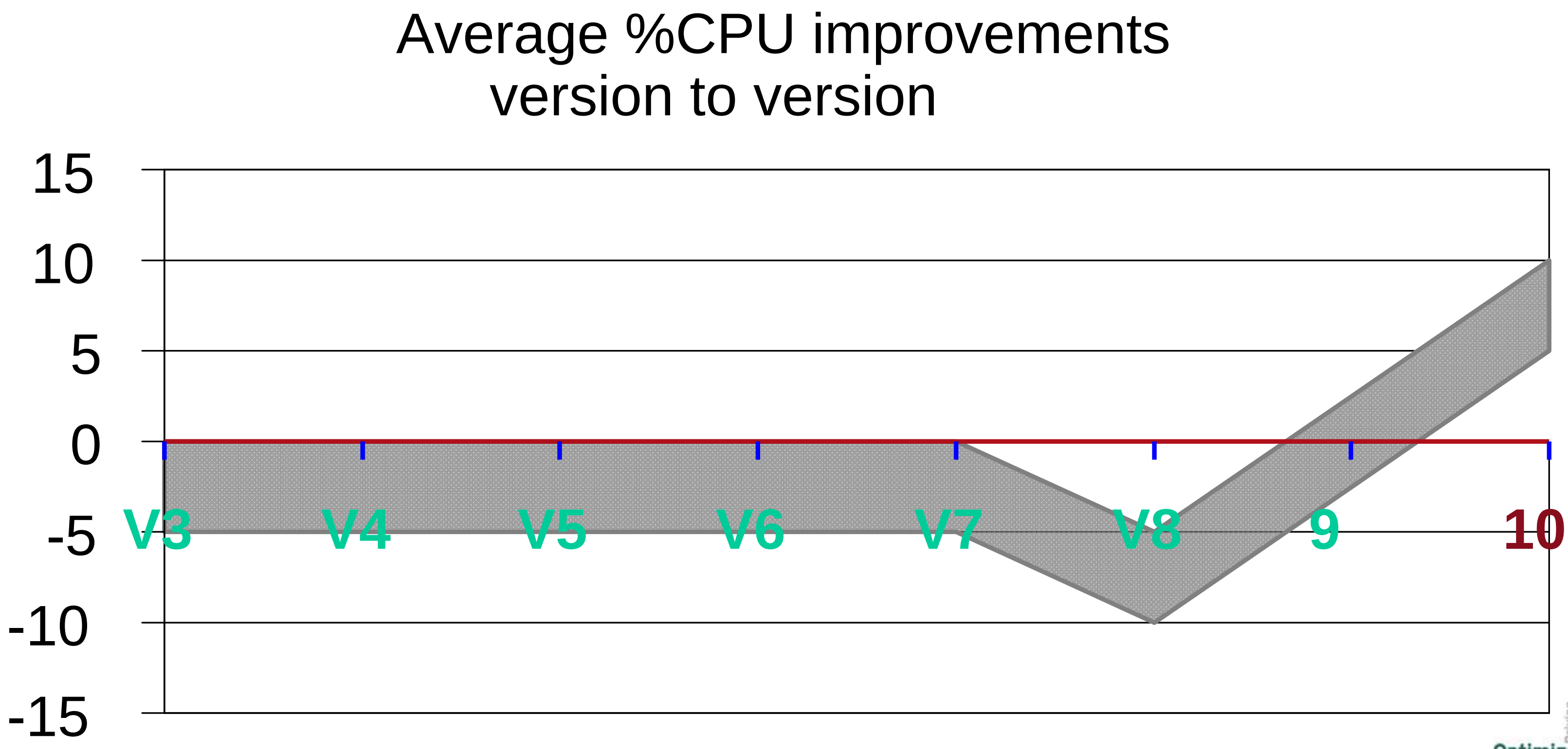
DB2 10 for z/OS Performance Topics

Tuesday, March 1, 2011: 11:00 AM-12:00 PM
Room 211A (Anaheim Convention Center)

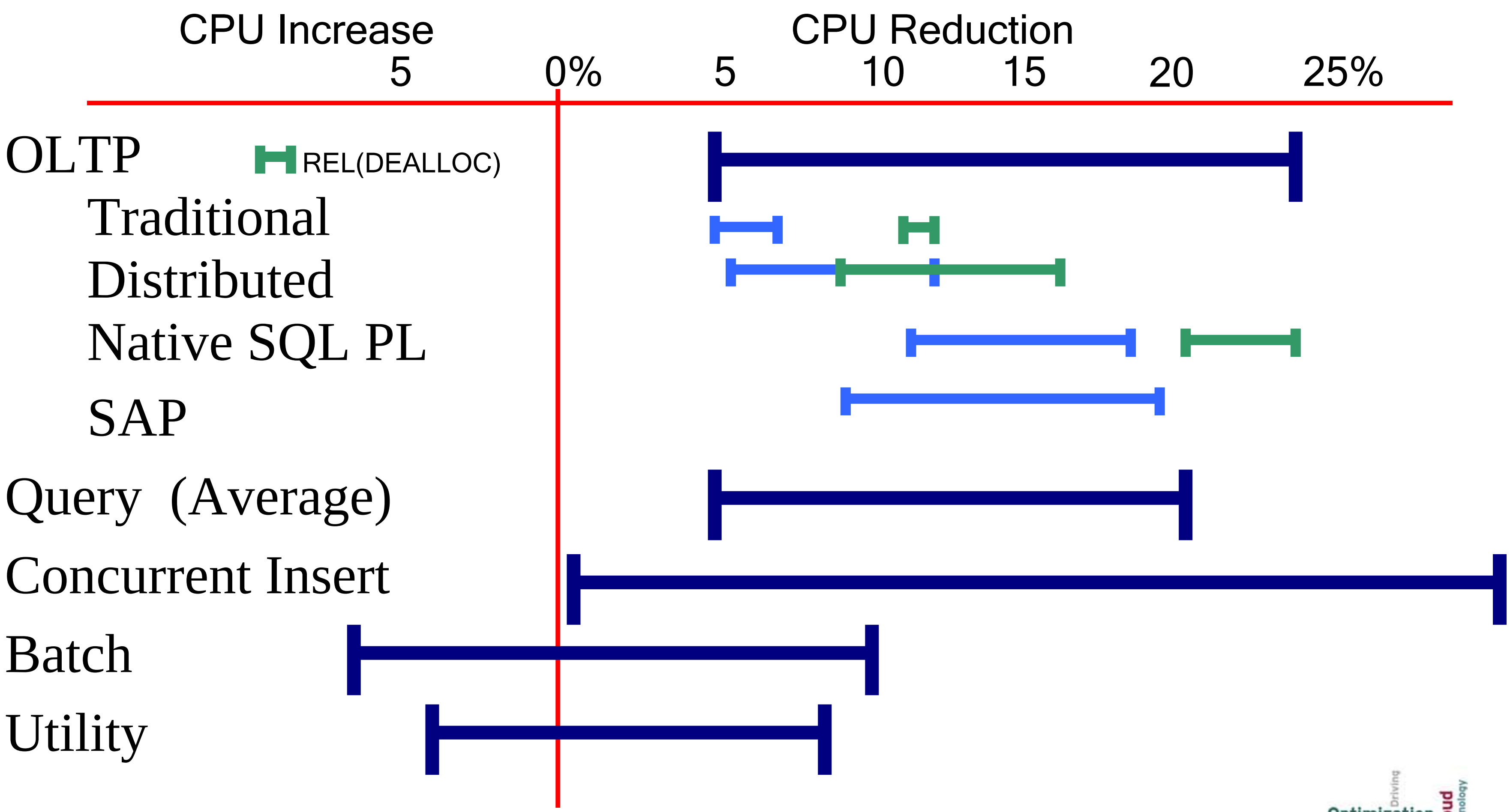
Roger Miller *IBM Silicon Valley Lab*

DB2 10 Performance Objective

Historical goal of <5 % version-to-version performance regression
Goal of 5% -10% performance improvement for DB2 10



