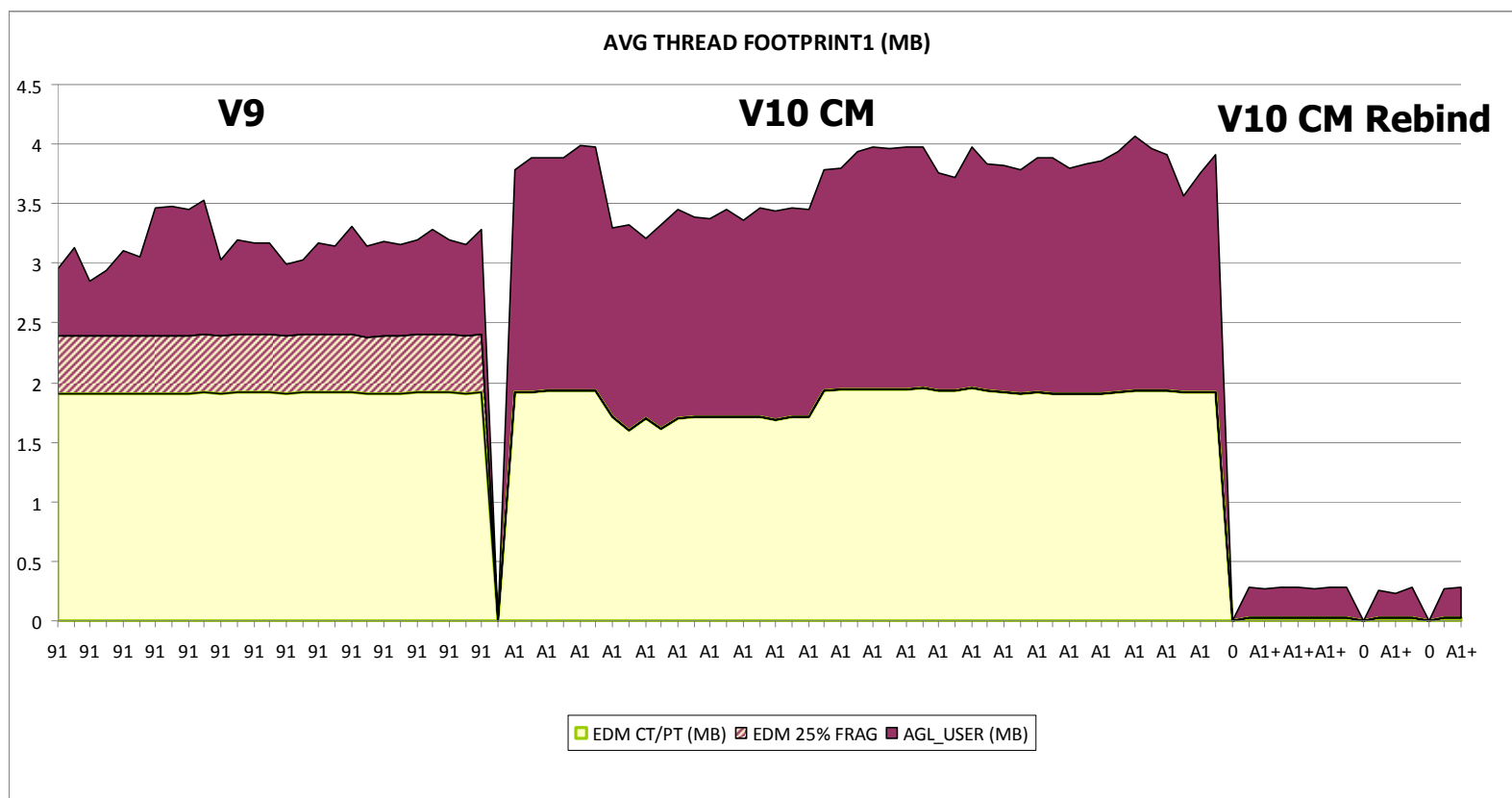
- Use of 1MB real storage page frames on z10 and z196 ...
  - Must partition real storage between 4K frames and 1MB frames
    - Specified by LFAREA xx% in IESYSnn parmlib member and only changeable by IPL
    - 1MB frames are non-pageable
    - If 1MB page frames are overcommitted, will use 4K page frames
    - Recommendation: to add 20% in size to allow for growth and tuning
  - Be careful
    - Make sure critical z/OS maintenance applied before using 1MB pages
  - Benefit based on customer experience 0 to 6% reduced CPU
  - Requirement for new parameter to separate use of PGFIX=YES from use of 1MB page size
    - Plan to address in the next release of DB2

## 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
  - 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
    - ESQA/ECSA (31-bit) storage
    - Active log write

