DB2 for z/OS Best Practices

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

John Campbell
Distinguished Engineer
DB2 for z/OS Development

CAMPBELJ@uk.ibm.com

Disclaimer/Trademarks

THE INFORMATION CONTAINED IN THIS DOCUMENT HAS NOT BEEN SUBMITTED TO ANY FORMAL IBM TEST AND IS DISTRIBUTED AS IS. THE USE OF THIS INFORMATION OR THE IMPLEMENTATION OF ANY OF THESE TECHNIQUES IS A CUSTOMER RESPONSIBILITY AND DEPENDS ON THE CUSTOMER'S ABILITY TO EVALUATE AND INTEGRATE THEM INTO THE CUSTOMER'S OPERATIONAL ENVIRONMENT. WHILE IBM MAY HAVE REVIEWED EACH ITEM FOR ACCURACY IN A SPECIFIC SITUATION, THERE IS NO GUARANTEE THAT THE SAME OR SIMILAR RESULTS WILL BE OBTAINED ELSEWHERE. ANYONE ATTEMPTING TO ADAPT THESE TECHNIQUES TO THEIR OWN ENVIRONMENTS DO SO AT THEIR OWN RISK.

ANY PERFORMANCE DATA CONTAINED IN THIS DOCUMENT WERE DETERMINED IN VARIOUS CONTROLLED LABORATORY ENVIRONMENTS AND ARE FOR REFERENCE PURPOSES ONLY. CUSTOMERS SHOULD NOT ADAPT THESE PERFORMANCE NUMBERS TO THEIR OWN ENVIRONMENTS AS SYSTEM PERFORMANCE STANDARDS. THE RESULTS THAT MAY BE OBTAINED IN OTHER OPERATING ENVIRONMENTS MAY VARY SIGNIFICANTLY. USERS OF THIS DOCUMENT SHOULD VERIFY THE APPLICABLE DATA FOR THEIR SPECIFIC ENVIRONMENT.

Objectives

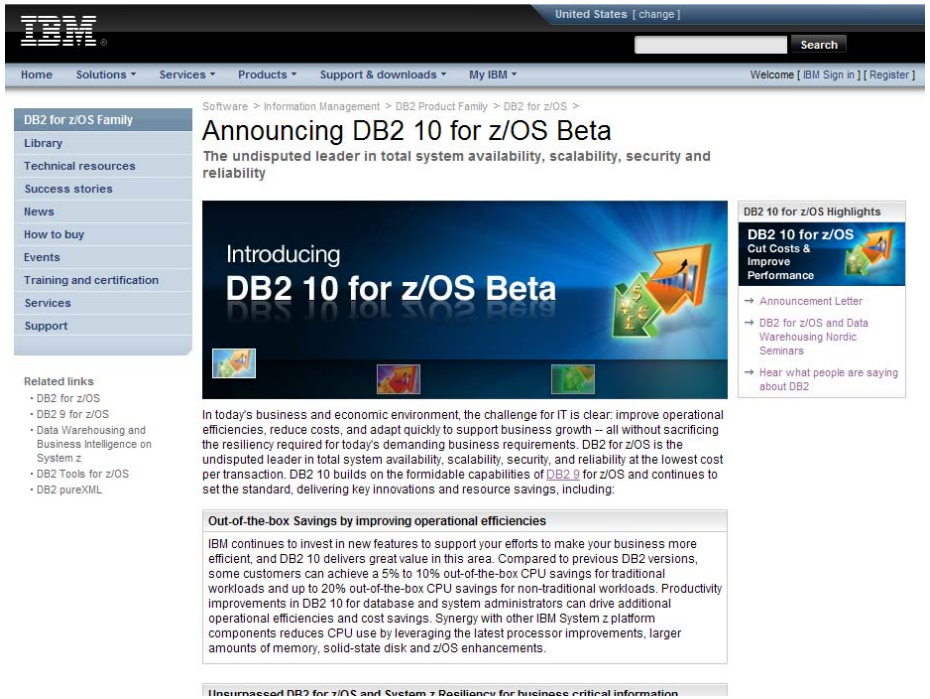
- Share lessons learned, surprises, pitfalls
- Provide hints and tips
- Address some myths
- Provide additional planning information
- Provide usage guidelines and positioning on new enhancements

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



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 month 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 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	20-30% CPU reduction in random access 16% CPU reduction comparing Hash Access and Index-data access. 5% CPU reduction comparing Hash against Index only access Further improvements delivered late in the beta program.