DB2 9 to 10 Migration CPU Reduction – IBM Workloads



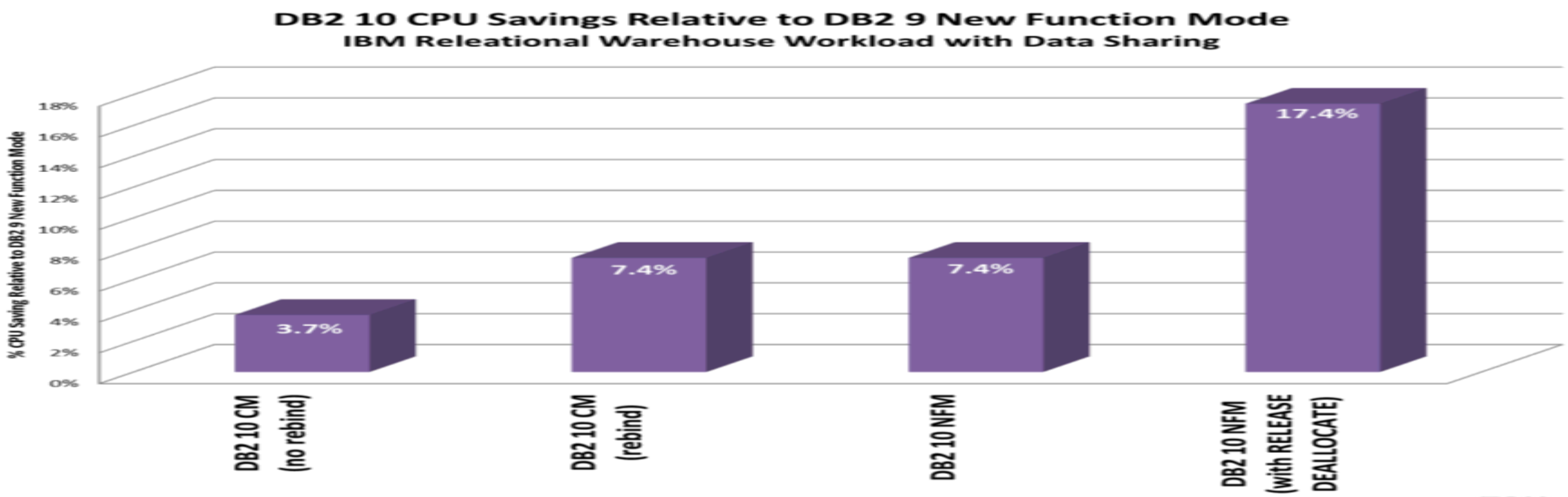
Performance Scalability - DB2 Latches

- Virtual Storage Reduction : 75% to 90%
- Faster process on latch suspend/resume
- Most of known DB2 latches are addressed in DB2 10
  - LC12 : Global Transaction ID serialization
  - LC14 : Buffer Manager serialization
  - LC19 : Log write in both data sharing and non data sharing
  - LC24 : EDM thread storage serialization (Latch 24)
  - LC24 : Buffer Manager serialization (Latch 56)
  - LC25 : EDM hash serialization
  - LC27 : WLM serialization latch for stored proc/UDF
  - LC32 : Storage Manager serialization
- Other contention relief
  - IRLM : IRLM latch contention on hash table
  - CML : z/OS Cross Memory Local suspend lock
  - UTSERIAL : Utility serialization lock for SYSLGRNG (*NFM)
  - Concurrent RE/BIND and most of DDL (*NFM)

Preliminary 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 measured 3.7% CPU reduction from DB2 9
- DB2 10 CM REBIND getting same access path measured 7.4% CPU reduction from DB2 9
- DB2 10 NFM measured same 7.4% CPU reduction from DB2 9
- DB2 10 CM or NFM with RELEASE(DEALLOCATE) measured additional 10% CPU reduction from DB2 10 NFM RELEASE(COMMIT)



Some Beta Customer Performance Feedback

Workload	Results
Distributed Concurrent Insert	50% DB2 elapsed time reduction; 15% chargeable CPU reduction after enabling high perf DBAT
CICS online transactions	Approx. 7% CPU reduction in DB2 10 CM after REBIND, Another 4% reduction with 1MB page usage
CICS online transactions	Approx 5% CPU reduction
Data sharing heavy concurrent insert	38% CPU reduction
Queries	Average CPU reduction 28% from V8 to DB2 10 NFM
Batch	Overall 28% CPU reduction after rebind packages
DDF OLTP	40% CPU reduction for JDBC stored procedures workload, 15% CPU reduction for securities trading

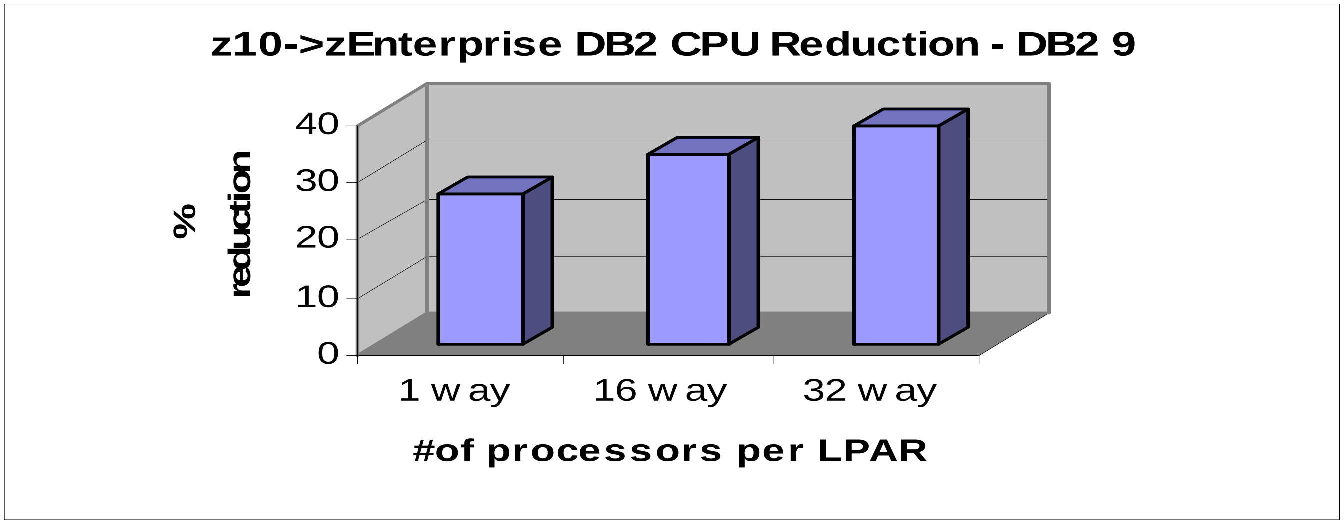
Beta Customer Feedback on Selected New Functions

Workload	Results
Multi row insert (data sharing)	33% CPU reduction from DB2 9, 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

DB2 and zEnterprise 196 performance

Measurement data currently available shows

- DB2 OLTP workloads observing 1.3x to 1.6x DB2 CPU reduction compared to z10 processors
- Higher DB2 CPU reduction can be achieved as number of processors per LPAR increases
- With DB2 10 and zEnterprise, CPU reduction can be up to 1.8x compared to DB2 9 and z10 with many processors per LPAR



System zEnterprise Benefits for DB2

Taking System z synergy to the next level

SHARE
Technology · Connections · Results

- Faster CPUs, more CPUs, more memory → better DB2 performance, scalability
- Compression hardware expected to increase DB2 data compression performance
- Cache optimization, 192M L4 Cache expected to benefit DB2 work
- Hybrid architecture query performance acceleration with IBM Smart Analytics Optimizer
- Excellent synergy with DB2 10 → significant CPU reduction and scalability increase
  - CPU reductions
  - Remove key single system scaling inhibitors: virtual storage, latching, catalog, utilities, …
  - Translation Lookaside Buffer changes expected to improve performance for 1MB page sizes
  - Buffer pool management

Hardware Trends Impacting DB2

- Drive towards multi core, slowing growth in processor frequency
  - Higher N-ways, more parallelism bring potential latching bottlenecks, memory cache thrashing, …
  - S/W techniques for single threaded performance growth
  - Clustered systems for massive scale out and continuous availability
- Specialty engines (price/performance)
- Hybrid systems, accelerators
  - Use cores that are more specialized to their purpose
  - New performance opportunities
  - New programming paradigms (e.g. OpenCL)
- Memory hierarchy design
  - Higher cpu frequencies, n-ways make cache utilization a critical factor
  - Translation Lookaside Buffer design, large System z page sizes
- Solid state disk (and other disk related improvements)
  - Performance, energy consumption, reliability benefits over HDD