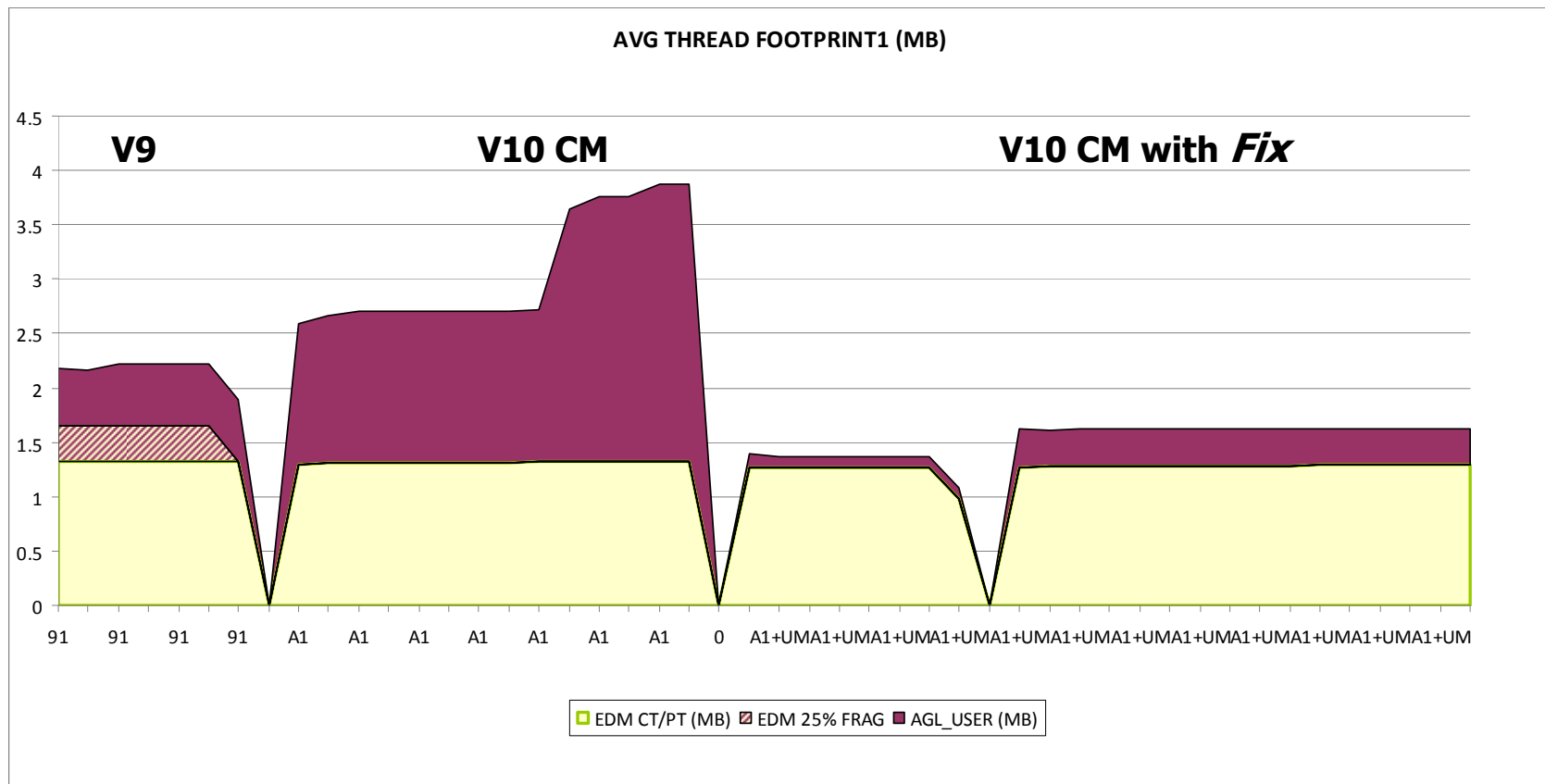
# Performance and Scalability ...

