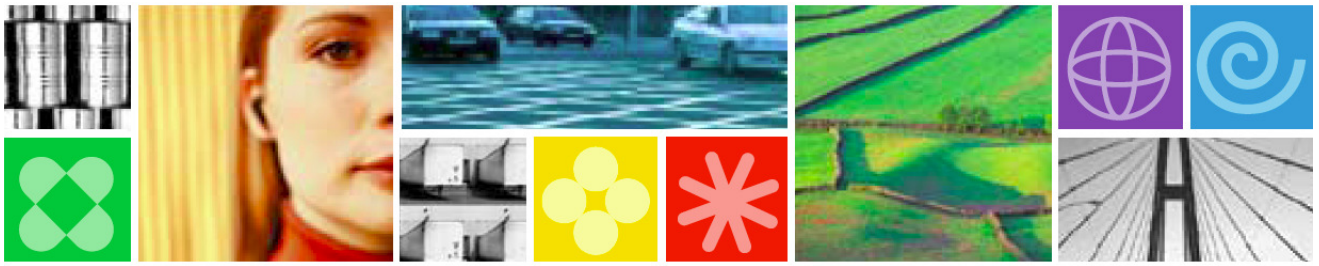
DB2® 10 for z/OS
Technical Overview



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.

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

DB2 10 for z/OS beta is announced, not the product

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.

Trademarks The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the pages of the presentation:

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both: AIX, AS/400, DataJoiner, DataPropagator, DB2, DB2 Connect, DB2 Extenders, DB2 OLAP Server, DB2 Universal Database, Distributed Relational Database Architecture, DRDA, eServer, IBM, IMS, iSeries, MVS, Net.Data, OS/390, OS/400, PowerPC, pSeries, RS/6000, SQL/400, SQL/DS, Tivoli, VisualAge, VM/ESA, VSE/ESA, WebSphere, z/OS, zSeries

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product, or service names may be trademarks or service marks of others.

Performance Disclaimer

This document contains performance information based on measurements done in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the numbers stated here.

DB2 for z/OS Family
Library
Technical resources
Success stories
News
How to buy
Events
Training and certification
Services
Support

Related links
- DB2 for z/OS
- DB2 9 for z/OS
- Data Warehousing and Business Intelligence on System z
- DB2 Tools for z/OS
- DB2 pureXML

Announcing DB2 10 for z/OS Beta

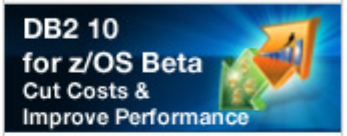
The undisputed leader in total system availability, scalability, security and reliability

In today's business and economic environment, the challenge for IT is clear: improve operational efficiencies, reduce costs, and adapt quickly to support business growth -- all without sacrificing the resiliency required for today's demanding business requirements. DB2 for z/OS is the undisputed leader in total system availability, scalability, security, and reliability at the lowest cost per transaction. DB2 10 builds on the formidable capabilities of DB2 9 for z/OS and continues to set the standard, delivering key innovations and resource savings, including:

Out-of-the-box Savings by improving operational efficiencies

IBM continues to invest in new features to support your efforts to make your business more efficient, and DB2 10 delivers great value in this area. Compared to previous DB2 versions, some customers can achieve a 5% to 10% out-of-the-box CPU savings for traditional workloads and up to 20% out-of-the-box CPU savings for non-traditional workloads. Productivity improvements in DB2 10 for database and system administrators can drive additional operational efficiencies and cost savings. Synergy with other IBM System z platform components reduces CPU use by leveraging the latest processor improvements, larger amounts of memory, solid-state disk and z/OS enhancements.

DB2 10 for z/OS Highlights



→ Announcement Letter

→ DB2 for z/OS and Data Warehousing Nordic Seminars

→ Hear what people are saying about DB2

DB2 for z/OS Roadshow - including DB2 10 for z/OS Sneak Preview

Coming to a city near you!

Feb 9 St. Louis, Feb 11 NYC, Feb 18 Toronto, Mar 4 Chicago and many more

→ Register Now - Track 2

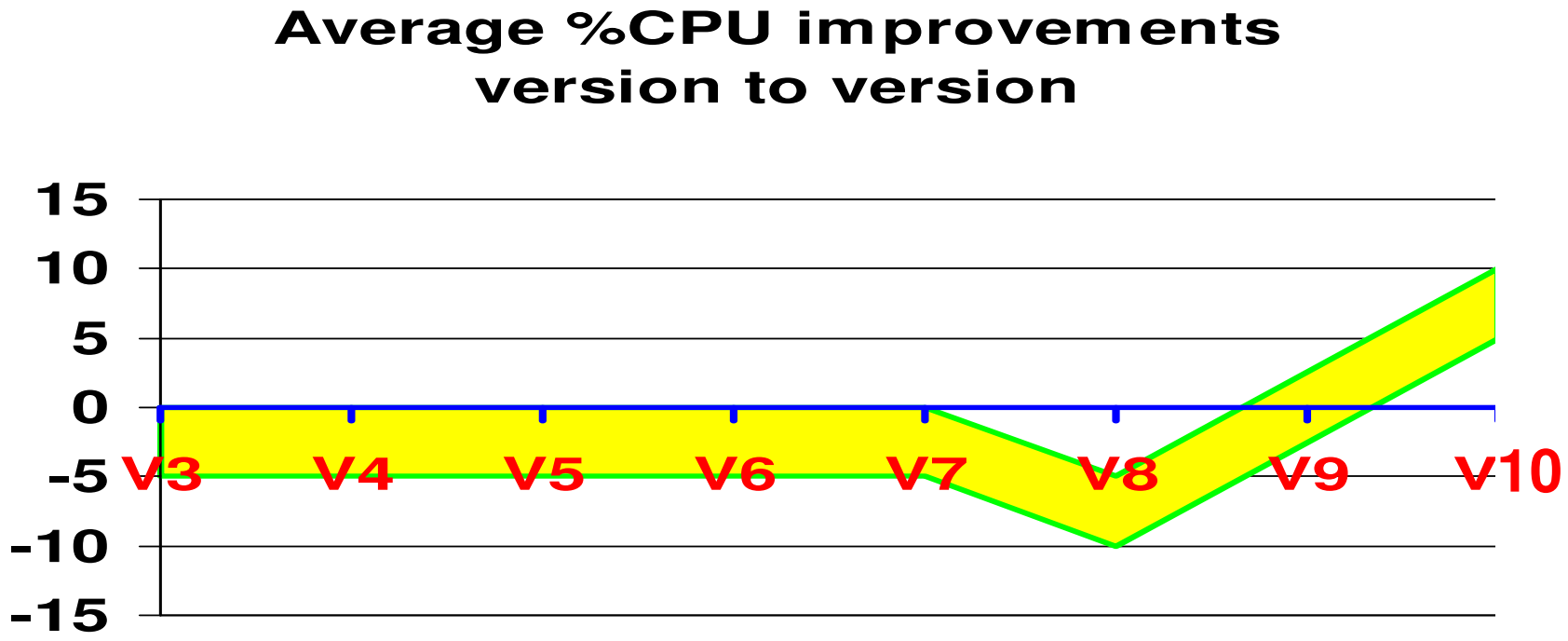
DB2 10 for z/OS TECHNICAL PREVIEW

Agenda

- CPU performance, scalability, and buffer management
- Insert performance
- Query performance and query management
- LOB performance
- Availability – online schema
- Utility enhancements
- Security and audibility enhancements
- Skip-level migration

DB2 10 Performance Objective

- Historical goal under 5% performance regression
- Goal 5% -10% initial performance improvement
- Many customers reduce CPU time 10% - 20%



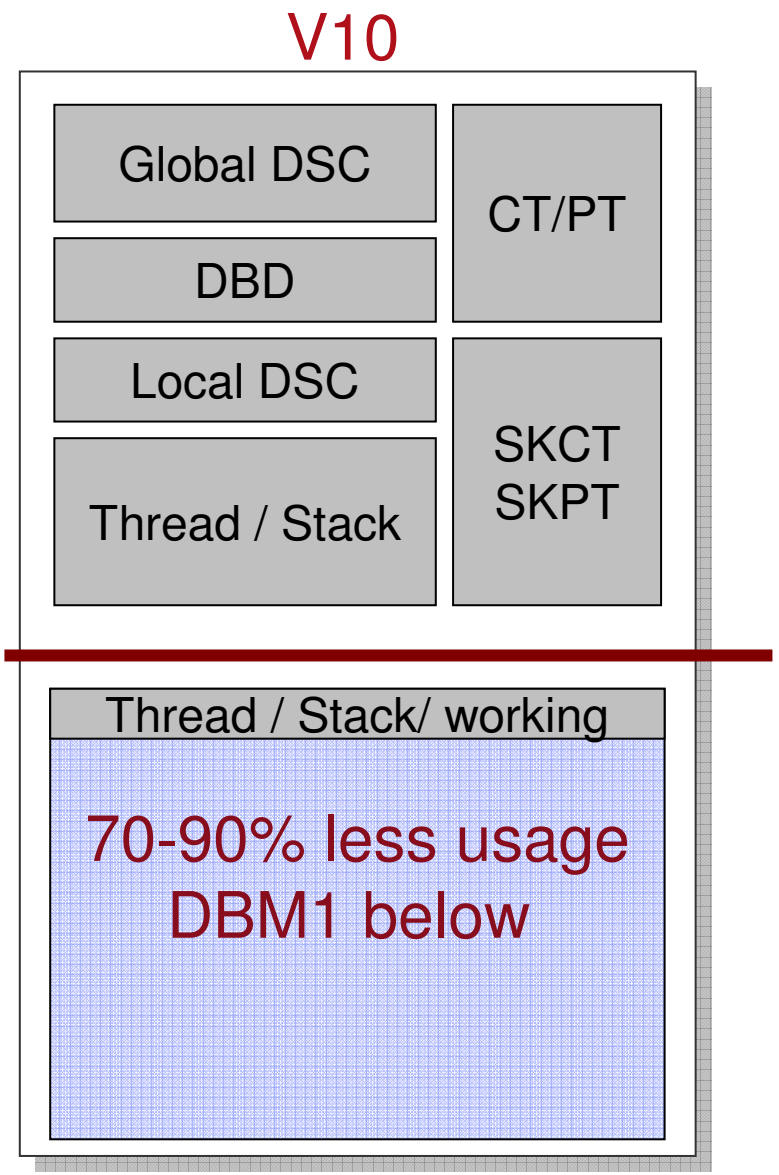
DB2 10 Performance Expectation

- Up to 10% CPU reduction after REBIND packages
  - Higher improvement with workload with scalability issues in V8/V9 or accessed thru DRDA