DB2 & IBM zIIP Add Value to Database Work

Portions of the following DB2 for z/OS V8, DB2 9 and 10 workloads may benefit from zIIP or zAAP for XML (DB2 9 in blue, DB2 10 in green)*:

1 – **DRDA over TCP/IP connections**

- **DB2 9 for z/OS Remote native SQL procedures**
- **DB2 9 XML parsing** **XML schema validation**
- **Increased portion of DRDA redirected to zIIPs to 60%**
  Improved performance via reduced processor switching

2 - Requests that use parallel queries

- **DB2 9 higher percentage of parallel queries zIIP eligible**
- **DB2 10 more queries eligible, more parallelism**

3 - DB2 Utilities LOAD, REORG & REBUILD functions used to maintain index structures and sort

- **DB2 10 RUNSTATS – options other than column group**

4 - DB2 10 buffer pool prefetch and deferred write

Performance Enhancements Few Changes (CM)

- SQL runtime improved efficiency
- Address space, memory changes to 64 bit, some REBINDs
- Faster single row retrievals via open / fetch / close chaining
- Distributed thread reuse High Performance DBATs
- DB2 9 utility enhancements in CM8
- Parallel index update at insert
- Workfile in-memory enhancements
- Index list prefetch
- Solid State Disk use
- Buffer pool enhancements
  - Utilize 1MB page size on z10
  - "Fully in memory" option (ALTER BUFFERPOOL)

Performance Enhancements need REBIND (CM)

- Most access path enhancements
- Further SQL runtime improvements
- Use of RELEASE(DEALLOCATE)
- SQL paging performance enhancements
  - Single index access for complex OR predicates:
- IN list performance
  - Optimized Stage1 processing (single or multiple IN lists)
  - Matching index scan on multiple IN lists
- Safe query optimization
- Query parallelism improvements
- More stage 2 predicates can be pushed down to stage 1
- More aggressive merge of views and table expressions
  - Avoid materialization of views
- If migrating from V8, get new RUNSTATS before mass rebind

Performance Enhancements requiring NFM

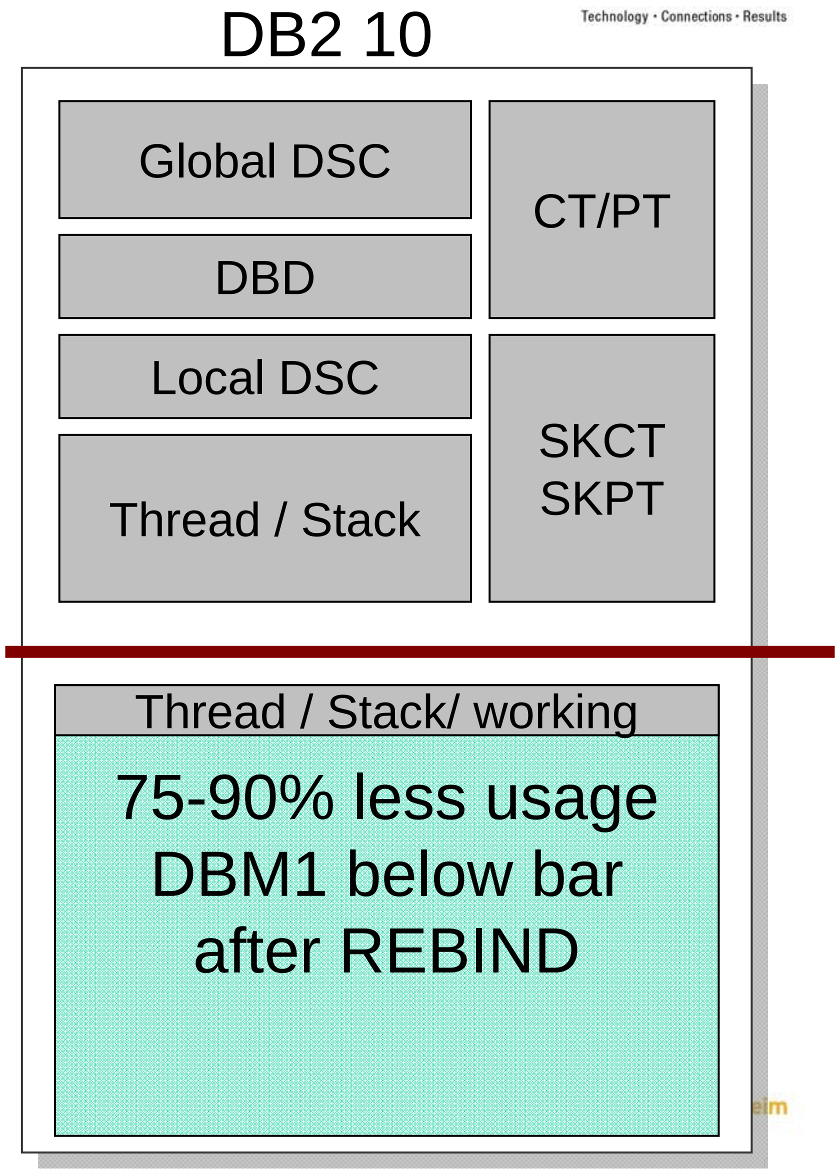
- DB2 catalog concurrency and productivity
- Compress on insert
- Most utility enhancements
- LOB streaming between DDF and rest of DB2
- Faster fetch and insert, lower virtual storage consumption
- SQL Procedure Language performance improvements
- Workfile spanned records, partition by growth
- Access to currently committed data
- Insert improvement for universal table spaces
- Locking improvement for multirow insert
- Efficient caching of dynamic SQL statements with literals

Performance Enhancements need NFM + DBA

- Hash access path Create + Reorg + rebind to activate
- Index include columns Alter + Rebuild + rebind to activate
- Inline LOBs Alter (need universal table space and reordered row format)
- DEFINE NO for LOB and XML columns
- MEMBER CLUSTER for universal table space Alter + Reorg
- Alter to universal table space, page size, data set size, segment size Alter + Reorg
- Online reorg for all catalog and directory table spaces

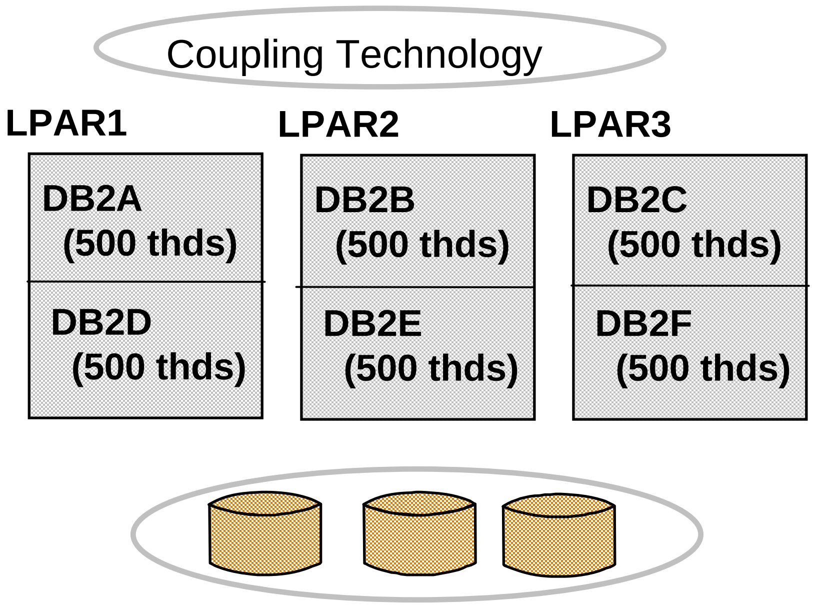
Virtual storage improvements

- DBM1 below 2GB
  - 75-90% less usage in DB2 10 compared to DB2 9
  - Some of working storage (stack, xproc storage) stays below 2GB
- Larger number of threads
  - Possible data sharing member consolidation
- Improve CPU with storage
  - More release deallocate
  - Larger MAXKEEPD values for KEEPDYNAMIC=YES