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



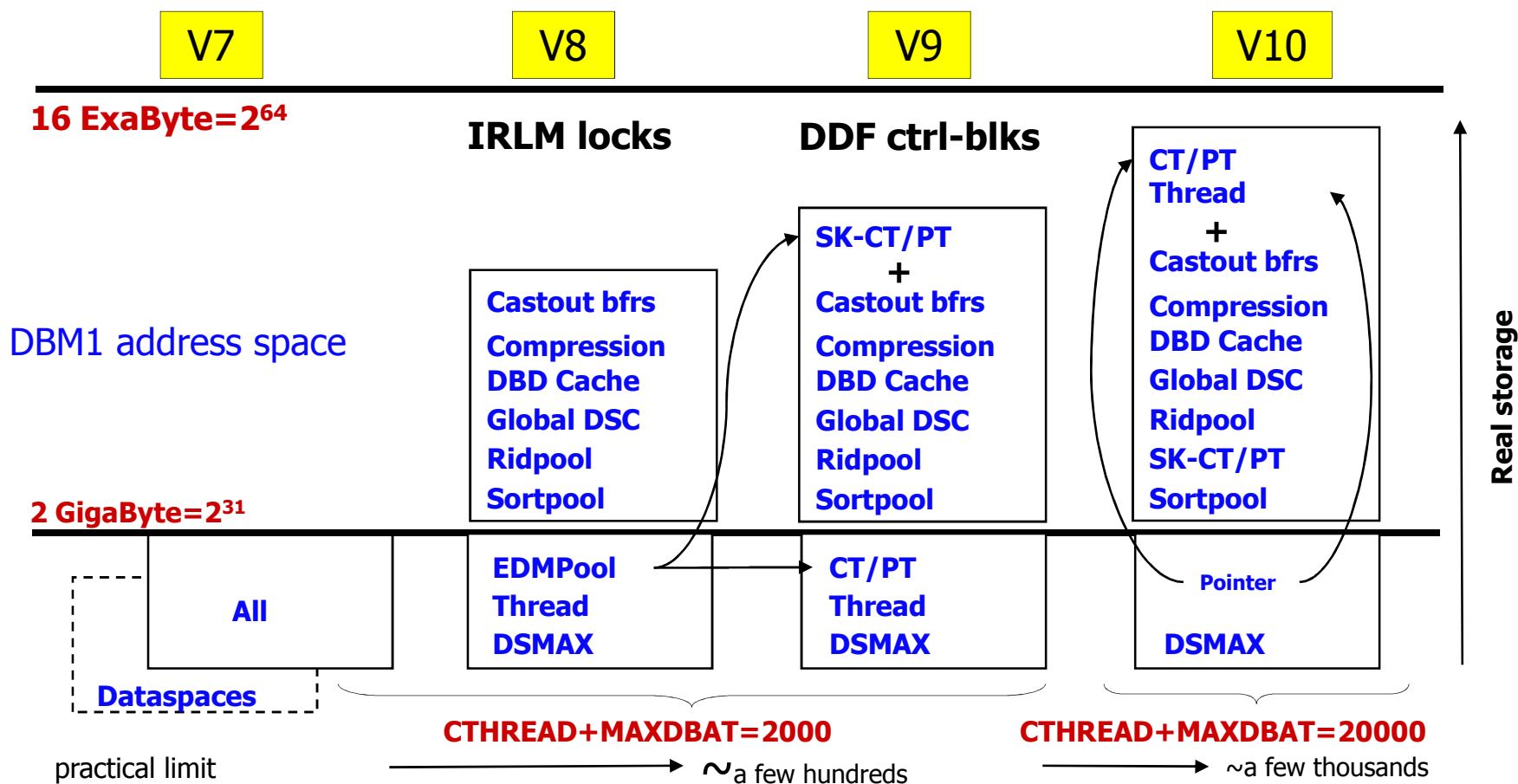
# Performance and Scalability ...

- 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



## Performance and Scalability ...

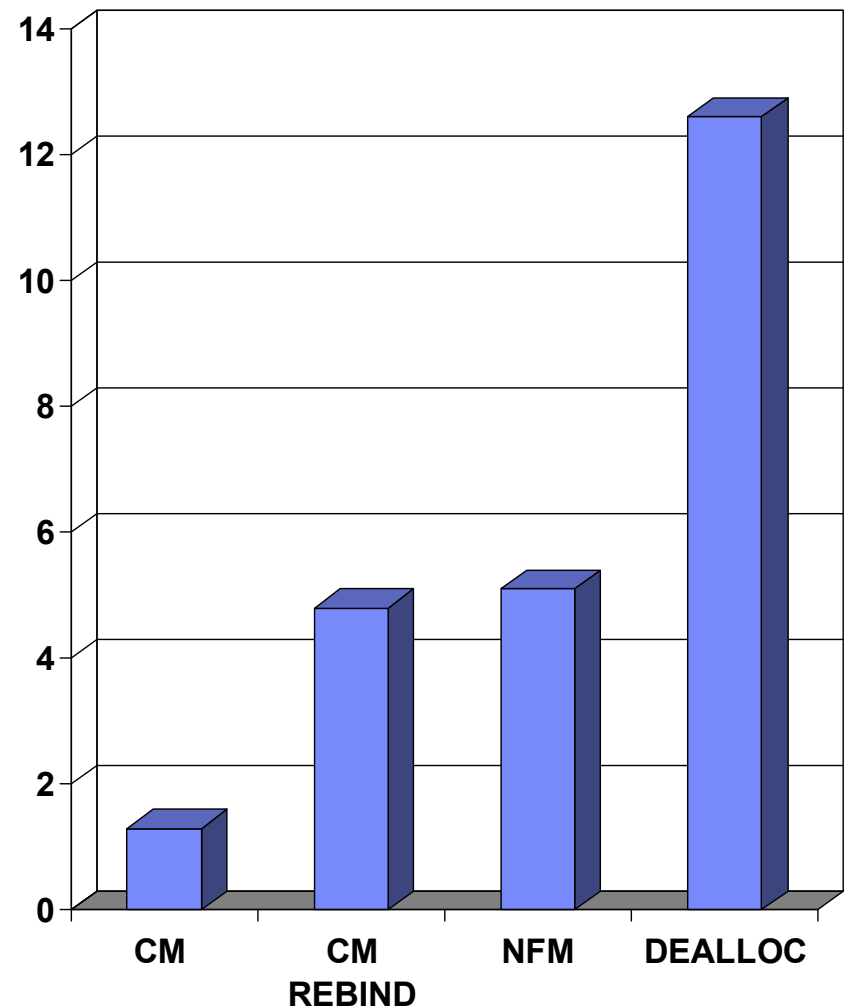
- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time ...
  - Major customer opportunities here for 31-bit VSCR and improved price/performance
  - Potential to reduce legacy OLTP transaction CPU cost through use of
    - More CICS protected ENTRY (persistent) threads
    - More use of RELEASE(DEALLOCATE) with persistent threads
    - Must provision additional real storage to back the requirement
  - 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
      - Will need additional real storage to support increased number of threads

## 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
  - 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
  - For RELEASE(DEALLOCATE) some locks are held beyond commit until thread termination
    - Mass delete locks (SQL 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

## 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 ...

- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time ...
  - Potential to reduce the number of DB2 subsystems
    - Collapse multiple DB2 members running on the same LPAR
    - Reduce the total number of DB2 members
    - May be able to reduce the number of LPARs
    - Consider the increase of logging rate per DB2 member
    - 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 RRSF 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-emphasize 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
      - Almost all the DB2 storage 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 ...