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 …

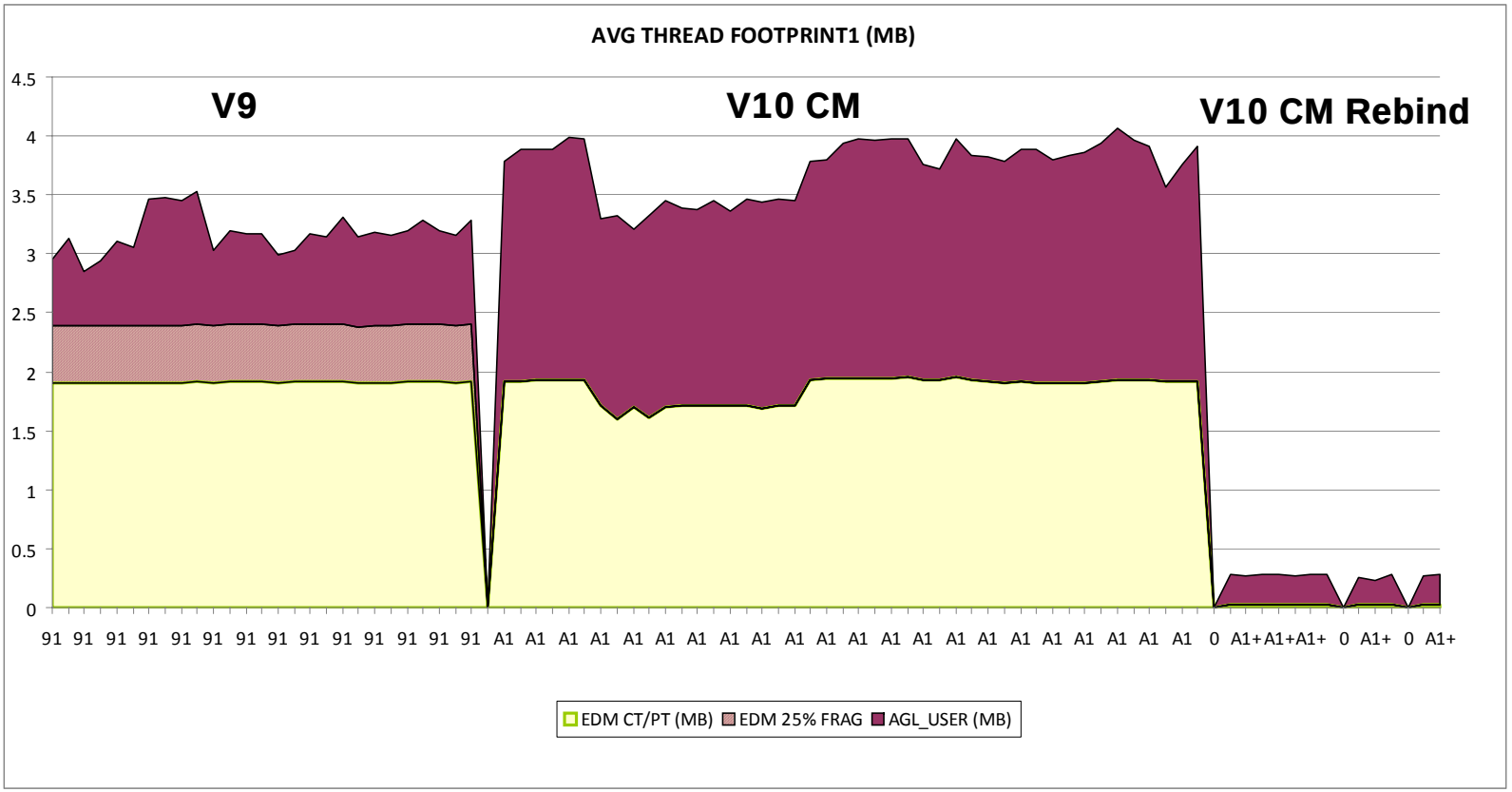
- Use of 1MB real storage page frames on z10 and z196 …
  - Must partition real storage between 4K frames and 1MB frames
    - Specified by LFAREA xx% in IESYSnn parmlib member and only changeable by IPL
    - 1MB frames are non-pageable
    - If 1MB page frames are overcommitted, will use 4K page frames
    - 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