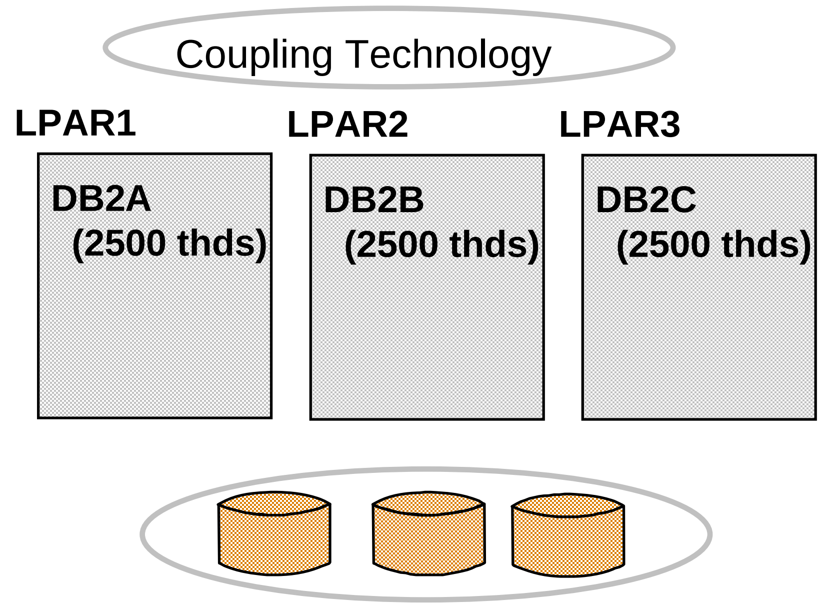
Running Many Active Threads

Today



- Data sharing and sysplex allows for efficient scale-out of DB2 images
- Sometimes multiple DB2s per LPAR

DB2 10



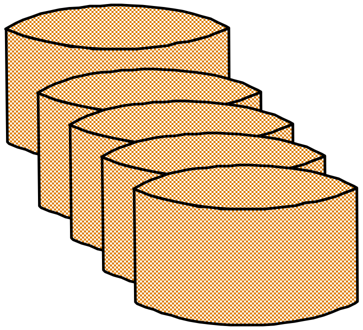
- More threads per DB2 image
- More efficient use of large n-ways
- Easier growth, lower costs, easier management
- Data sharing and Parallel Sysplex still required for very high availability and scale
- Rule of thumb: save ½% CPU for each member reduced, more on memory

Other System Scaling Improvements

- Other bottlenecks can emerge in extremely heavy workloads
  - several improvements reduce latching and other system serialization contention
  - new option to for readers to avoid waiting for inserters
  - eliminate UTSERIAL lock contention for utilities
  - Use 64-bit common storage to avoid ECSA constraints
- Concurrent DDL/BIND/Prepare processes may compete
  - restructure parts of DB2 catalog to avoid the contention
- SPT01 64GB limit can be a constraint, especially if package stability is enabled
  - Allow many more packages by using LOBs
- Improved accounting rollup, compress SMF option

Major changes in DB2 10 catalog & directory

- Improve availability and productivity
- Increase maximum size substantially
- Reduce contention: BIND, DDL, utilities
- Catalog changes: Remove links
  - Many more table spaces, partition by growth
  - Row level locking, reordered row format
  - CLOB and BLOB columns for long strings
    - Inline for performance
  - Online reorganization and check
  - More automatic: DB2-managed SMS-controlled



Insert Performance Improvements

DB2 9

- Large index pages
- Asymmetric index split
- Data sharing Log latch contention and LRSN spin loop reduction
- More index look aside
- Support APPEND option
- RTS LASTUSED support
- Remove log force write at new page (Segmented and UTS) via PK83735

DB2 10 CM

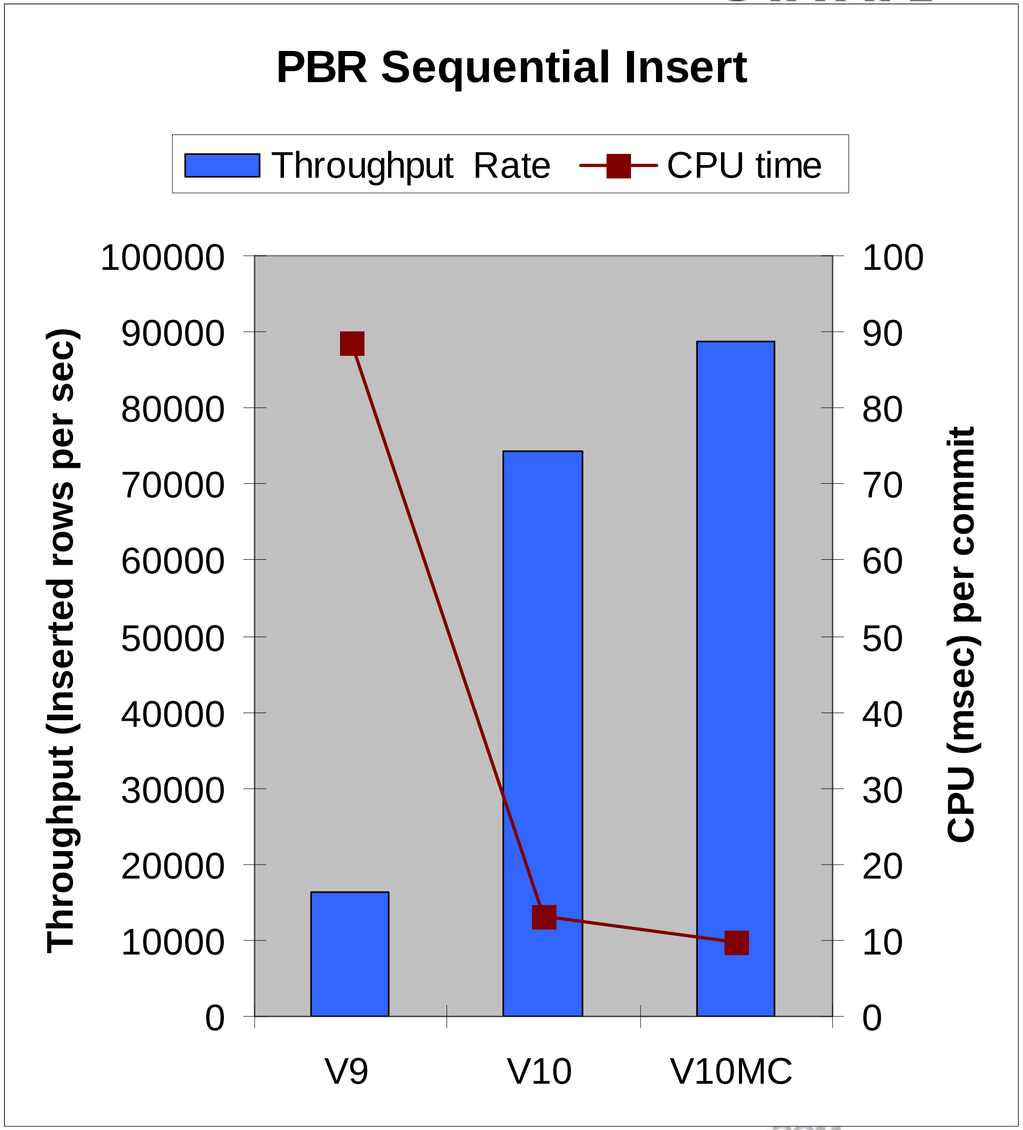
- Space search improvement
- Index I/O parallelism
- Log latch contention reduction and faster commit process
- Additional index look aside

DB2 10 NFM

- INCLUDE index
- Member Cluster in UTS
- More LRSN spin avoidance

Sequential Inserts

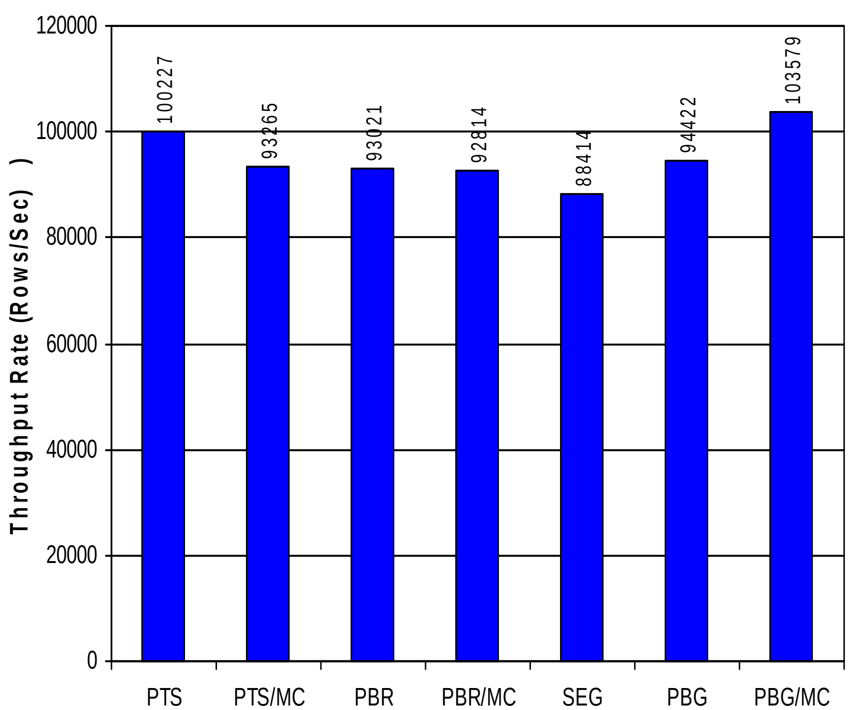
- Optimize when index manager picks the candidate RID during sequential insert
- Member Cluster to distribute space map/data page
- Result: Higher chance to find the space and avoiding a space search
- Less page latch contention with MC
- Test case: Sequential key insert into 3 UTS Partitioned By Range TSs from jdbc 240 clients in 2way data sharing. Multi row insert.