- 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 to set to 2x normal DB2 subsystem usage
      - Kills the DB2 subsystem when SPRMRSMX value reached
    - With V10, will need to now 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
      - V10 makes extensive use of 64-bit shared
    - LPAR level instrumentation buckets for REAL and AUX storage use
      - If more the 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 and will probably close in June
    - 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
    - Will also introduce 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 try to 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 ...

- 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 ...

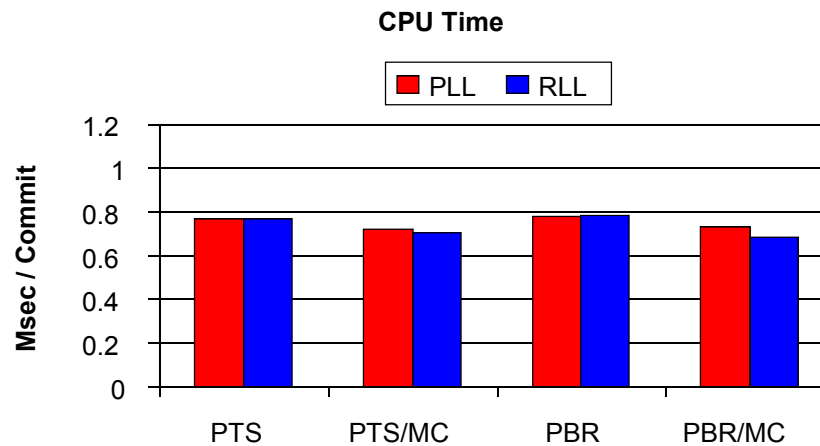
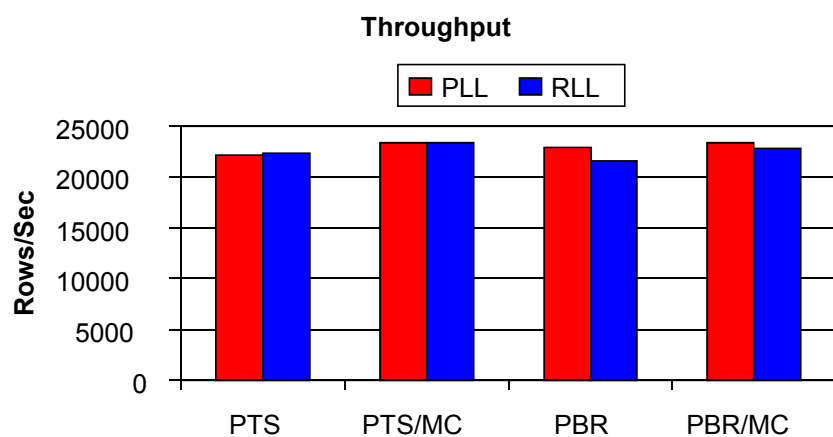
- High INSERT performance ...
  - Reduced LRSN spin for inserts to the same page
    - Works well for MRI and INSERT within loop in a data sharing environment
  - Optimization for 'pocket' sequential insert works well
    - Index manager picks the candidate RID during sequential insert (next lowest key 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 offload to zIIP to compensate

# High Insert Workload Description

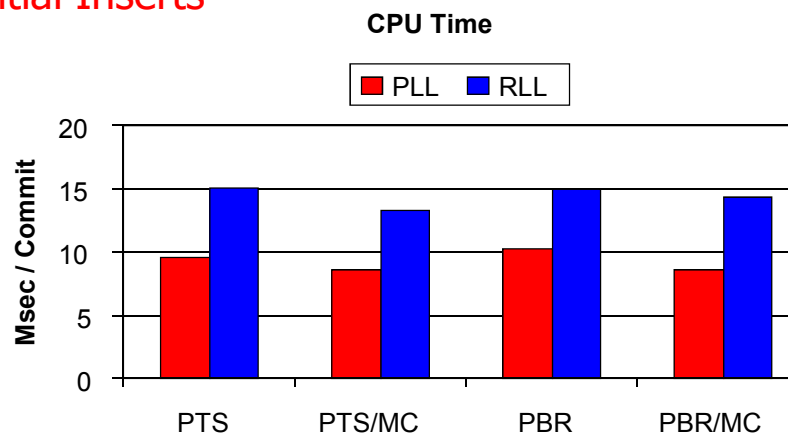
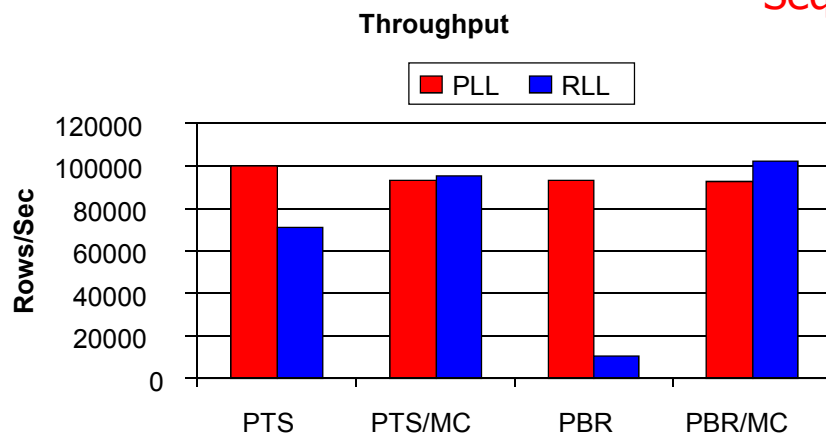
- 2-way data sharing
- Database schema
  - 3 tables with total of 6 indexes (4 unique, 2 non-unique indexes, 2 secondary indexes)
  - Table space types: Classic Partitioned, Classic Segmented, UTS (PBR, PBG)
- SQL
  - INSERTs contain 5, 9 and 46 columns of integer, bigint, char, varchar, decimal and 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 -----