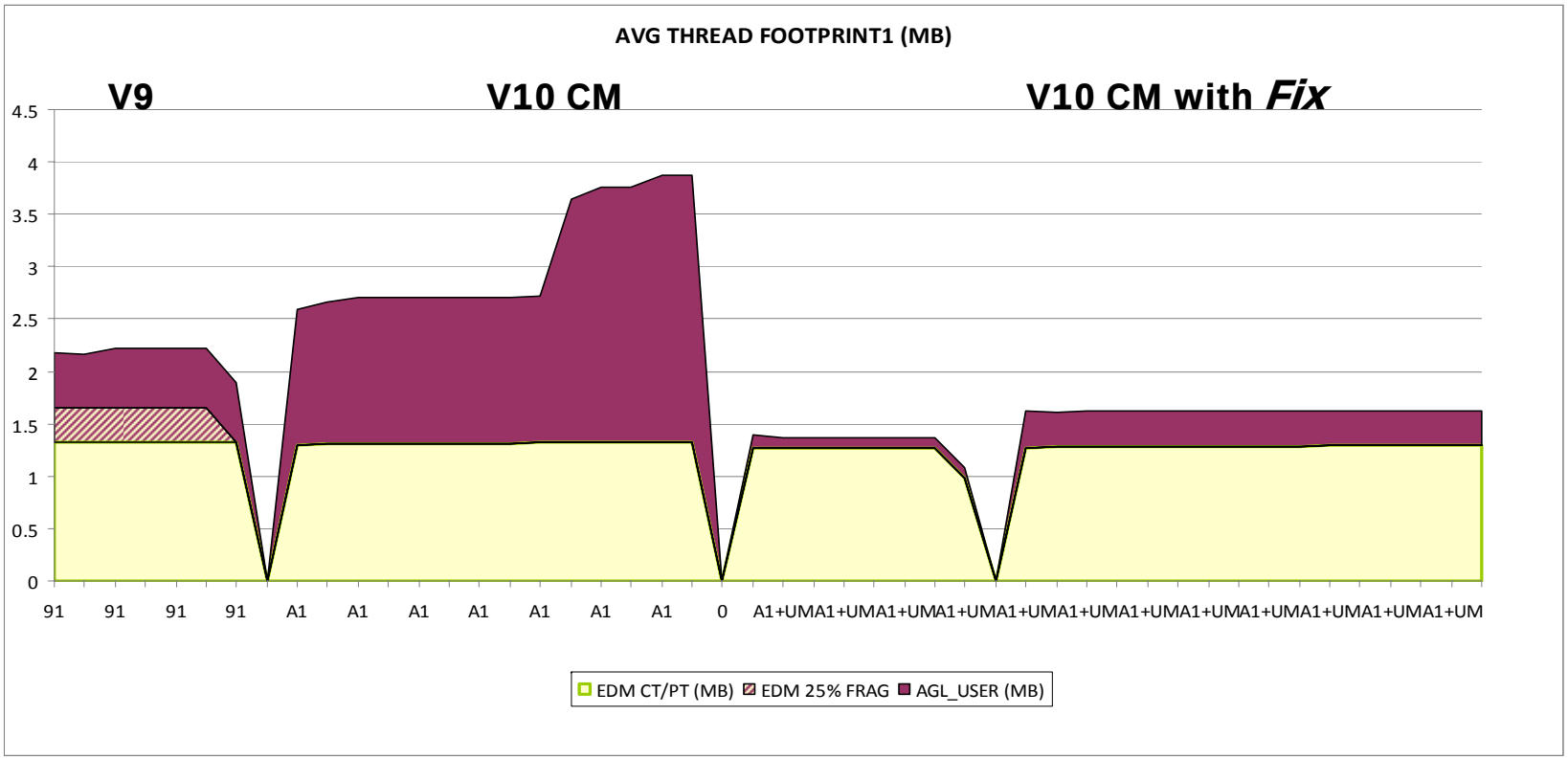
Performance and Scalability …

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



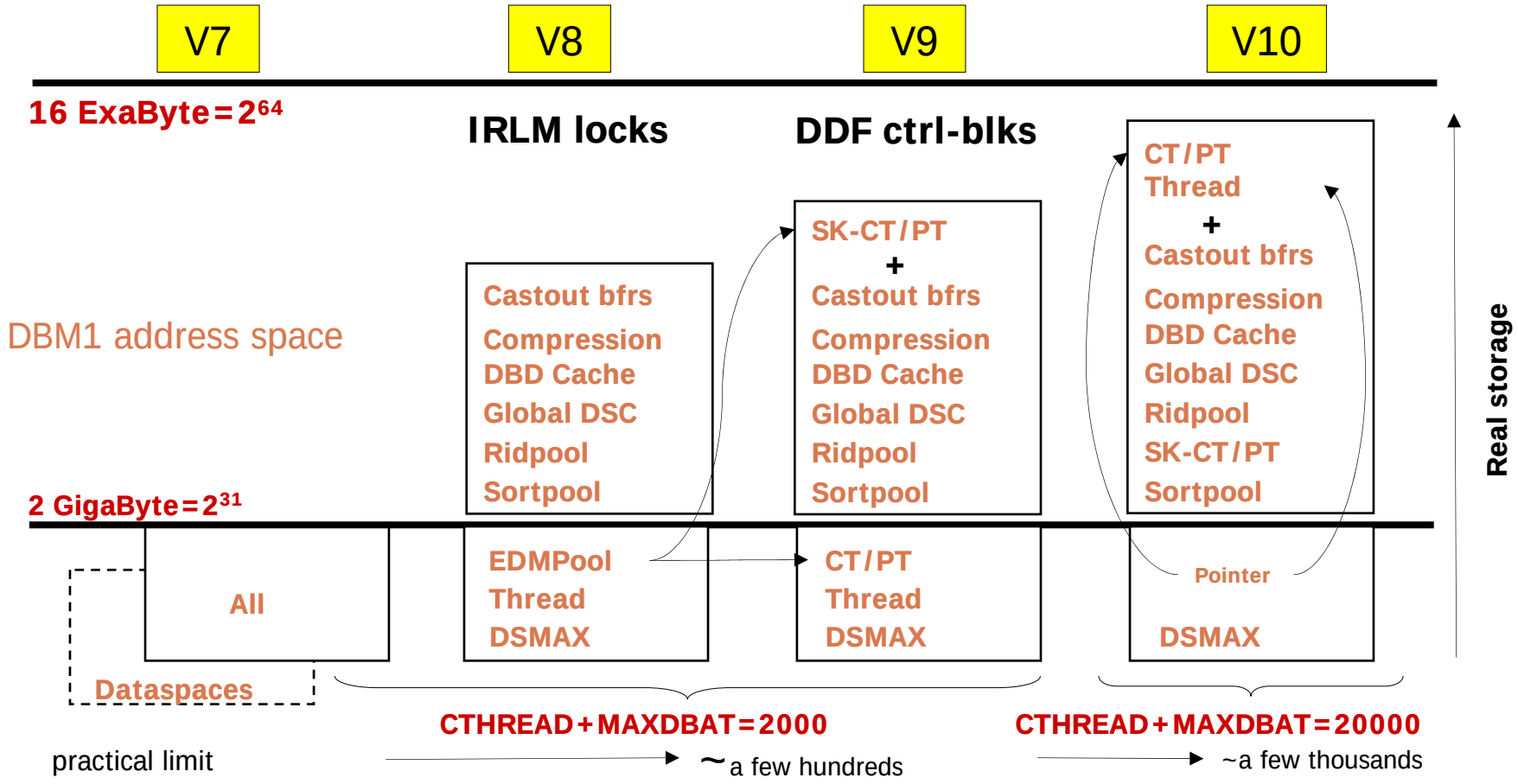
Performance and Scalability …

- 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 …