High Insert Rate Comparisons
Sequential / Random Inserts – Page Level Locking

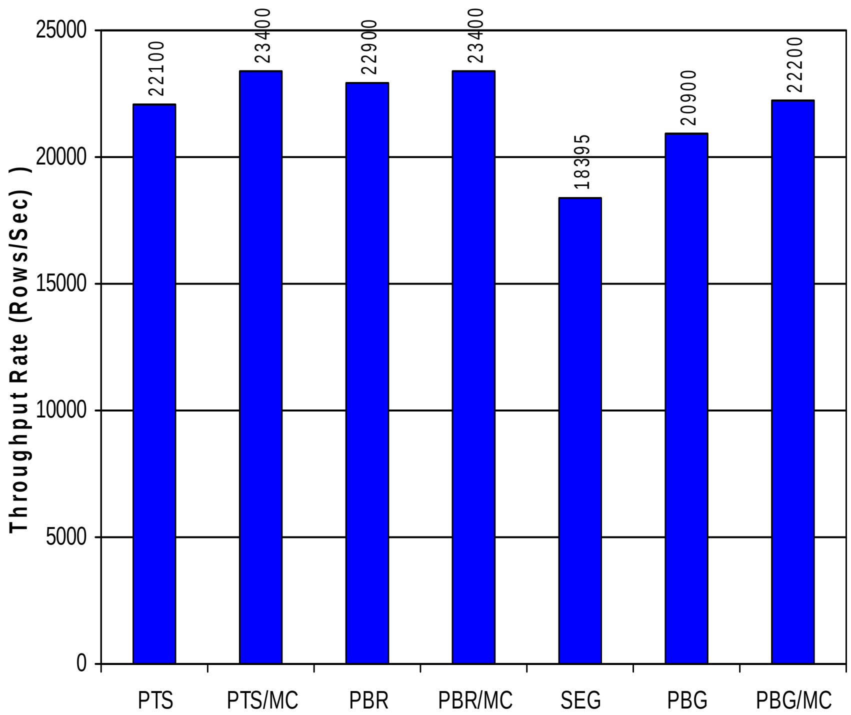
SHARE
Technology · Connections · Results

Sequential Inserts



- Multi-Row Inserts (100)
- Page Level Locking
- 240 concurrent threads
- Commit every 3 inserts with no delay

Random Inserts



- Single-Row Inserts to populated tables
- Page Level Locking
- 200 concurrent threads
- Commit every 3 inserts with 20 msec think time

SHARE in Anaheim 2011

High insert rate summary

- Sequential Insert Performance 10 vs. 9
  - Insert rate improved up to 400%
  - CPU time improved up to 90%
  - UTS with Member Cluster up to 8.8 times insert rate
- Random Insert Performance   10 vs. 9
  - Insert rate improved up to 220%
  - CPU time improved up to 40%
  - UTS with Member Cluster up to 35% insert rate
- Classic vs. UTS Performance
  - o UTS/PBR performance is close to classic PTS
  - o UTS/PBG performance is close to classic SEG
  - o Except for UTS sequential insert with RLL, recommend use of Member Cluster

Select/Fetch Performance Improvement

SHARE
Technology • Connections • Results

9	Sort performance improvement , In memory workfile/Sparse index Index on Expression Many access path related improvements – Plan Stability for static SQL statements – Histogram statistics, cluster ratio, data repeat factor, etc.
10	CPU reduction on index predicate evaluation Better performance using a disorganized index Row Level Sequential Detection Group by using Hash, More in memory workfile usage Sproc improvement by removing column size limitation Dynamic statement cache support for literal constants Many access path related enhancements – Plan stability for both static and dynamic statements – Parallelism improvement – IN list access improvement – Auto stats…and more

CPU reduction in Predicate Evaluation (CM)

- Optimize index predicate evaluation process
  - Applicable in any workload but query with many predicate shows higher improvement

- Performance improvement
  - Preliminary measurements shows average 20% CPU reduction (1% thru 50%) using TPC-H schema and 150 home made queries.

Improvement in using Disorganized Index (CM)

- Index scan using disorganized index causes high sync I/O wait
- Disorganized index detection at execution
- Use List Prefetch on index leaf pages with range scan
  - Reduce Synchronous I/O waits for queries accessing disorganized indexes.
  - Reduce the need of REORG Index
  - Throughput improvement in Reorg, Runstats, Check Index
  - Limited to forward index scan
- Preliminary Performance results
  - Observed 2 to 6 times faster with simple SQL statements with small key size using list prefetch compared to Sync I/Os

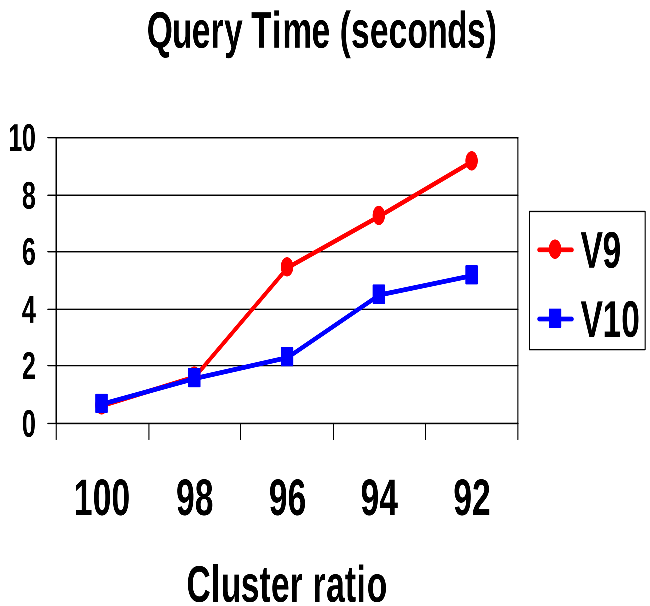
Row Level Sequential Detection (CM)

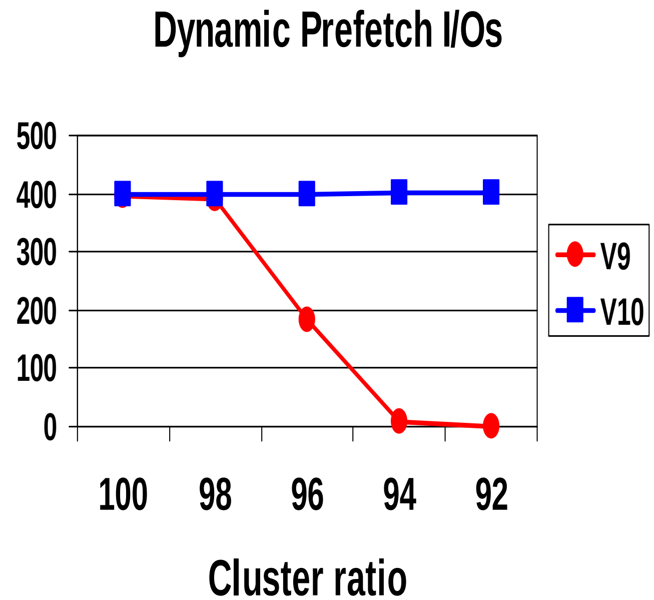
- Problem : Dynamic prefetch sequential detection can work poorly when the number of rows per page is large
- Solution: Row Level Sequential Detection (RLSD)
  - Count rows, not pages to track the sequential detection
- Since DB2 10 will trigger prefetch more quickly, it will use progressive prefetch quantity:
  - For example, with 4K pages the first prefetch I/O reads 8 pages, then 16 pages, then all subsequent I/Os will prefetch 32 pages (as today).
  - Also applies to indexes.