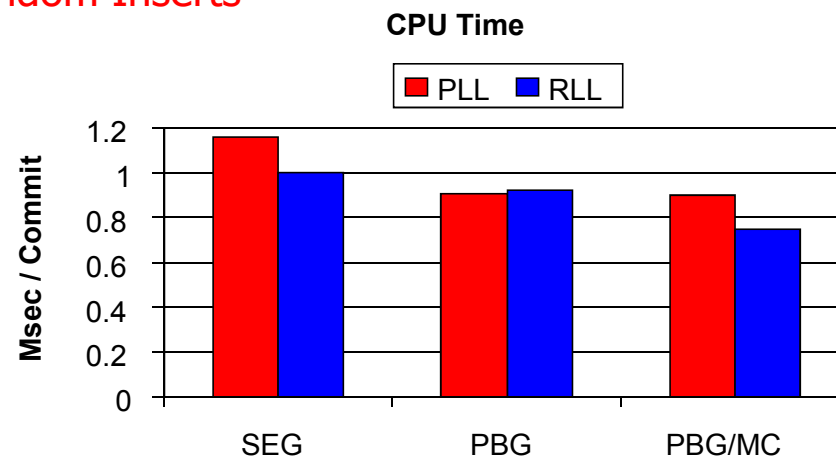
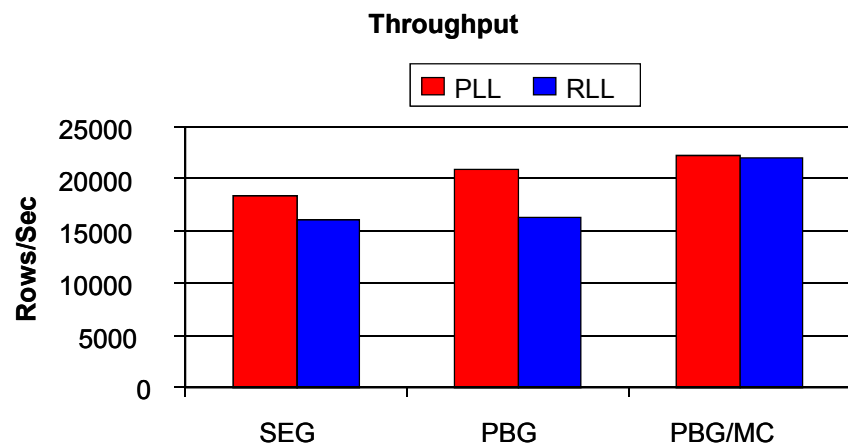
## ----- Sequential Inserts -----



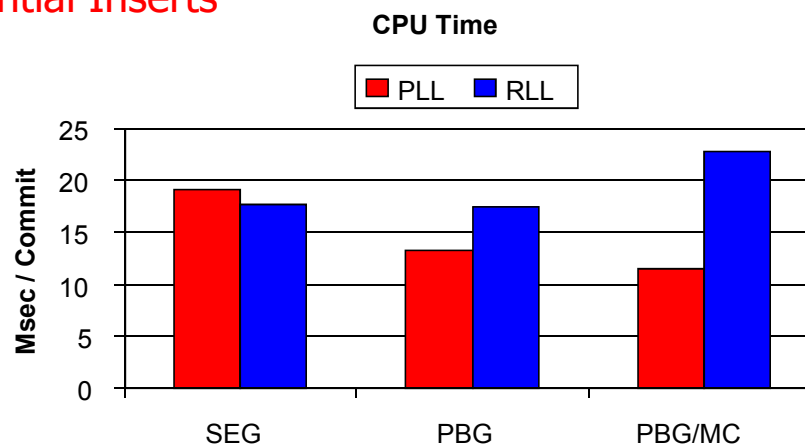
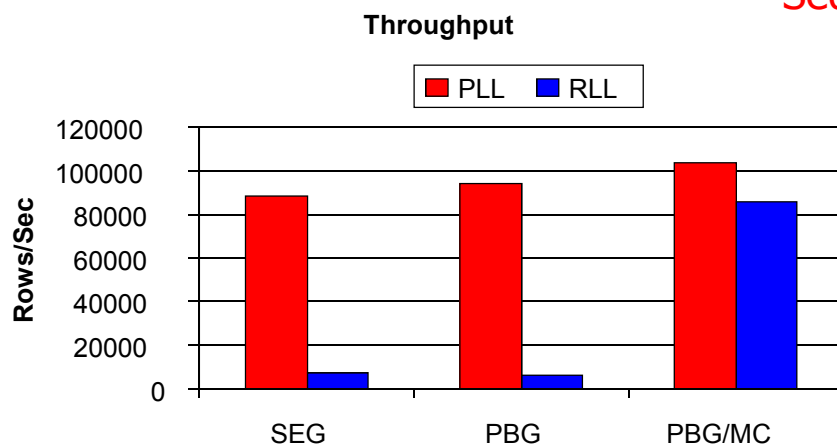