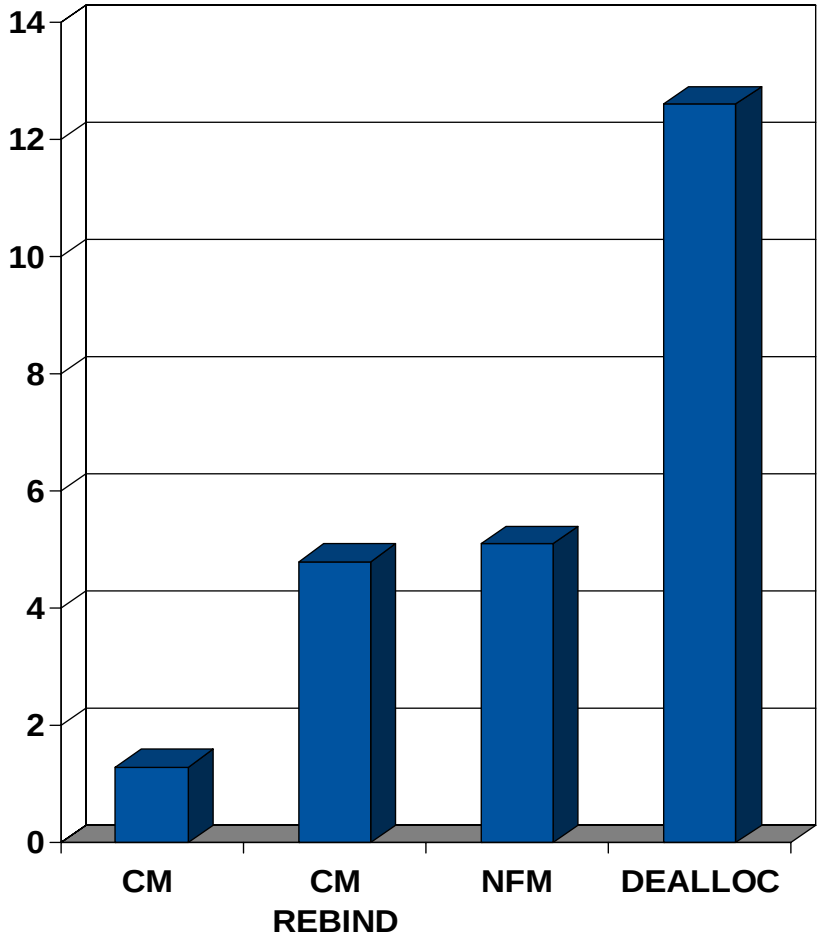
- DBM1 31-bit Virtual Storage Constraint Relief with 64-bit SQL run time …
  - Major customer opportunities here for 31-bit VSCR and improved price/performance
  - Potential to reduce legacy OLTP transaction CPU cost through use of
    - More CICS protected ENTRY (persistent) threads
    - More use of 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 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

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 RRSAF 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 …