- Workload using native SQL procedures : up to 20% CPU reduction after DROP/CREATE or REGENERATE the procedures
- Concurrent sequential insert: 0-40% CPU reduction depends on table space type
- Query : 0-20% CPU reduction without access path change
  - Higher Improvement with positive access path change

DBM1 VSCR

- DBM1 below 2GB
  - 70-90% less usage in V10 compared to V9
  - 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



Performance Scalability - DB2 Latches (CM)

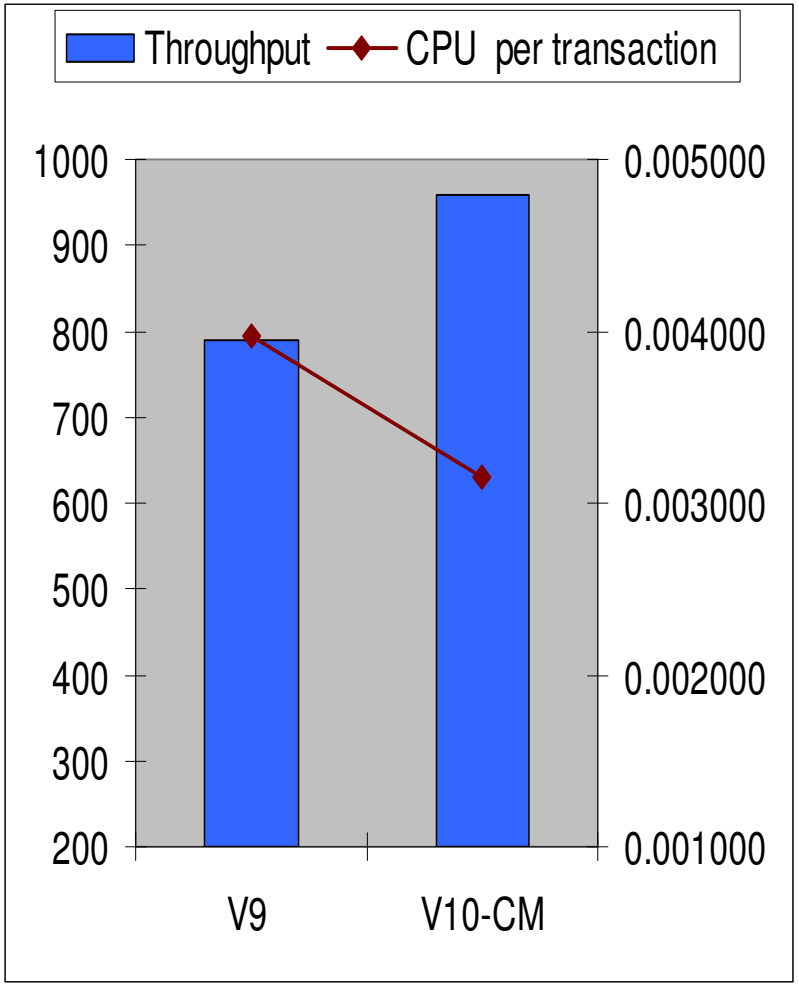
Most of DB2 latches from 64 cp scalability evaluation will have a relief

- 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
- IRLM : IRLM hash contention
- CML : z/OS Cross Memory Local suspend lock
- UTSERIAL : Utility serialization lock for SYSLGRNX *need to be in NFM

Preliminary Measurements of IBM Relational Warehouse Workload (IRWW) with data sharing

- V9 NFM REBIND with PLANMGMT(EXTENDED)
- V9 NFM -> V10 CM without REBIND
  - Measured 3.7% CPU reduction from V9
- V10 CM REBIND with APREUSE(YES) APCOMPARE(ERR)
  - Measured 7.4% CPU reduction from V9
- V10 NFM
  - Measured same 7.4% CPU reduction from V9
- V10 NFM with RELEASE (DEALLOCATE)
  - Measured additional 10% CPU reduction from V10 NFM RELEASE(COMMIT)

Preliminary Measurements – SQLPL (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

Catalog and directory restructure

- Uses row level locking, which avoids serialization of DDL and Bind/Prepare processing
- Fewer timeouts/deadlocks
- DSN1CHKR no longer needed
- Allow online Reorg for all catalog and directory table spaces
- Partition By Growth, for more efficient space management of the catalog
  - Requires system managed storage, DB2 managed objects
- Allow more packages by using LOBs for SPT01
- Individual Bind/Prepares take longer and use more CPU time

Performance Scalability - H/W synergy

- Exploitation of z10 features
  - CPU improvement using z10 specific instructions
  - Large fixed page frames for buffer pool
    - Buffer pools with PGFIX=YES
    - Define LFARE (size of 1MB page area) in IEASYSxx
    - CPU reduction by reducing miss in TLB (translation lookaside buffer)
    - Observed 1-4% CPU reduction
- Avoidance of sequential BP scan during page set p-lock negotiation
  - Performance impact with large buffer pools in data sharing
- In memory buffer pool with large real
  - DB2 managed in memory buffer pool
    - PGSTEAL = NONE
    - Pre-load the data at the first open or at ALTER BPOOL
    - Avoid unnecessary prefetch request (similar to VPSEQT=0)
    - Avoid LRU maintenance -> no LRU latch (LC14)

Insert Performance Improvement

V9

- 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

V10 CM

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

V10 NFM

- INCLUDE index
- Support Member Cluster in UTS
- Multirow Insert LRSN spin avoidance

General Insert Enhancements

- Log latch reduction in both data sharing and non data sharing
  - Multirow Insert LRSN spin avoidance (NFM)
- Eliminate Mass Delete Locks from UTS
- Referential integrity check performance
  - Sequential detection and index look aside for RI
  - Avoid RI check for each insert of a child under the same parent
- Performance improvement for sequential inserts into the middle of a cluster index
  - Significant space search improvement in sequential insert
- Member Cluster option now available with UTS (PBG/PBR)

Universal Table Space (UTS) – Member Cluster (NFM)

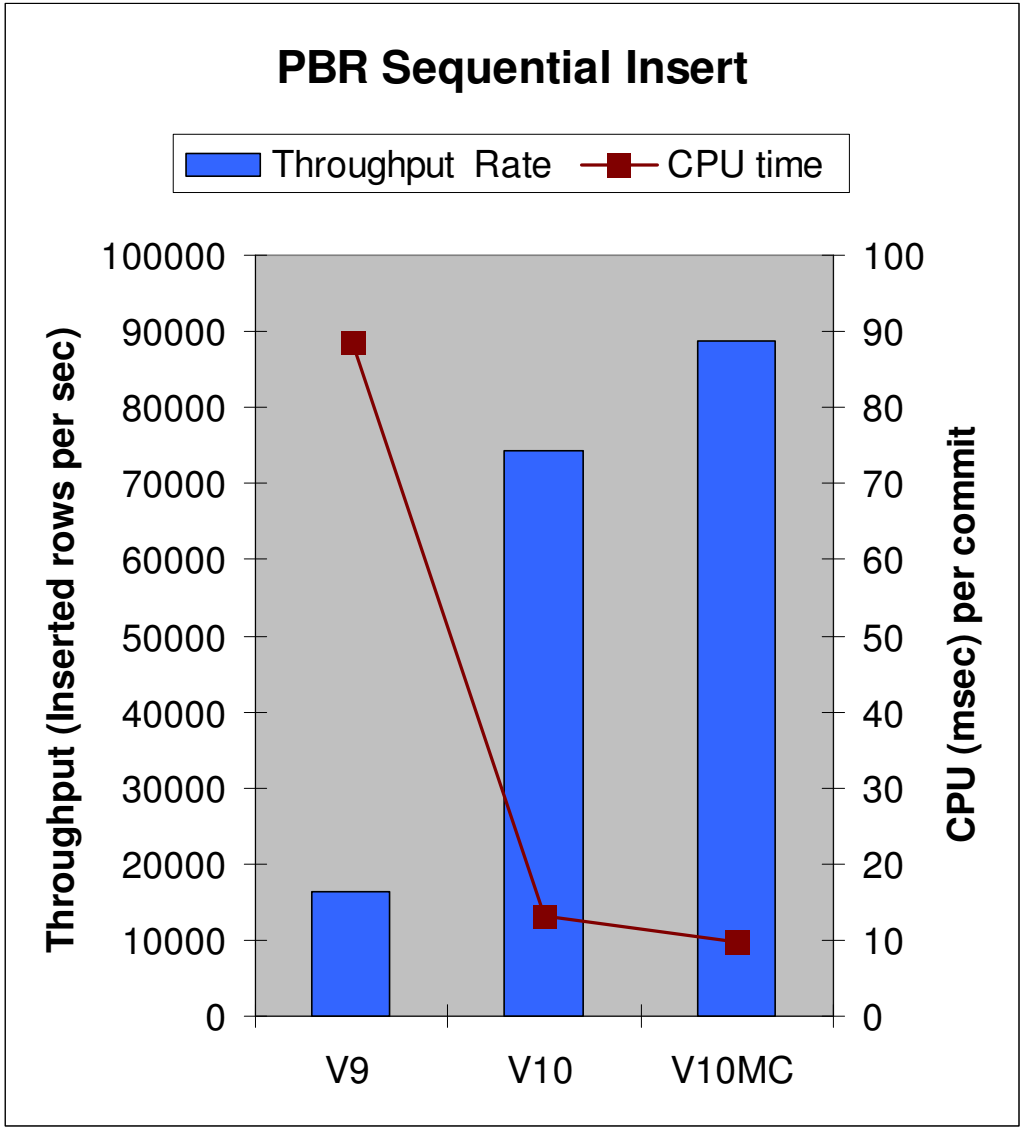
- Member Cluster option in create table space
  - Assigns a set of pages and associated space map page to each member
  - Remove the “hot spots” in concurrent sequential insert in data sharing
  - It does not maintain data cluster during the INSERT
  - Data cluster needs to be restored via REORG
  - Each space map contains 10 segments
- Altering to MEMBER CLUSTER

  ALTER TABLESPACE MyTableSp

  MEMBER CLUSTER YES/NO;
  - REORG/LOAD on the table space level to materialize the pending alter
  - RECOVER needs image copy taken from REORG that materialized the pending ALTER