# DB2 10 Non-range Defined Table Spaces

## 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
  - 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 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
  - Competes against index only access
    - 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

## Availability

- Online Schema Evolution ('Deferred Alter')
  - Note: UTS is pre-requisite for Hash, Inline LOB, Currently Committed
  - Migrate from classic table space types (simple, segmented, partitioned) to UTS PBG/PBR
    - One way ticket only
  - Once migrated to UTS PBG/PBR can change attributes such as
    - 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

- Online REORG with FORCE
  - Customers perceive limited value
  - Only running threads which are blocking are cancelled
  - Threads which are suspended / inactive will cause REORG to still fail
- Online REORG LOB with DISCARD
  - Customers perceive limited value
  - Cannot handle LOB columns greater than 32KB

## 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)
- 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 ...

- Single thread BIND/REBIND
  - Degraded CPU and elapsed time performance on entry 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 in data sharing mode
  - 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 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



## Other ...

- Value of REBIND under V10
  - Performance from new run time (avoid puffing, enable SPROC)
  - Maximize DBM1 31-bit VSCR
  - Reduce exposure to problems with migrated packages from earlier releases
- How to 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
  - Use APREUSE and APCOMPARE (see next slide)

## Other ...

- Access Path Reuse - APREUSE and APCOMPARE
  - Introduced in V10 through APARs PM33767 and PM25679
  - 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
    - Applies to BIND REPLACE and REBIND
    - Not guaranteed to succeed in all cases
  - Applies the first time under V10 only to customers who have migrated from V9, and performed a BIND/REBIND on V9