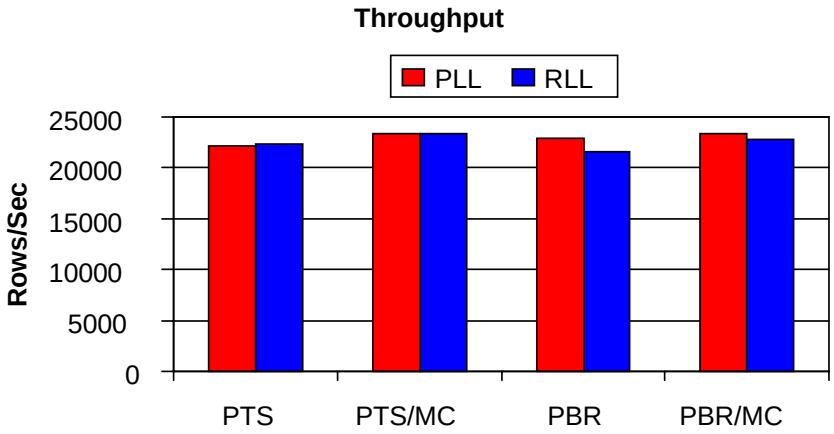
- High INSERT performance …
  - Reduced LRSN spin for inserts to the same page
    - Works well for MRI and INSERT within loop in a data sharing environment
  - Optimization for 'pocket' sequential insert works well
    - Index manager picks the candidate RID during sequential insert (next lowest key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