Index—>Data Range Scan

Row size = 49 bytes, page size = 4K (81 rows per page)

Read 10% of the rows in key sequential order





- Row level sequential detection (RLSD) preserves good sequential performance for the clustered pages

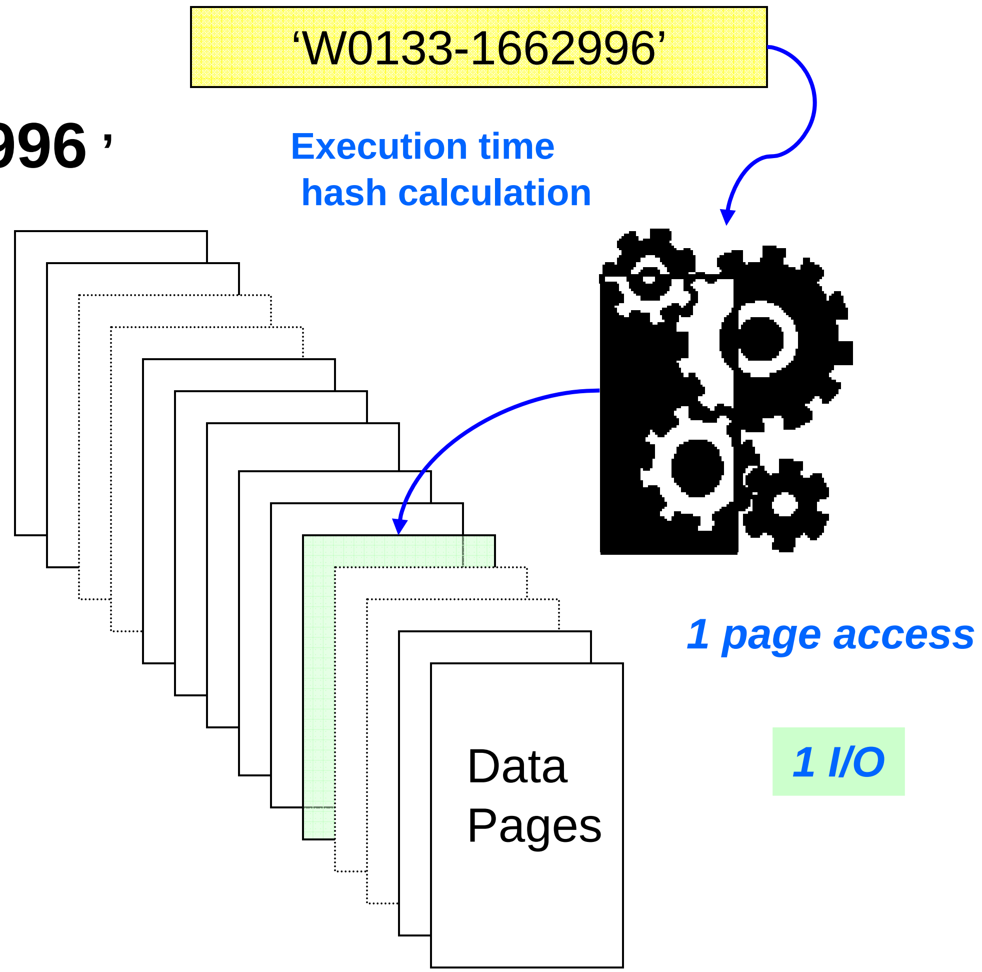
Fastest Access Available In DB2 10: Hash Access

SELECT * …WHERE
ITEMNO = 'W0133-1662996 '

Locate the data row
by hashing the key value.

-Reduced page visits
-Reduced CPU & elapsed time
-Possibly eliminate an index

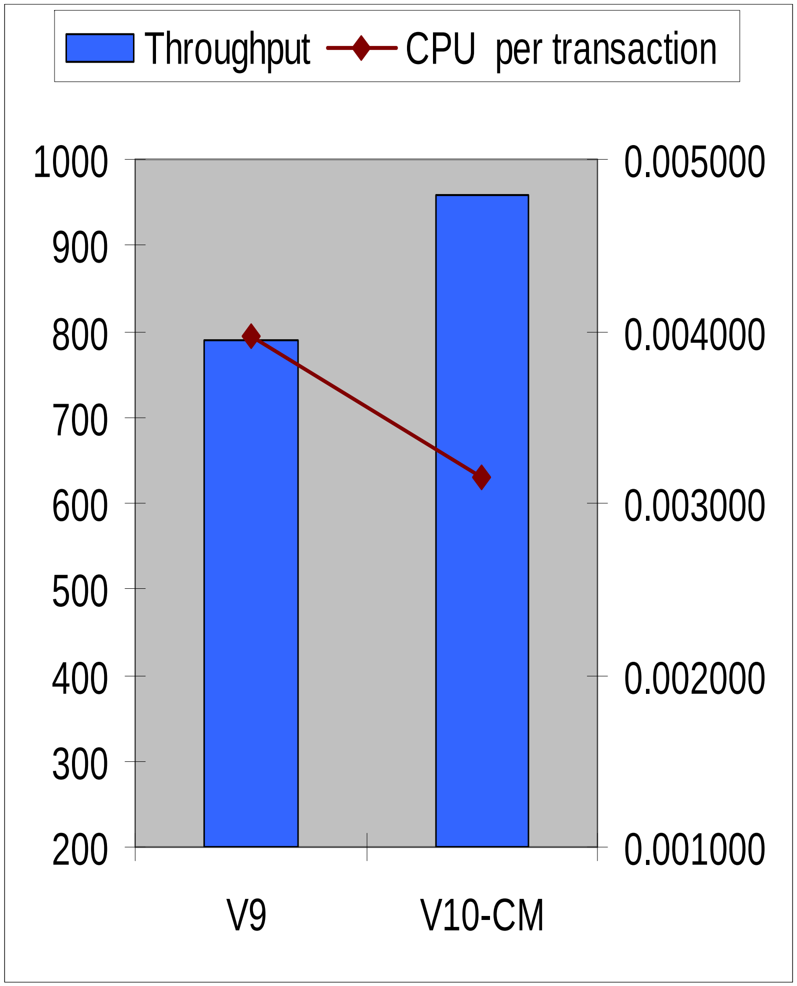
-Tradeoff: extra space used



SQL Procedure Performance (CM)

9	Introduced native SQL Procedure Improvement by executing procedures in DBM1 instead of WLM address space
10	Native SQL Procedures Further performance optimization Specific CPU reduction in commonly used area – Section load avoidance with SET statements with function – Pathlength reduction in IF statement – Optimization in SELECT x from SYSDUMMYx Chained SET statement support (NFM)

Preliminary Measurements – SQL PL (CM)



- OLTP using SQLPL
  - 20% CPU reduction with V10 CM
  - 89% DBM1 Below the Bar usage reduction
  - 5% resp time improvement due to latch contention relief

Local JDBC and ODBC Application Performance

- **Local Java and ODBC applications did not always perform faster compared to the same application called remotely**
  - DDF optimized processing with DBM1 that was not available to local ODBC and JDBC application.
  - zIIP offload significantly reduced chargeable CP consumption
- Open support of DDF optimization in DBM1 to local JCC type 2 and ODBC z/OS driver
  - Limited block fetch
  - LOB progressive streaming
  - Implicit CLOSE
- Expect significant performance improvement for applications with
  - Queries that return more than 1 row
  - Queries that return LOBs

DDF Enhancements

- **Improved performance**
  - Optimizes communication between DDF and DBM1
  - Apply DRDA rule “receiver makes right” to Stored Procedure output parameter
  - No external action required to get the benefit
- **Support of implicit close for cursors declared WITH HOLD and FETCH FIRST FOR 1 ROWS ONLY**
  - Avoids network trip for CLOSE CURSOR if only 1 row is qualifying and cursor is defined WITH HOLD and WITH HOLD is the only reason implicit close could not be used before
- **Optimized Special Register and Inactive thread processing**
  - Improves performance of distributed application that set special register frequently
- Use Unicode for DRDA metadata to avoid EBCDIC->Unicode conversion