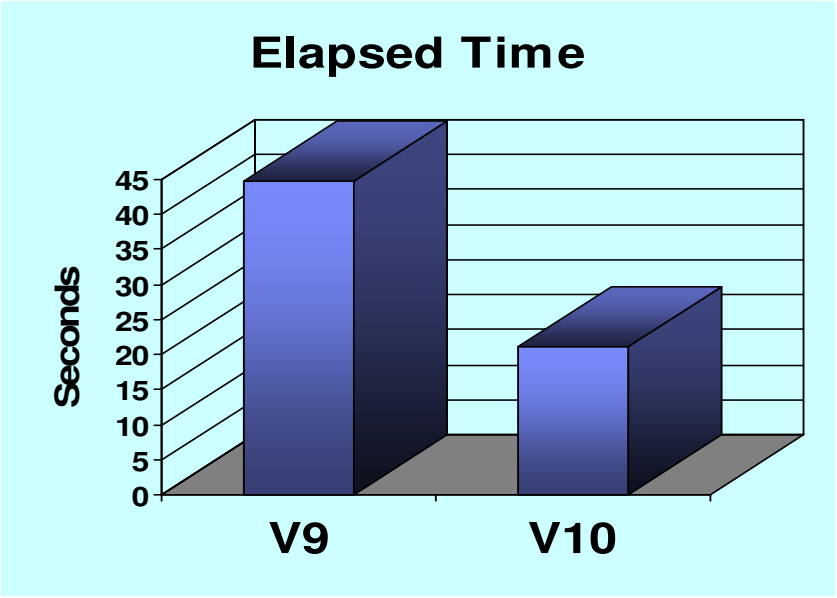
Sequential Inserts

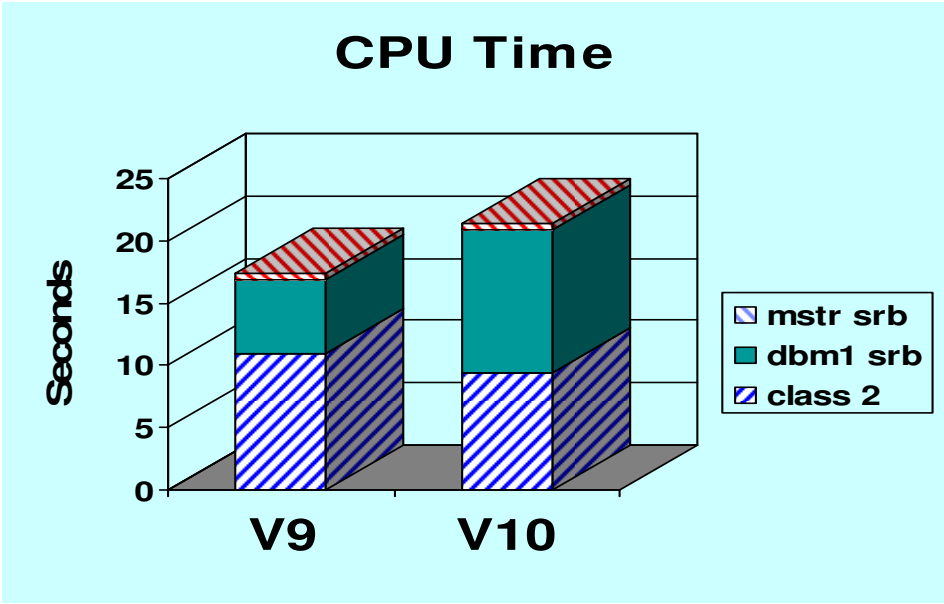
- Optimize when index manager picks the candidate RID during sequential insert
- Member Cluster to distribute space map/data page
- Result: Higher chance to find the space and avoiding a space search
- Less page latch contention with MC
- Test case: Sequential key insert into 3 UTS Partitioned By Range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



Insert I/O Parallelism for index reads

Example: 2000 random inserts, 6 indexes





Additional Non-key Columns in a unique index (NFM)

V9 — Multiple indexes per table

An index is used to enforce uniqueness constraint. Additional indexes are necessary to achieve index only access on columns not part of the unique constraint during queries.

Higher Insert / Delete CPU time, Increased storage requirements

V10 — Additional Non-key Columns in an unique indexes

Reduce index maintenance cost during insert, DASD space saving

Preliminary measurement

2 Index vs 1 index with INCLUDE columns shows 30% cpu reduction in insert with same query performance using the indexes.

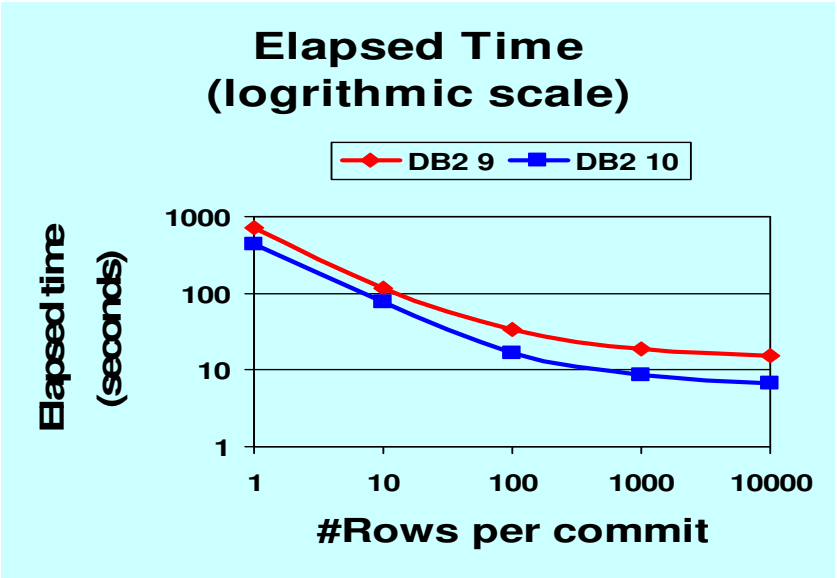
Auto compress

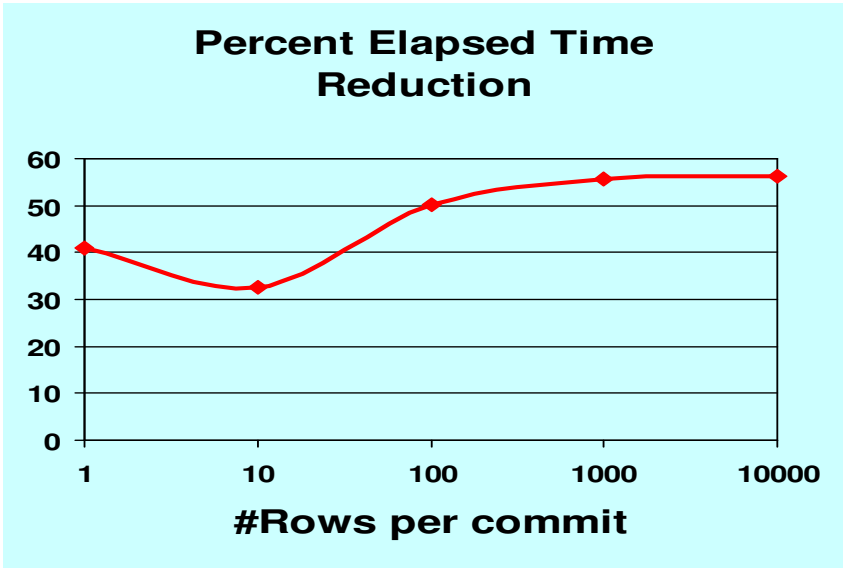
- DB2 9
  - A table space or partition created with COMPRESS YES will not compress data until REORG or LOAD utility builds a compression dictionary
  - For some customers, REORG or LOAD are not executed for a long period of time after data is initially inserted thereby not allowing for compression.
- DB2 10
  - INSERT, MERGE, or a LOAD SHRLEVEL CHANGE can trigger the asynchronous building of a compression dictionary when hitting a threshold

Improved log performance

- Reduced log latch time
- Log I/O reduction and improved overlap of primary/secondary log I/Os

Insert performance when CPU and log latch are unconstrained





- At 10,000 rows/commit, DB2 10 is becoming CPU constrained
- Expect more value when using PPRC or equivalent function

Logging Enhancements…..

- Dynamic add of active logs
  - New –SET LOG NEWLOG option
  - New active log must be IDCAMS defined and preformatted by DSNJLOGF
  - Only a single log data set at a time
    - Issue command twice for dual logging
  - Limit is still 93 active log pairs
  - No dynamic delete of active logs
- Preemptable backout
  - Pre-DB2 10, abort/backout schedules non-preemptable SRB
    - On single CPU system may give impression of DB2 hang
  - DB2 10: Create enclave at restart for preemptable SRB backout processing

…..Logging Enhancements

- Checkpoint intervals based on both time and log records
  - Meaning of CHKFREQ is unchanged
    - Minimum # of log records raised from 200 to 1000
  - New ZPARMS to control behavior
    - CHKLOGR – number of log records between checkpoints
    - CHKMINS – number of minutes between checkpoints
    - CHKTYPE SINGLE|BOTH – govern old/new
  - Set by dynamic ZPARM or –SET LOG command
    - -SET LOG change does not persist across restart
  - -DIS LOG command command indicates settings and if mode is SINGLE or BOTH

Hash Access

- Applicable when a unique key is used to retrieve a single row
- Performance improved by avoiding index
  - Save the CPU cost for each Getpage in a multi-level index
  - Save the CPU time for the index I/Os
  - Save the elapsed time for the index I/Os
- Hash organization is not suitable for high-insert tables
- Hash organization will hurt the performance of range scans that were otherwise based on a clustering key
- The size of the hash table must be carefully chosen to balance performance with efficient space usage

Query plan management

- Dynamic SQL with literals can now be re-used in the cache
- Plan stability extensions
  - Capture access paths for static and dynamic SQL in a repository
  - Save multiple copies of access paths
  - Switch back to prior access paths
  - Control when captured dynamic SQL queries would be re-optimized
  - Regenerate runtime structures without changing access paths
    - Important for mass REBIND at DB2 migration
  - Perform before/after access path comparisons

Query performance enhancements

- Safe query optimization
- View/Table expression merge
- IN List Processing
- SQL Pagination
- Parallel Enhancements

Safe query optimization

- Default RID pool size increased from 10MB to 400MB

V9	Optimizer avoids choosing list prefetch if it estimates the RID list will be too big If runtime runs out of RID pool, the query falls back to table scan
V10	Optimizer may liberalize its use of list prefetch If runtime runs out of RID pool, overflow to a work file If a work file overflows its buffer pool, it will spill to DASD

- Work file usage may increase. Mitigate by allocating more work file data sets.