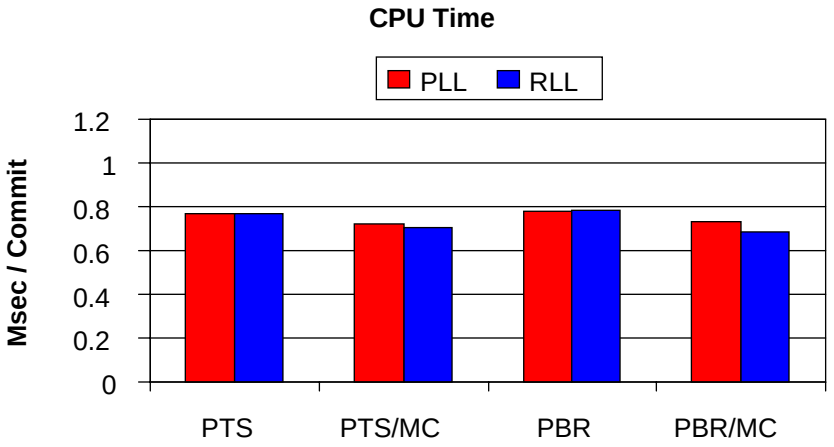
High Insert Workload Description

- 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

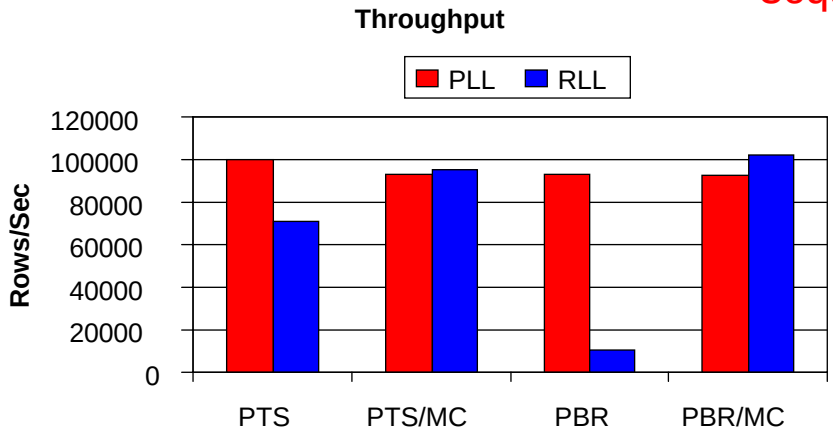
DB2 10 Range Defined Table Spaces

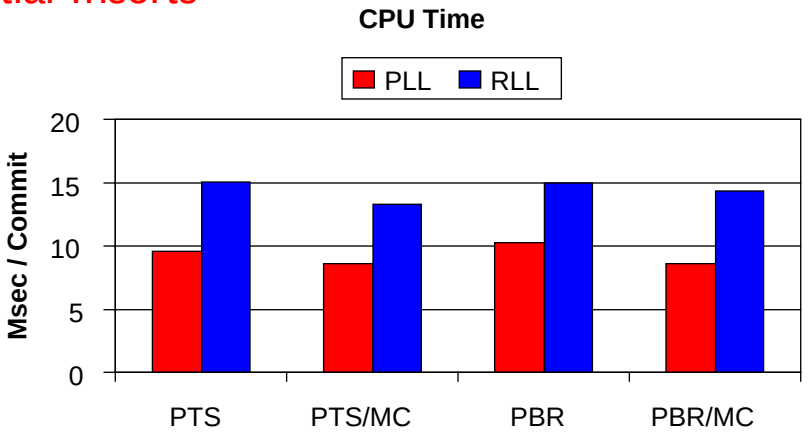
Random Inserts





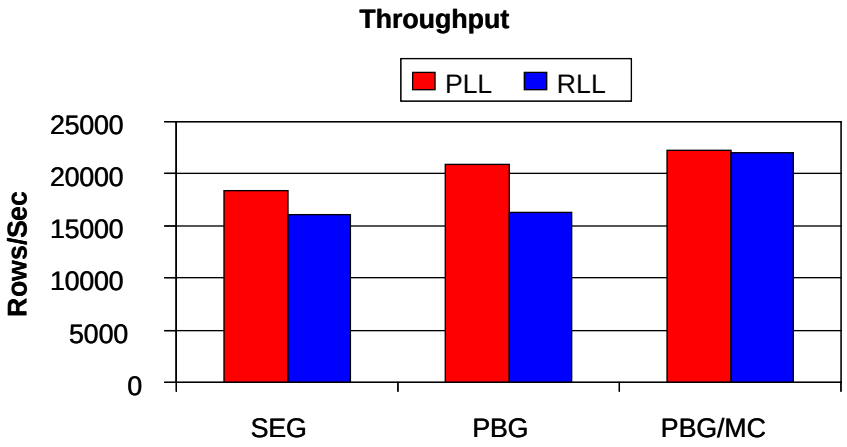
Sequential Inserts

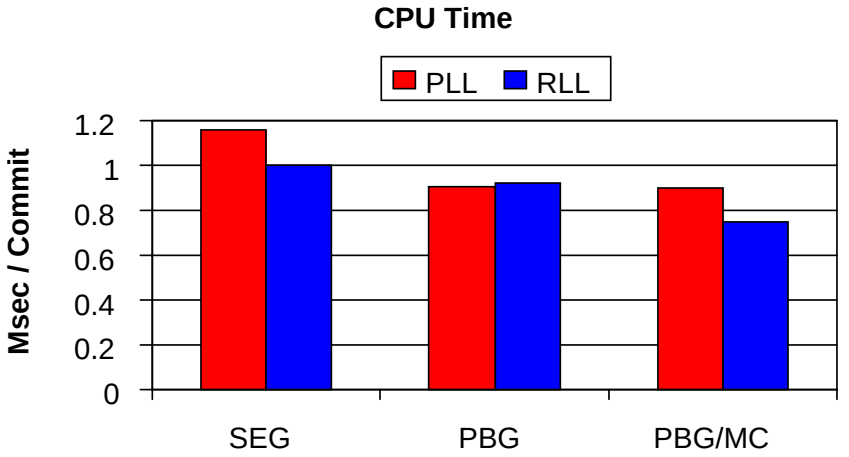




DB2 10 Non-range Defined Table Spaces

Random Inserts





Sequential Inserts