High performance distributed

- Re-introducing RELEASE(DEALLOCATE) in distributed packages
  - Could not break in to do DDL, BIND
  - V6 PQ63185 to disable RELEASE(DEALLOCATE) on DRDA DBATs
- High Performance DBATs reduce CPU consumption by
  - RELEASE(DEALLOCATE) to avoid repeated package allocation/deallocation
  - Avoids processing to go inactive and then back to active
  - Bigger CPU reduction for short transactions
- Using High Performance DBATs
  - Stay active if at least one RELEASE(DEALLOCATE) package exists
  - Connections will turn inactive after 200 times
  - Normal idle thread time-out detection will be applied
    - Connection will go inactive and DBAT closed
  - Good match with JCC packages
  - Not for KEEPDYNAMIC YES users

Inline LOBs (NFM)

- CREATE or ATLTER TABLE INLINE LENGTH on UTS
  - INLINE to base table up to 32K bytes
- Completely Inline LOBs
  - Reduce DASD space
    - No more one LOB per page, Compression
  - CPU and I/O saving
    - Avoid LOB aux indexes overhead
    - Small inline LOBs uses 5-10% more than VARCHAR
  - Potential impact on SQLs which does not touch LOBs
- Split LOBs
  - A part of LOB resides in base and other part in LOB TS
  - Incur the cost of both inline and out of line
  - Index on expression can be used for INLINE portion

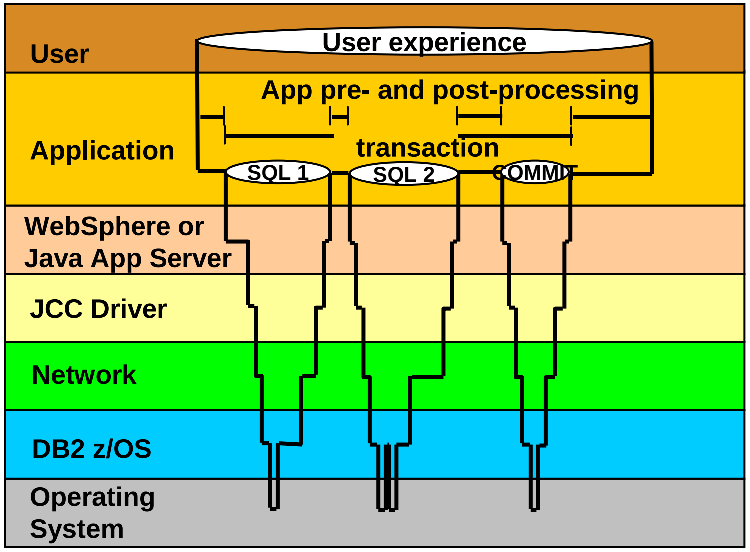
XML performance improvement

- Significant Performance improvement in DB2 9 service stream

- DB2 10 performance improvement
  - Binary XML support
    - Avoid the cost of XML parsing during insert
    - Reduce the XML size
    - Measured 10-30% CPU and elapsed time improvement
  - Schema Validation in engine
    - No more UDF call for validation
    - Utilize XML System Service Parser
      - 100% zIIP / zAAP eligible for validation parser cost
  - XML Update
    - No more full document replace

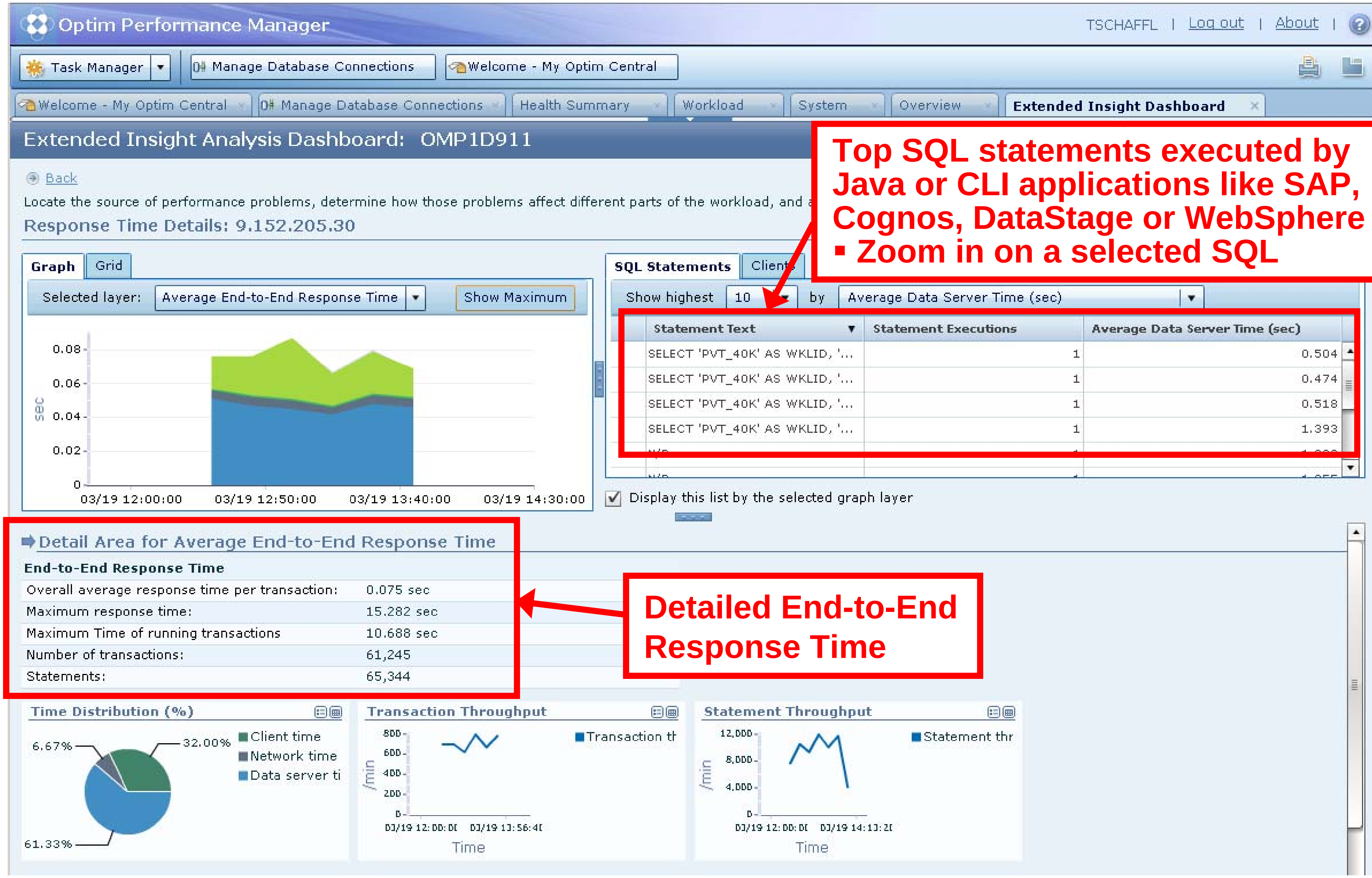
OMEGAMON XE for DB2 Performance Expert 5.1

- **Extended Insight**
  - Surface DB2 for z/OS end-to-end response time metrics
    - Visibility to **all** the components that make up end-user response time
    - Facilitates platform-agnostic identification of response time bottlenecks
    - Enables near-instantaneous response to and prevention of application slowdowns
  - Leverages Tivoli Enterprise Portal GUI
  - Support DB2 9 & 10
- **Summary SQL Reporting**
- **Manage thousands of Threads**
- **Support new DB2 10 Monitoring Data**
- **Lower Monitoring Overhead**



OMEGAMON DB2 PE 5.1 Extended Insight

Zoom into selected workload and see the TOP SQL list



Summary from Beta Evaluations

- Most OLTP shows 5-10% improvement "out of box"
  - A few saw improvement without REBIND
  - Most needed REBIND
  - Some needed protected threads, Rel(DEALLOC), high perf. DBATs
  - Some customers needed 1MB page frames
- Exception of less than 5% CPU savings for very simple SQL
  - Transactional cost does not make SQL optimization visible
- V8 → DB2 10 cumulative performance improvements
  - Utilities Query optimization & stability Insert
- Solid DBM1 virtual storage reduction after REBIND
- Customer requests for estimation
  - No generic way to determine the saving
  - Benchmark key customer workloads