Autonomic Statistics

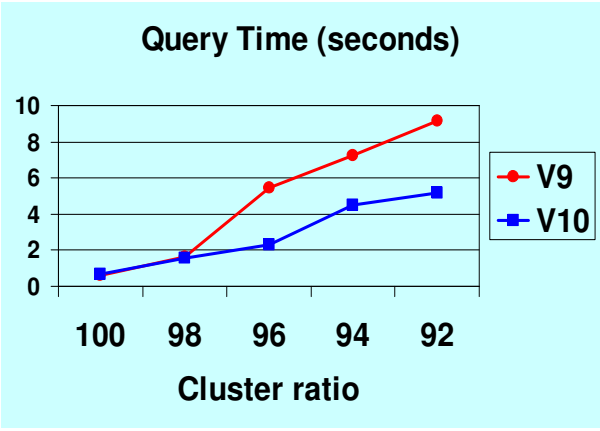
- Automate process
- Autonomic Statistics is implemented through a set of Stored Procedures
  - ADMIN_UTL_MODIFY
  - ADMIN_UTL_EXECUTE
  - ADMIN_UTL_MONITOR
    - Identifies out-of-date/missing/inconsistent stats, updates profile
- SP's run automatically according to a predetermined schedule
- Working together, these SP's
  - Determine what stats to collect
  - Determine when stats need to be collected
  - Schedules and performs the stats collection
  - Records activity for later review

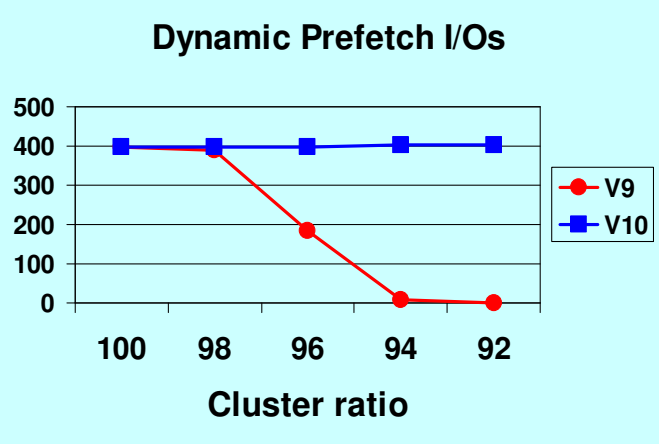
Index—>Data Range Scan

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

Test case	Cluster ratio	Cardinality	NPAGES
1	100%	20,000,000	253167
2	98%	20,200,000	256024
3	96%	20,400,000	258882
4	94%	20,600,000	261740
5	92%	20,800,000	264598

Read 10% of the rows in key sequential order



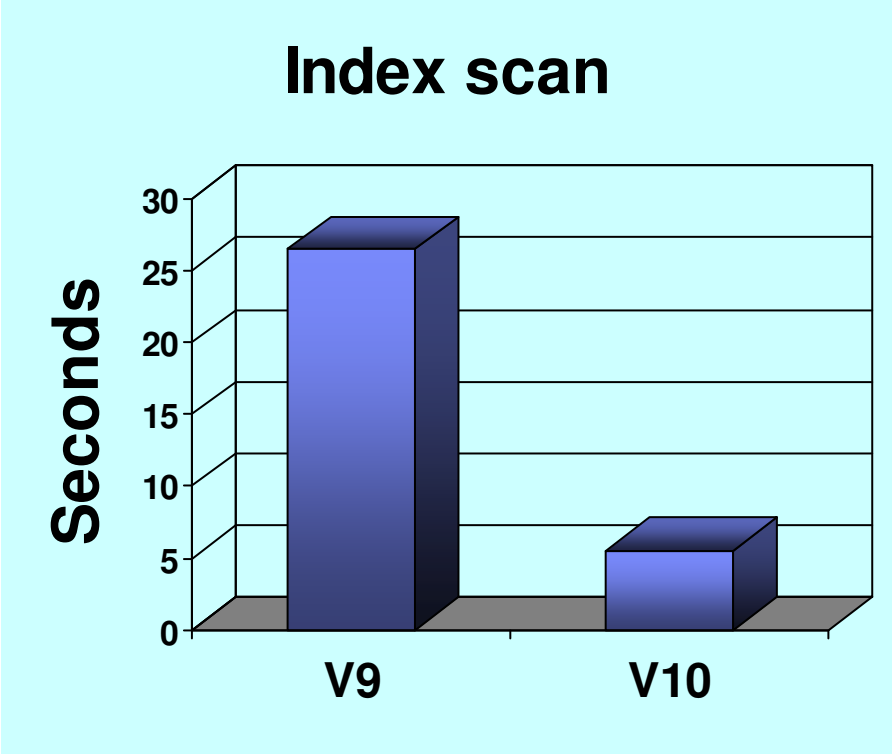


- Row level sequential detection (RLSD) preserves good sequential performance for the clustered pages

Progressive dynamic prefetch quantity

- RLSD may trigger prefetch more quickly than dynamic prefetch in DB2 9.
- The first prefetch I/O will use a “quarter” quantity and the second will use “half”. Therefore, the full prefetch quantity will be used.
  - With 4K page size, the full prefetch quantity is still 32 pages
- Progressive prefetch applies to indexes too.