## 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
```

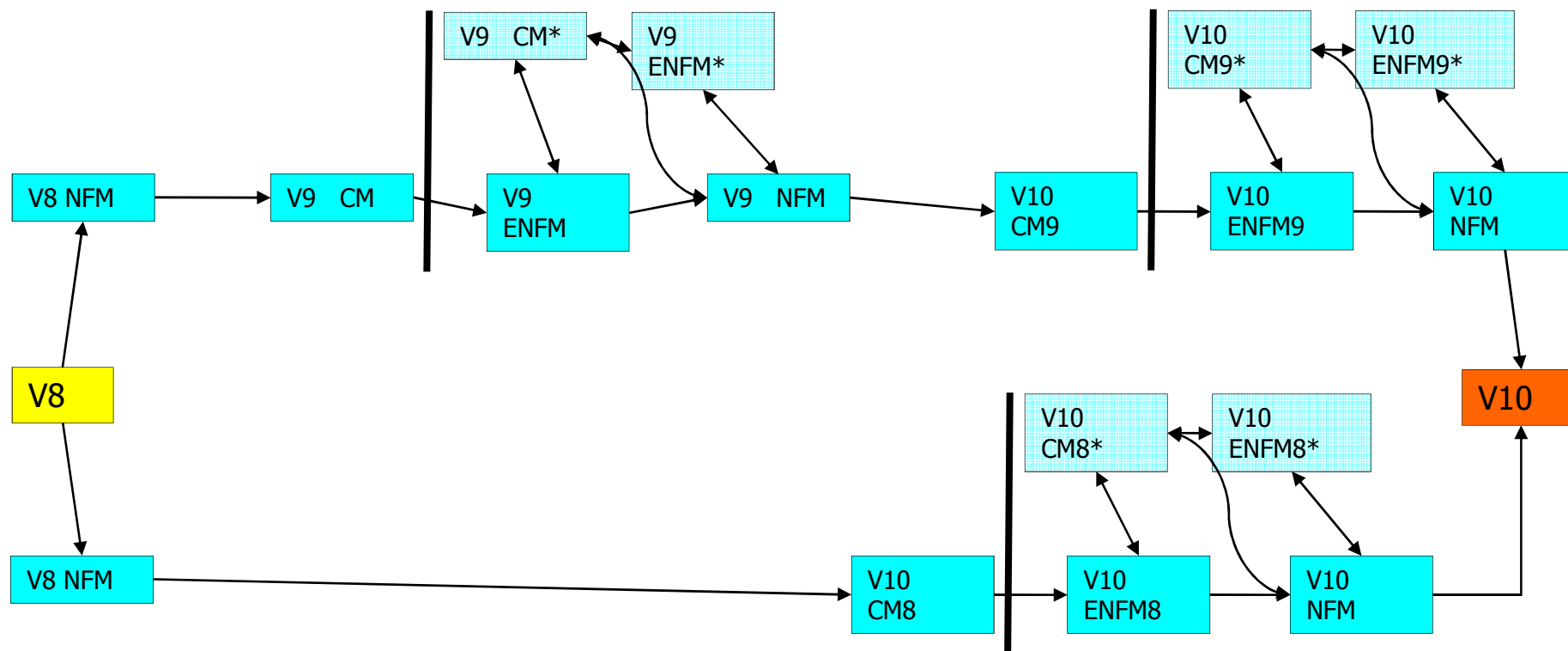
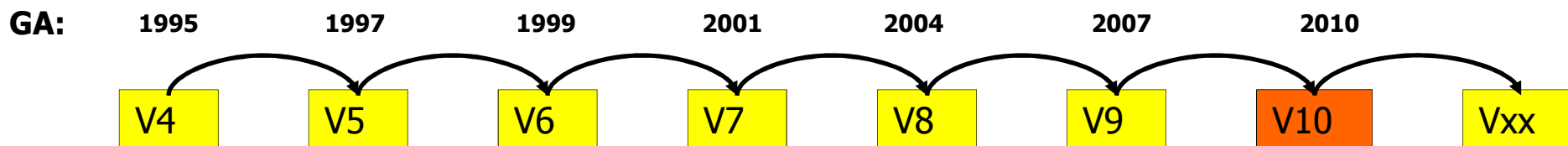
## Other ...

- 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



## 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
- Cannot migrate
  - 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\\_migr8cm8checklist.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z_migr8cm8checklist.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\\_migr9cm9checklist.htm](http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.inst/src/tpc/db2z_migr9cm9checklist.htm)



## Migration and Planning ...

- If coming from V8
  - BSDS must be reformatted for larger active / archive tracking
- DB2 Connect
  - Minimum level
    - V9.1 FP1 (recommended is FP7)
    - V9.5 (recommended is FP4)
  - Level for new functions
    - V9.7 FP3A
- 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 ...

- 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
- Old plans and packages bound prior to V6 will require REBIND
- Catalog and Directory must be SMS managed (EF, EA) ahead of ENFM
- PDSEs required for SDSNLOAD, SDSNLOD2, ADSNLOAD
- DSNHDECP NEWFUN=V10|V9|V8

## Migration and Planning ...

- Must absolutely eliminate all use of DDF Private Protocol before migrating
  - No longer supported In V10
  - Local packages miss tagged with DDF Private Protocol will be tolerated
  - Otherwise package must exist in both local and remote site
  - 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
  - Since V8, APAR PK92339 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
    - 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)
  - Since V8, APAR PK64045 (PE, PM14816) adds 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

## Migration and Planning ...

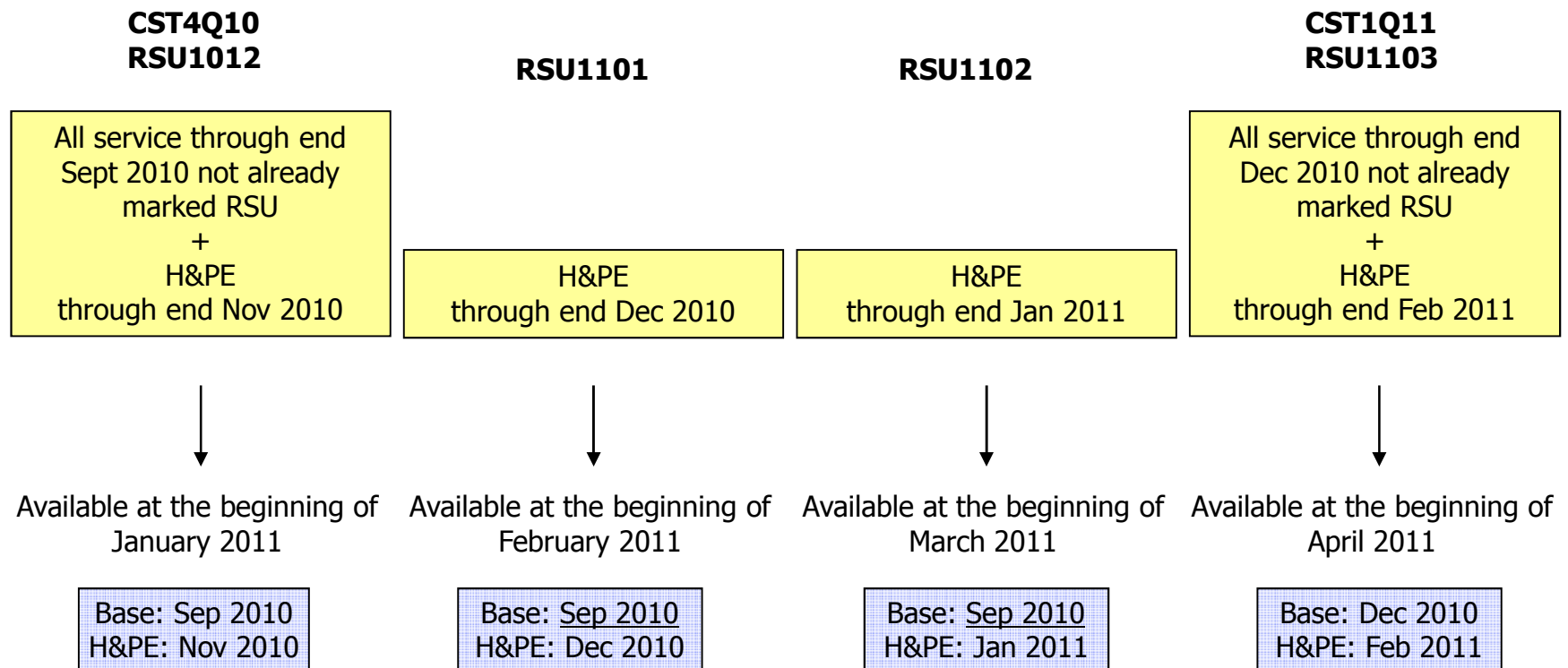
- EXPLAIN tables
  - Format and CCSID from previous releases is deprecated in V10
    - Cannot use pre V8 format
      - SQLCODE -20008
    - V8 or V9 format
      - Warning SQLCODE +20520 regardless of CCSID EBCDIC or UNICODE
    - Must not use CCSID EBCDIC with V10 format
      - EXPLAIN fails with 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

## Migration and Planning ...

- Early customer adopters of V10 migrating from either V8 or V9 should make plans and take extra care to mitigate against the risk of instability
  - Perform application regression and stress testing to keep ‘fires’ away from production
  - Need to be more aggressive on planned continual application of preventative service
    - Will have to stay a lot more current than 2 full ‘major’ drops a year
    - Regular full ‘major’ maintenance drops including HIPERs/PEs essential required for the first year or so
      - May be 4 ‘major’ drops in the first year
    - Can move to 2 ‘major’ and 2 ‘minor’ maintenance drops as the release passes the early adopter curve
    - Exploit CST/RSU recommended maintenance as opposed to the PUT route
      - Recommended maintenance after successful testing for a least one month
      - Testing performed over and above that performed by DB2 Development
      - CST testing still does not replace customer regression/stress testing
  - Must be prepared to tolerate hit some ‘bumps in the road’
  - Customer who are not prepared to take mitigating actions and have no tolerance for ‘bumps in the road’ should not be early adopters
    - V8 customers should migrate to V9 quickly as it is relatively stable
  - Answer depends on your risk profile and then setting yourself up for best chance for success