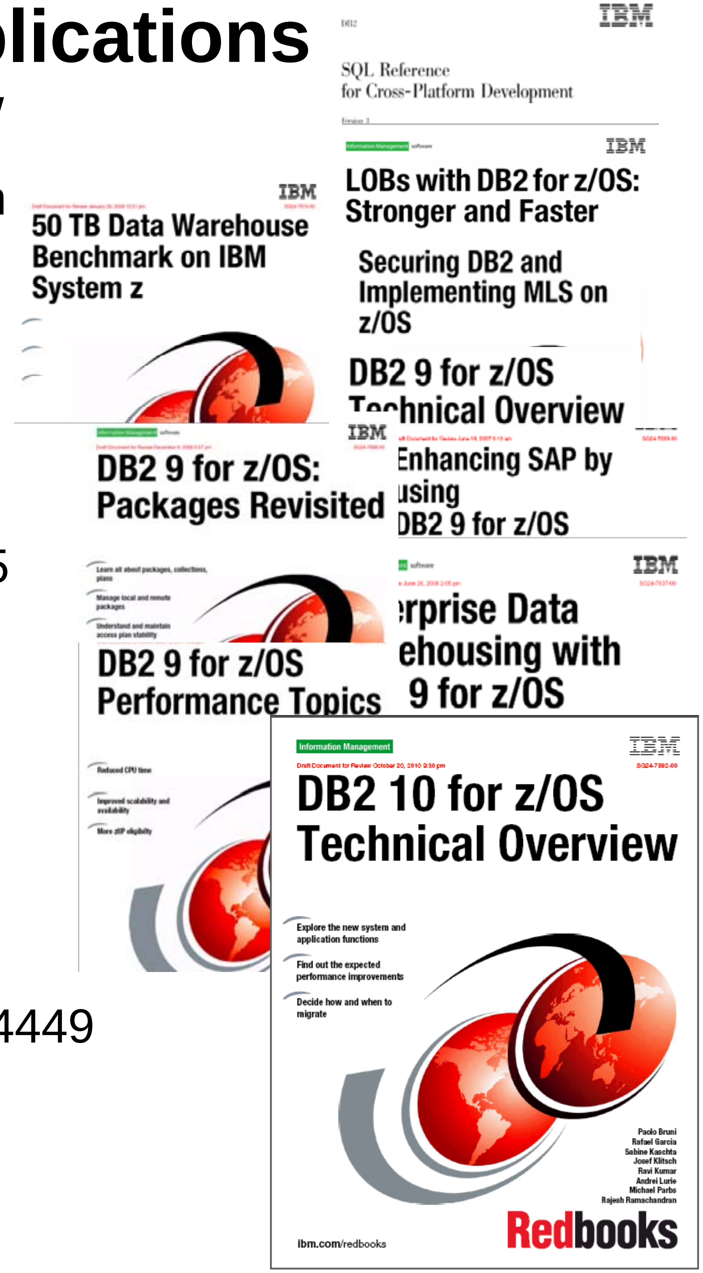
Questions?

Watch for upcoming white papers and conferences

IOD IDUG Share

DB2 9 and 10 IBM Redbooks Publications

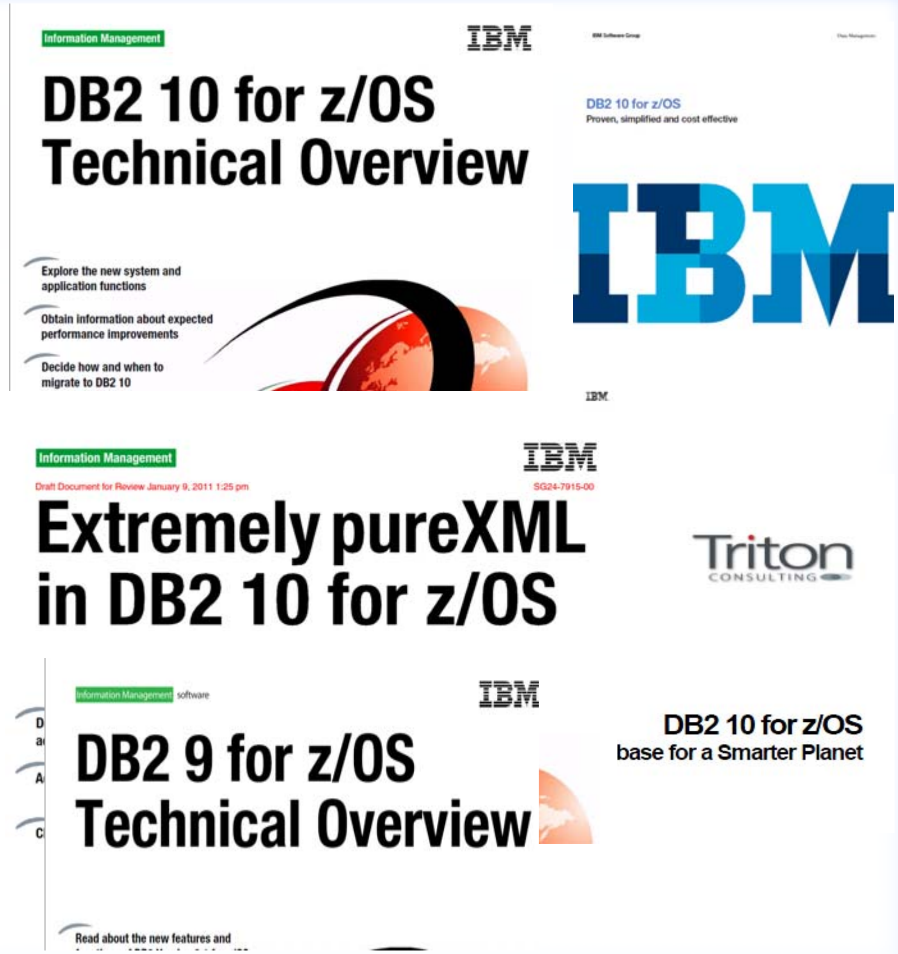
1. DB2 10 Technical Overview SG24-7892 new
2. Extremely pureXML DB2 10 & 9 SG24-7915 new
3. DB2 10 Performance Topics coming soon
4. DB2 9 Technical Overview SG24-7330
5. DB2 9 Performance Topics SG24-7473
6. DB2 9 Stored Procedures SG24-7604
7. Serialization and Concurrency SG24-4725-01
8. Distributed Functions SG24-6952
9. Utilities SG24-6289-01
10. DB2 and Storage Management, SG24-7823
11. Index Compression with DB2 9 for z/OS redp4345
12. SQL Reference for Cross-Platform Development
13. Enterprise Database Warehouse, SG24-7637
14. 50 TB Data Warehouse on System z, SG24-7674
15. LOBs with DB2 for z/OS SG24-7270
16. Deploying SOA Solutions SG24-7663
17. Enhancing SAP - DB2 9 SG24-7239
18. Best practices SAP BI - DB2 9 SG24-6489-01
19. Data Sharing in a Nutshell, SG24-7322
20. Securing DB2 & MLS z/OS SG24-6480-01
21. Data Sharing: Dist Load Balancing & config. redp4449
22. Packages Revisited, SG24-7688
23. Ready to Access Solid-State Drives redp4537
24. Buffer Pool Monitoring & Tuning redp4604
25. Securing & Auditing Data SG24-7720



More information and resources

SHARE
Technology • Connections • Results

- **DB2 main web page**
  http://www.ibm.com/software/data/db2/zos/
- **DB2 10 web page**
  http://www.ibm.com/software/data/db2/zos/db2-10/
- **DB2 books, Information Center**
  - http://www.ibm.com/support/docview.wss?rs=64&uid=swg27011656
  - http://publib.boulder.ibm.com/infocenter/imzic
- **DB2 best practices web page**
  - https://www.ibm.com/developerworks/data/bestpractices/db2zos/
- **DB2 for z/OS IBM Redbooks publications**
  - http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=db2&SearchOrder=4&SearchFuzzy=
- **DB2 presentations**
  ftp://ftp.software.ibm.com/software/data/db2/zos/presentations/



Coming soon to a web screen near you

Information Management

IBM

Draft Document for Review February 25, 2011 11:01 am

SG24-7942-00

DB2 10 for z/OS Performance Topics

Discover the functions that provide reduced CPU time in CM and NFM

Disclaimer/Trademarks

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.