Disorganized indexes use list prefetch

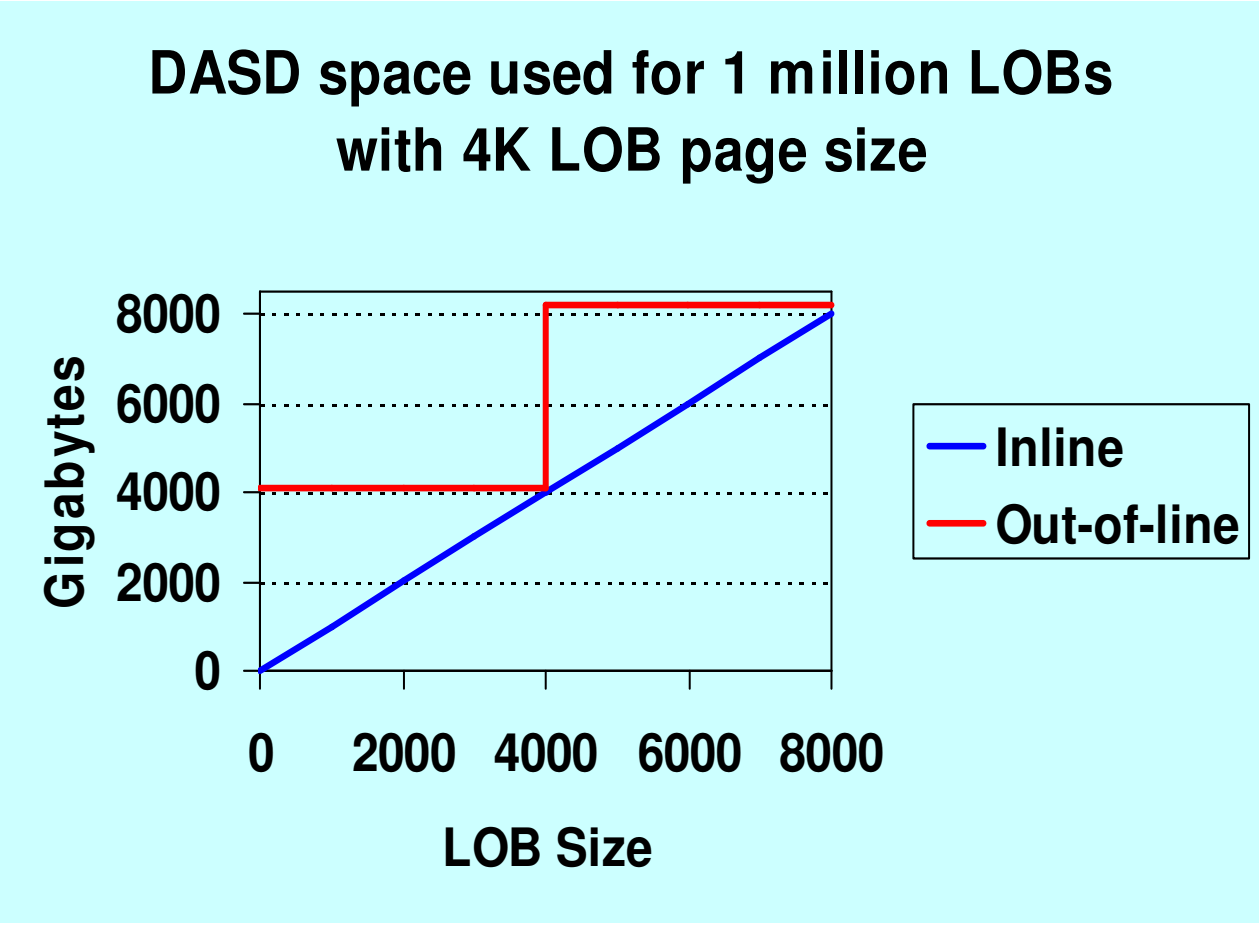


80% reduction in elapsed time

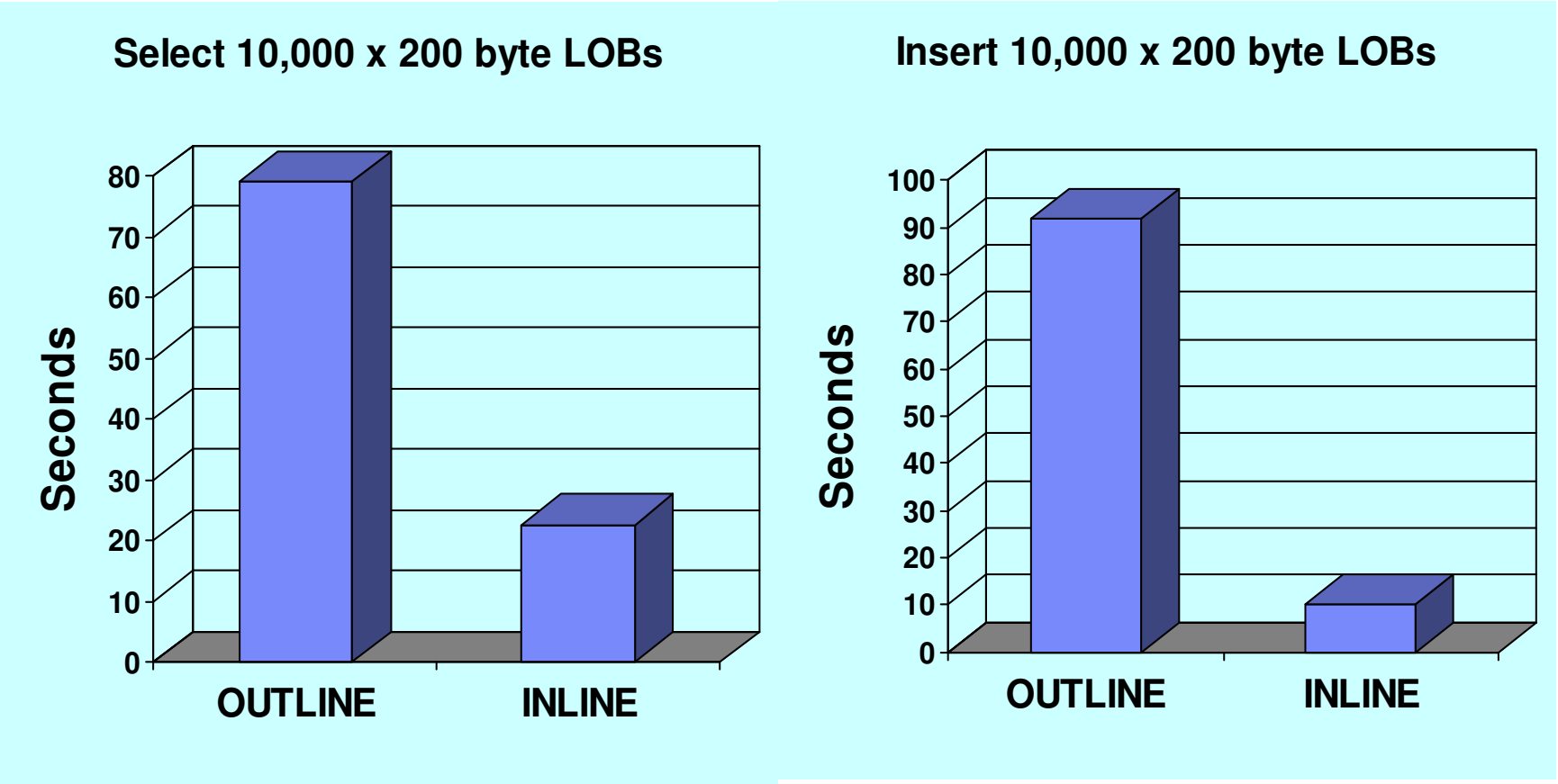
Inline LOBs

- Treat small LOBs like VARCHAR, with VARCHAR-like performance
  - Avoid all of the Getpages and I/Os associated with an auxiliary index and LOB table space
  - Enable Dynamic Prefetch for small LOBs
- Save DASD space for small LOBs
- Support index on expression for inline portion of the LOB
  - Exploited by Spatial
- The inline portion of a LOB may be compressed
- You may want to increase the page size to accommodate larger rows. Also rebuild the dictionary.

Using Inline LOBs to save DASD space

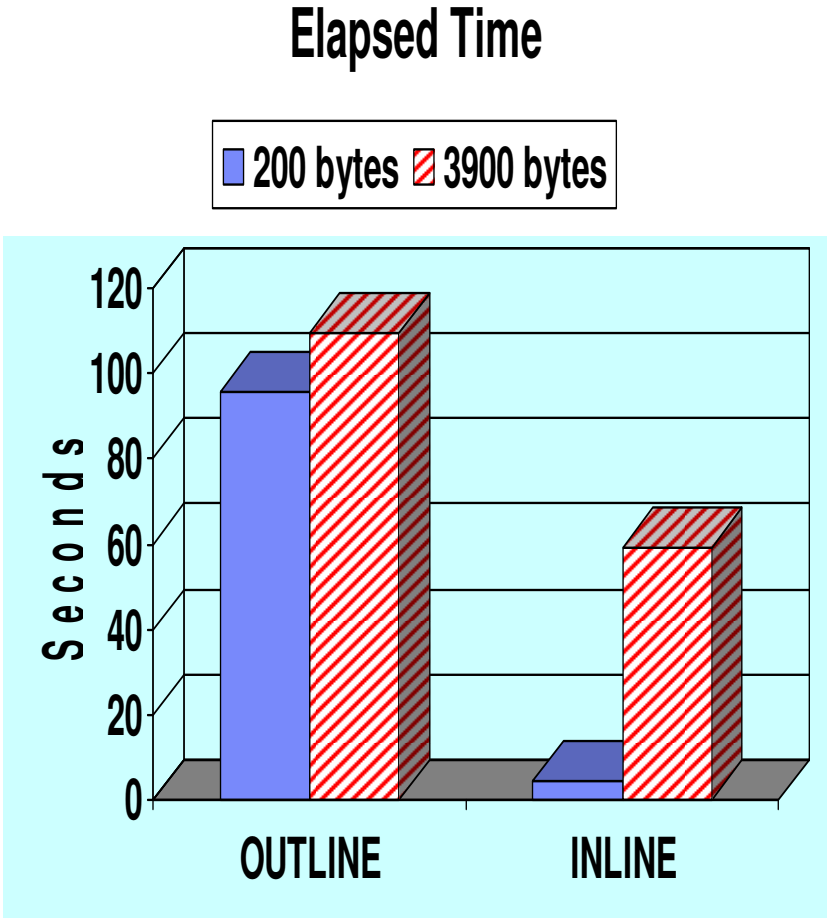


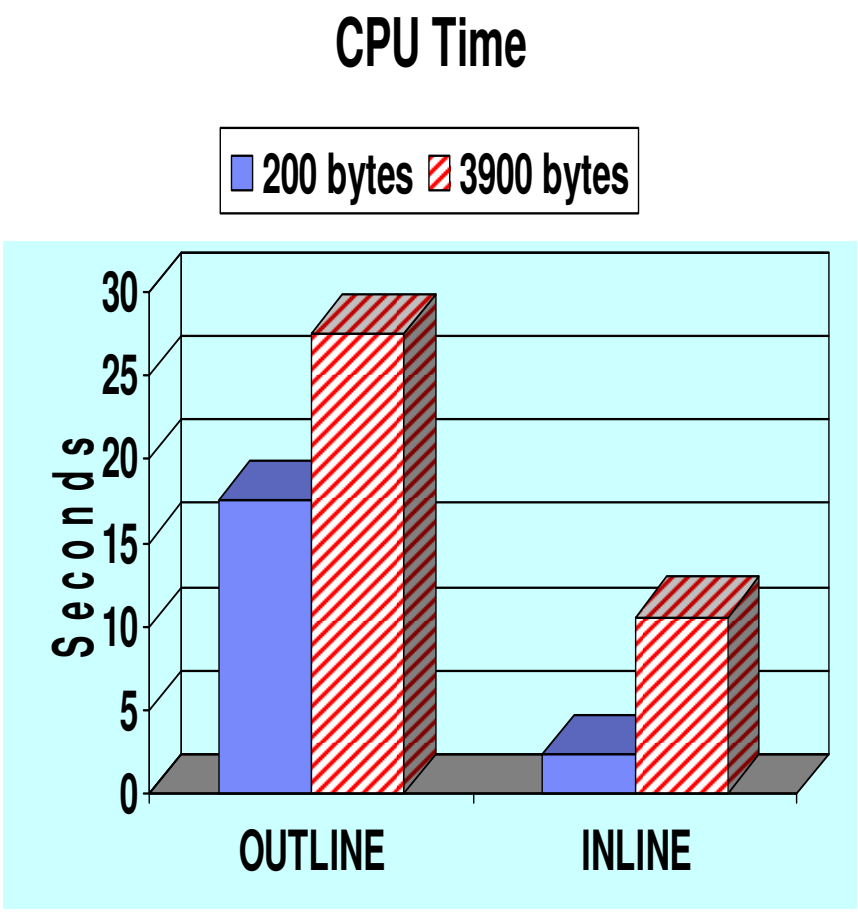
Random Access to small LOBs



70% faster random Selects, 90% faster Inserts

Faster LOAD REPLACE of small LOBs

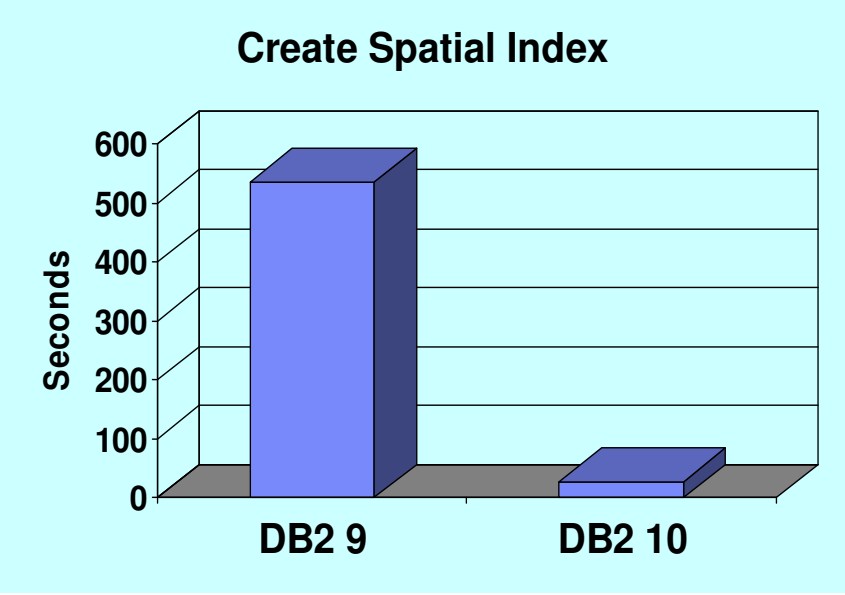


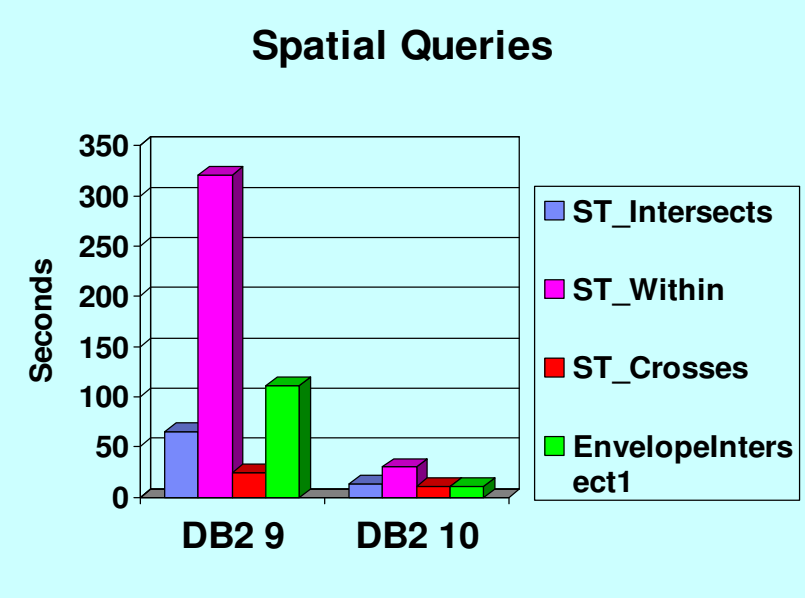


50-95% elapsed time reduction, 60-85% CPU reduction

Spatial Support

- Spatial Support was introduced in DB2 9 to manage geospatial information
- DB2 10 Spatial Support improves performance by exploiting Index on Expression with inline LOBs