# Migration and Planning ...

- CST and RSU example



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

## Security considerations when removing DDF Private Protocol

- 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
- Until DB2 10 - 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 ...

- Now APAR PM37300 provides control over the authorization checks performed when migrating from private protocol to DRDA protocol
- DB2 10 private protocol security semantics are no longer used as default for access from a DB2 for z/OS requester
- DB2 V8 and V9 uses DRDA authorization checks when private protocol is disabled by
  - Setting system parameter DSN6FAC PRIVATE\_PROTOCOL=NO



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

- Before disabling private protocol, ensure all appropriate grants are performed by
  - 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 clients
- DB2 V8 and V9 can disable private protocol but maintain private protocol authorization checks by
  - Setting system parameter `DSN6FAC 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 `DSN6FAC PRIVATE_PROTOCOL=AUTH`

## Items Planned for post-GA Delivery via APAR

- **APREUSE, APCOMPARE (PM33767, PM25679)**
- Delete data sharing member (PM31009)
- Enhancements for new DBA authorities (PM28296)
  - Prevent privileged users from stopping audit traces
  - No implicit system privileges for DBADM
- Inline LOBs for SPT01 (PM27811)
  - Compression, BIND performance
- Online REORG concurrency for materializing deferred ALTERs (PM25648)
- Temporal enhancements
  - TIMESTAMP WITH TIMEZONE support (PM31314)
  - Enhancement for data replication (PM31315)
  - ALTER ADD COLUMN, propagate to history table (PM31313)

## Items Planned for post-GA Delivery via APAR ...

- New system profile filters based on “client info” fields (PM28500)
  - Three new columns for userid, appname, and workstation
  - Wildcard support: if column is ‘\*’ then all threads pass that qualification
- Zparm to force deletion of CF structures on group restart (PM28925)
- Relief for incompatible change in CHAR of decimal data (PM29124)
- **Real storage monitoring and contraction enhancements (PM24723)**
- Hash LOAD performance (PM31214)
- DSSIZE > 64GB – APAR coming
- REORG REBALANCE SHRLEVEL CHANGE – APAR coming

## RSM APAR OA35885 vital for DB2 10

- Strong advice not to go into major production environment without the PTF and corresponding DB2 support in the PTF for **APAR PM24723**
- Enables proper monitoring of REAL storage usage by providing DB2 with 64-bit memory object statistics
- Provides some protection against the system paging or running out of AUX storage

## Summary

- Very good release in terms of the opportunities for price/performance and scalability improvements
  - 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
  - Opportunity for scale up and LPAR/DB2 consolidation

## Summary ...

- Carefully plan, provision and monitor real storage consumption
- Early customer adopters of V10 migrating from either V8 or V9 should make plans and take extra care to mitigate against the risk of instability
  - Regular full ‘major’ maintenance drops
  - Exploitation of CST/RSU recommended maintenance
  - Perform application regression and stress testing to keep ‘fires’ away from production
  - Must be prepared to tolerate some ‘bumps in the road’
- Answer to should you migrate from V8 to V9 or directly to V10?
  - Depends on your risk profile
  - Setting yourself up for best chance of success

# Questions