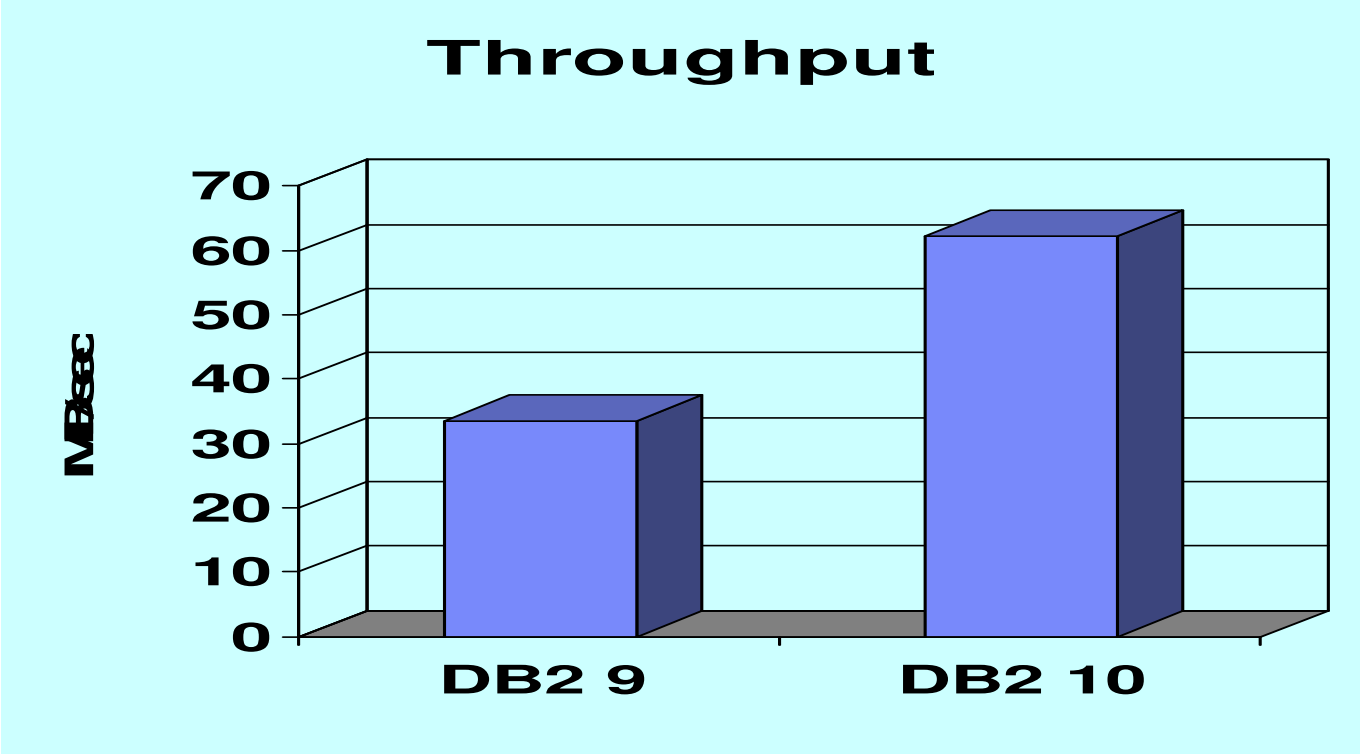




Reference: http://www.esri.com/library/brochures/pdfs/mainframeadvantage.pdf

Faster LOAD REPLACE of large LOBs

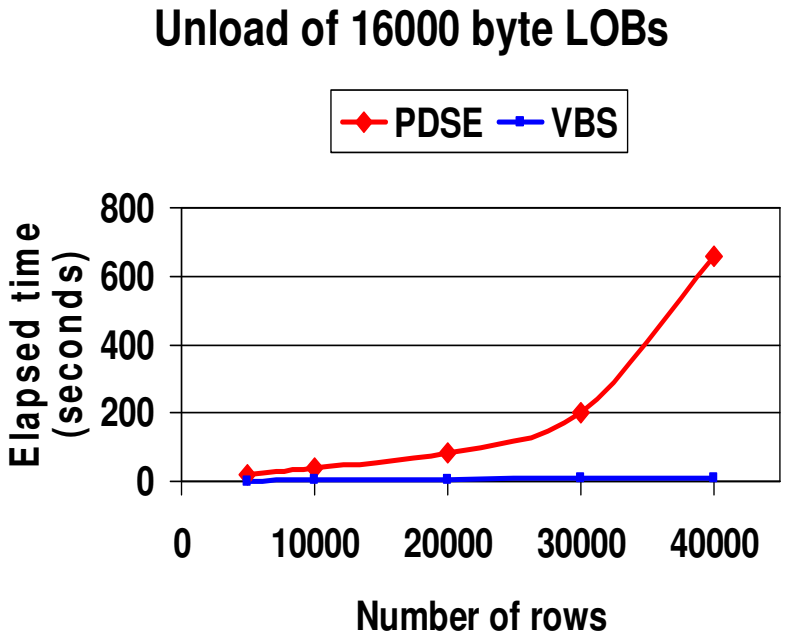
Load 20K LOBs



Throughput is doubled

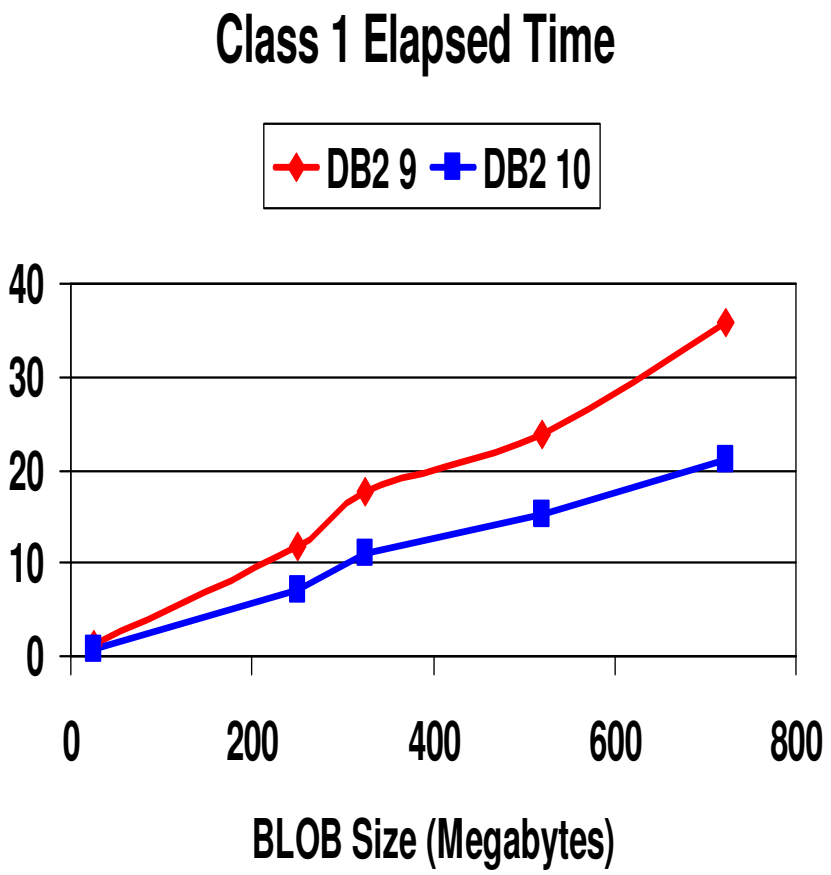
Load/Unload support for spanned records (RECFM=VBS)

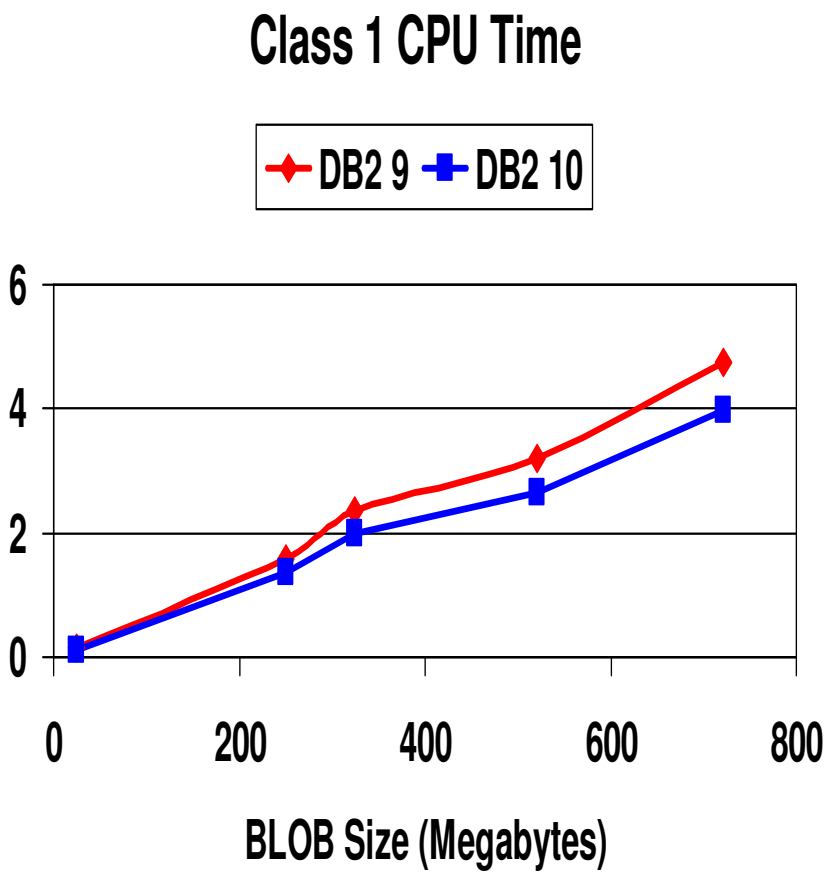
- Alternative to LOB File Reference Variables when LOBs are > 32K
- Support for sequential VBS data sets
  - Can be multi-volume
  - Can be compressed and striped on DASD
  - Can be stored on tape



100,000 LOBs took almost three hours

LOB Materialization Avoidance
DRDA BLOB Inserts





- Limit DBM1 storage consumption to 2MB per LOB

XML performance improvement

- Significant Performance improvement in V9 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
  - XML Update
    - No more full document replace

DB2 10 Monitoring Enhancements and Changes

1. New Monitor class 29 for statement detail level monitoring
   - IFCID 316/318 for dynamic, 400/401 for static
2. EXPLAIN MODE special register for remote package monitoring
3. Record index split with new IFCID 359
4. Separate accounting to identify DB2 latch and transaction lock in class3
5. Package LAST USED
6. Storage statistics(IFCID225) for DIST address space
7. Accounting : zIIP SECP values
   - Possible redirection value is no longer supported, always zero
   - SE CPU (actual offloaded CPU time) continues to be available

DB2 10 Monitoring Enhancements and Changes

8. Package accounting information with Rollup

9. Statistics trace interval
   - Always 1 minute interval in V10 no matter what you use in STATIME

10. Compression for DB2 trace data in SMF with a new zparm (SMFCOMP)
    - Overhead is minimum
    - 70-80% SMF data set saving from preliminary measurements
    - Trace formatter needs to be modified to call z/OS compression to decompress the data

Performance Enhancements with Few Changes (CM)

- SQL runtime improved efficiency
- VSCR
- Distributed thread reuse High Performance DBATs
- Buffer pool enhancements
- Improved dynamic prefetch (Row Level Sequential Detection)
- Index list prefetch for disorganized indexes
- Parallel index updates during Inserts
- DB2 9 utility enhancements in CM8
- Workfile in-memory enhancements

Performance Enhancements requiring REBIND (CM)

- Most access path enhancements
- 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
- 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
- REBIND enables further SQL runtime improvements
- If migrate from V8, get new RUNSTATS before mass rebind

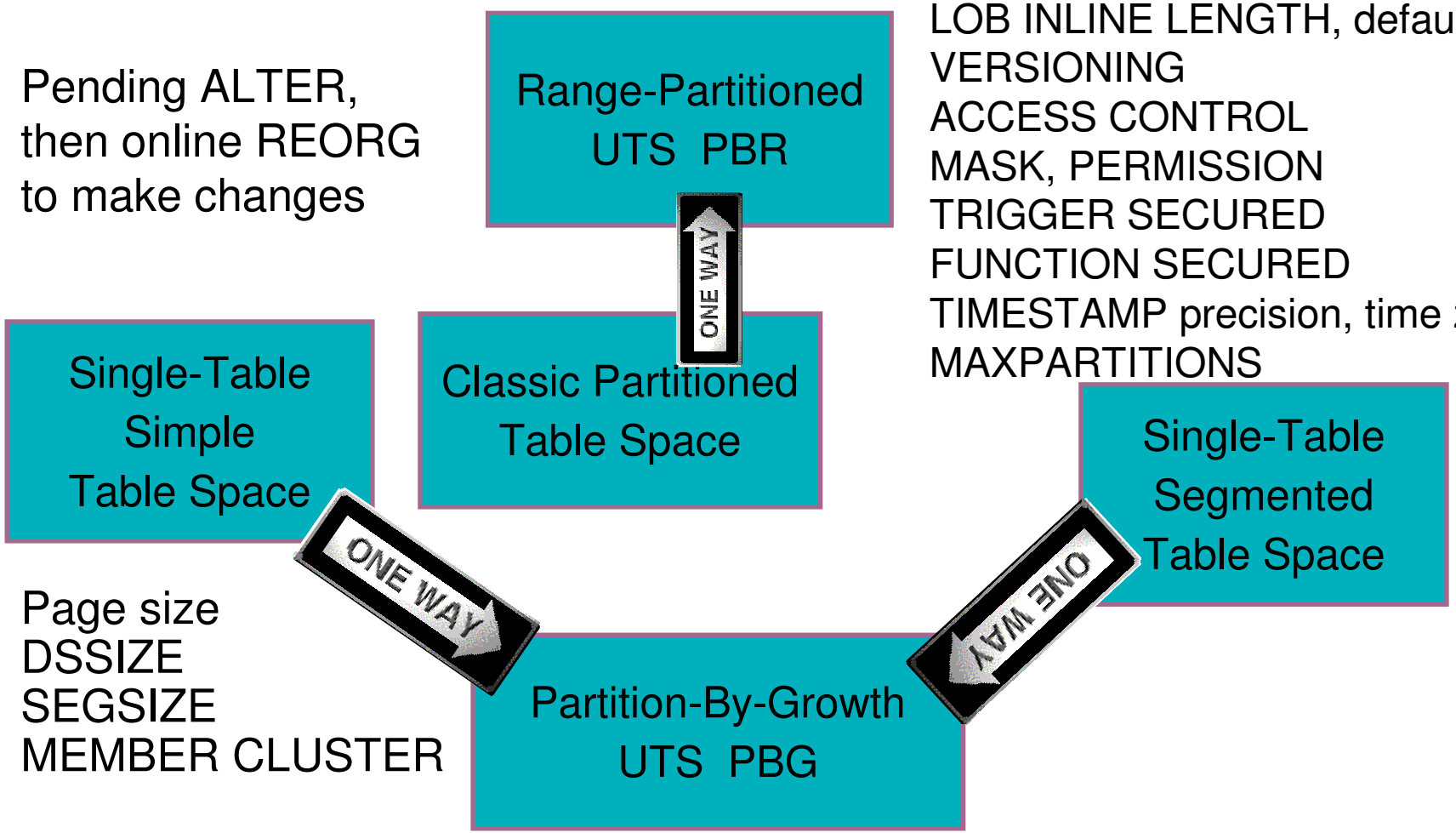
Performance Enhancements requiring NFM

- Efficient caching of dynamice SQL statements with literals
- Most DB2 10 utility enhancements
- LOB materialization avoidance in DBM1
- Faster fetch and insert, lower virtual storage consumption
- SQL Procedure Language performance improvements
- Workfile spanned records, PBG
- Insert improvement for UTS
- Local JDBC (type 2) and ODBC application performance
  - Limited block fetch, LOB progressive streaming, progressive CLOSE now working with JCC type 2 and ODBC z/OS drivers

Enhancements requiring NFM + DBA work

- Hash acess
- Index Include columns
- Inline LOBs (small)
- MEMBER CLUSTER for UTS
- DEFINE NO for LOB and XML columns
- Alter to UTS, page size, DSSIZE, SEGSIZE
- Online reorg all catalog & directory table spaces

Improved availability ALTER REORG



UTS Enhancements

- Provide ability to add new partitions to PBG
  - Enables DSN1COPY to be used for new partitions
  - ALTER TABLE… ADD PARTITION;
    - Aux objects implicitly created
    - Single partition at a time
- Provide ability to create multiple partitions on PBG Create
  - Primarily to support hash access

    CREATE TABLESPACE…..MAXPARTITIONS 10 NUMPART 10….
- Deprecate classic partioned table spaces
  - Creation of new classic partitioned table spaces requires explicit specification of SEGSIZE 0
  - Old syntax will create PBR
    - Default SEGSIZE for UTS in 10 is 32

REORG AUX YES Enhancement

- UTS or classic partitioned
- AUX YES automatically reorgs the LOB table spaces and allows REORG to move a row across partitions when the row contains a LOB column
- Essential for PBG
- Allows REBALANCE when LOBs are used
- Default is AUX NO unless:
  - Multi-part REORG of PBG with LOB columns
  - REBALANCE of PBR/class partitioned with LOB columns
  - REORG of PBR/classic partitioned with multiple parts in REORP
- Warning: REORG AUX YES may be slower than REORG LOB, unless the aux index buffer pool is large enough to contain the aux index

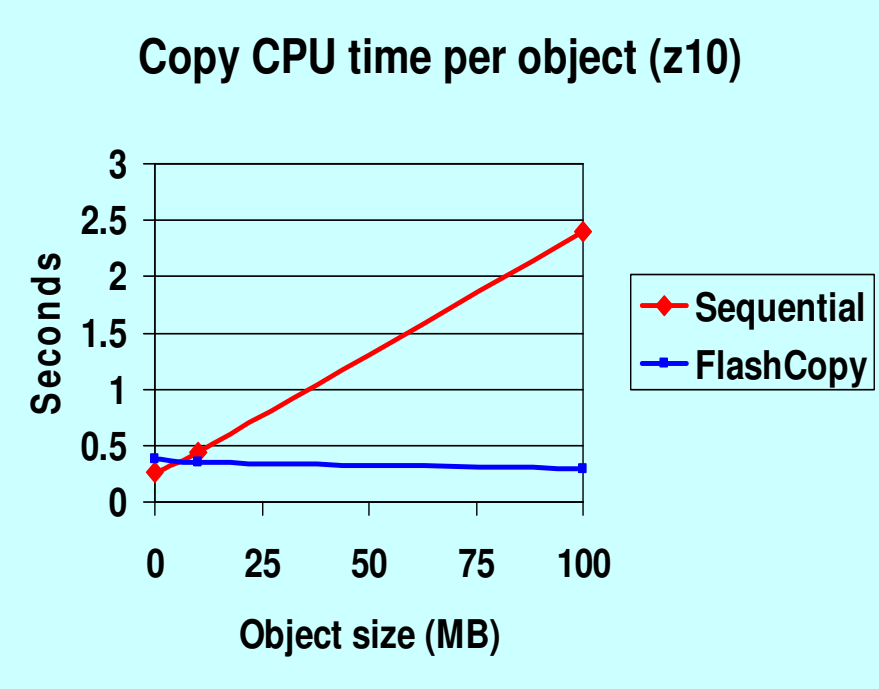
FlashCopy Enhancements

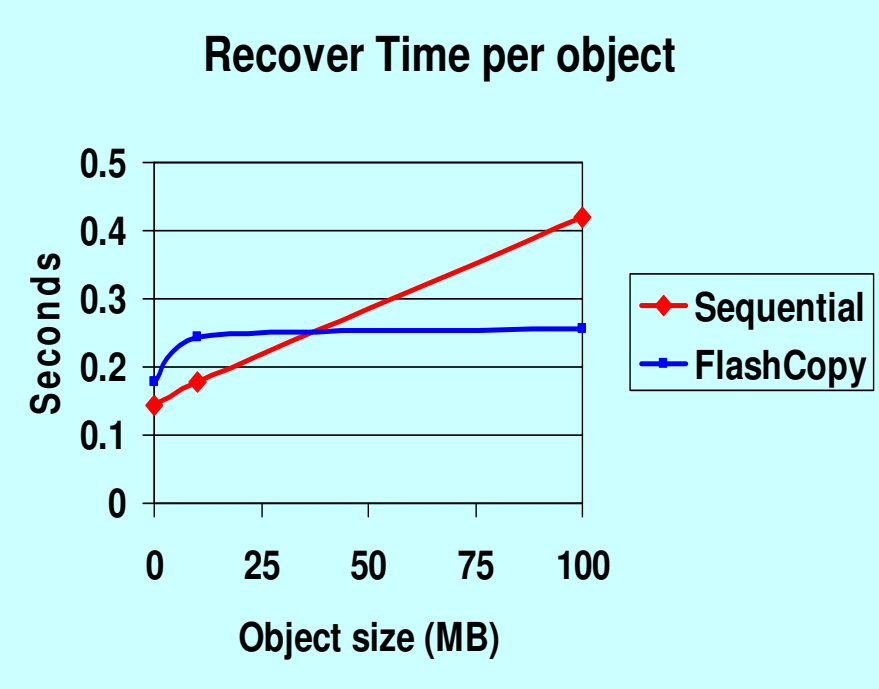
- Data set level FlashCopy supported by COPY/RECOVER and for inline copies (LOAD/REORG/REBUILD)
  - Can combine with sequential copy if required
- Allow creation of CONSISTENT copies with no outage
- FlashCopy can be used as input to:
  - RECOVER, COPYTOCOPY, DSN1COPY, DSN1PRNT
- FlashCopy does not support incremental copies and FlashCopies cannot be written directly to tape

FlashCopy Performance

- For large objects, lower CPU time and faster recovery
- Class 1 CPU time does not capture FlashCopy CPU time. Must use total system to determine CPU cost.
- REUSE not supported by Recover
  - Scratch and data set creation is the most significant part of the performance when restoring small objects
  - Volume level backup/recovery avoids this problem

FlashCopy Performance (without REUSE)





- FlashCopy adds 50% CPU time for empty objects, but it saves lots of CPU time (compared to *full image* sequential copies) for objects greater than ~7 MB.
- Recover is slower for objects less than about 40 MB.
- The effect of FlashCopy on OLTP or log apply performance is unknown

PIT Recovery

- New BACKOUT option on RECOVER
  - Roll back on log from current point instead of restoring recovery base and rolling forward
  - Works with PIT consistency, so changes prior to logpoint may be backed out
  - Can only be done once for a given log range

Other Utility Enhancements

- COPY CHANGELIMIT
  - Delay allocating output data set until CHANGELIMIT checked
  - &ICTYPE in template will no longer be “C”. Instead will reflect the correct type of image copy
- Incremental copies
  - Delay allocating input/output data set until pages to be copied are found
  - Insert dummy SYSCOPY record to register empty IIC
- REORG SHRLEVEL CHANGE for LOB page sets
- REORG SHRLEVEL NONE for LOBs is a no-op

New system authorities details

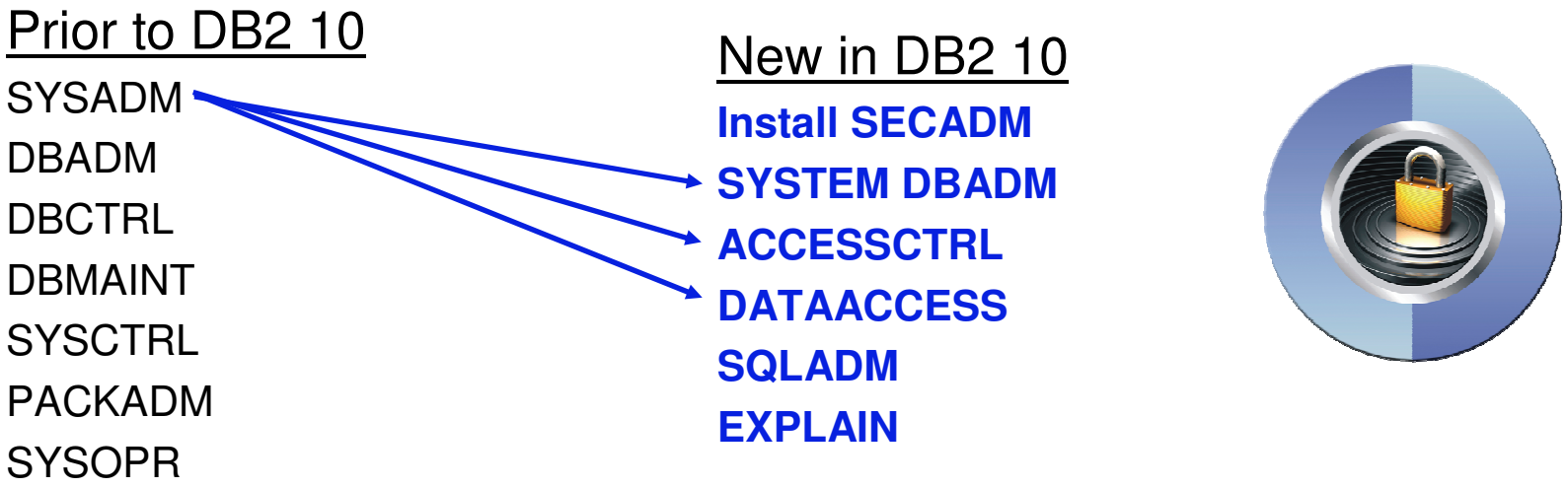
- SECADM - Install security administrator authority
  - Security related tasks (GRANT, REVOKE, ROLE)
- DBADM - System level database administrator authority
  - Perform DDL for all data bases
- DATAACCESS - System level data access authority
  - Access data in all tables and can execute plans, packages, functions and procedures
- ACCESSCTRL - System level access control authority
  - Issue GRANT or REVOKE for any object

New row and column level access controls

- New controls to protect access to individual rows
- New controls to protect access to sensitive columns

Minimize the use of SYSADM

- New granular system authorities and install security parameters



- Prevent SYSADM and SYSCTRL from granting or revoking privileges
  - New separate security install parameter
  - New install SECADM has grant and revoke privileges
- Control cascading effect of revokes
  - New revoke dependent privileges install parameter
  - New revoke dependent privileges SQL clause

Temporal tables

- Two types of time sequences of table rows are supported through the introduction of database defined time periods
- **SYSTEM_TIME** is used for system maintained history for a new concept of “versioning” which archives old rows into a history table
- **BUSINESS_TIME** is a period that represents when a row is valid to the user or application. The BUSINESS_TIME period can be used to model data in the past, present, and future as the data values are controlled by the application
  - A unique index can be defined for BUSINESS_TIME period to enforce non-overlapping time periods
  - The enforcement of uniqueness over a period of time is the important functionality delivered by BUSINESS_TIME periods
- A bitemporal table includes both SYSTEM_TIME and BUSINESS_TIME

Bitemporal example

- Database consists of employee salary
- Scenario:
  - An employee was hired on April 1 with a salary of $100K
    - Business start time is February 1
  - On June 15, his salary was raised to $102K effective July 1
    - The base table will have one row showing the salary from April 1 to July 1, and another row showing the new salary starting July 1
    - The history table has one row showing what happened on May 15
  - On July 15, the employee's salary is raised to $103K, retro-active to July 1
    - The second row in the table is updated
    - The history table has two rows, demonstrating both the first raise and the retroactive raise

Bitemporal example: Employee salary database

On March 20, an employee was hired, to be effective April 1, with a starting salary of $100K

EMP_DB

Name	Salary	Bus_start	Bus_end	Sys_start	Sys_end
Doe, John	$100K	04-01-2010	12-31-9999	2010-03-20.14.12.00	9999-12-31.00.00.00

EMP_HIST

Name	Salary	Bus_start	Bus_end	Sys_start	Sys_end

On June 15, his salary was raised to $102K effective July 1

EMP_DB

Name	Salary	Bus_start	Bus_end	Sys_start	Sys_end
Doe, John	$100K	04-01-2010	07-01-2010	2010-03-20.14.12.00	2010-06-15.10.48.00
Doe, John	$102K	07-01-2010	12-31-9999	2010-06-15.10.48.00	9999-12-31.00.00.00

EMP_HIST

Name	Salary	Bus_start	Bus_end	Sys_start	Sys_end
Doe, John	$100K	04-01-2010	12-31-9999	2010-03-20.14.12.00	9999-12-31.00.00.00

On July 15, the employee's salary is raised to $103K, retro-active to July 1

EMP_DB

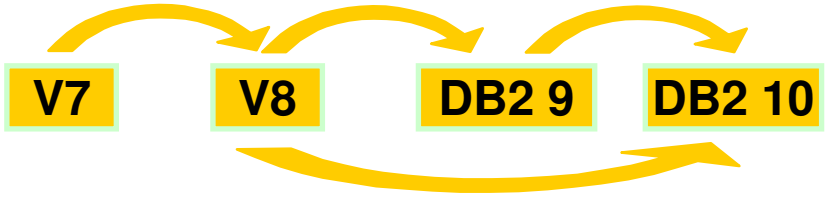
Name	Salary	Bus_start	Bus_end	Sys_start	Sys_end
Doe, John	$100K	04-01-2010	07-01-2010	2010-03-20.14.12.00	2010-06-15.10.48.00
Doe, John	$103K	07-01-2010	12-31-9999	2010-06-15.10.48.00	9999-12-31.00.00.00

EMP_HIST

Name	Salary	Bus_start	Bus_end	Sys_start	Sys_end
Doe, John	$100K	04-01-2010	12-31-9999	2010-03-20.14.12.00	2010-06-15.10.48.00
Doe, John	$102K	07-01-2010	12-31-9999	2010-06-15.10.48.00	9999-12-31.00.00.00

DB2 10 for z/OS: Skip-Level Migration

- May move from V8 to DB2 10, but just because you can doesn't mean you should
- Migration, fallback and data sharing coexistence fully supported
  - Mix of DB2 9 and 10 or DB2 V8 and 10
- Key considerations
  - Risk/reward analysis
    - What's your risk? Tolerance level?
    - How will you do it? What's your mitigation plan? Are ISVs ready?
    - What workloads do you need to test can can you test them properly?
    - Are you missing out on DB2 9 value in the meantime?
  - Migration cost savings is not 2X versus two migrations
    - Migration considerations for two versions still apply
    - Larger migration project, longer migration timeline, more risk
    - Applications and ISVs may not be ready



DB2 10 Performance/Scalability Summary

- Up to 10% CPU reduced after REBIND
- 70-90% of DBM1 Below the Bar storage reduction
- Higher CPU improvement for the sweet spots or workloads with scalability issues (storage, DB2 latches)
  - SQL PL
  - DDF using high performance DBAT
  - Simple queries evaluating multiple rows thru indexes
  - Concurrent sequential insert
  - Small LOBs
- Potential for more threads per member and for data sharing member consolidation

Key details about DB2 10: getting ready

Prerequisites: migrate from DB2 9 for z/OS or DB2 for z/OS V8

- z/OS V1.10 SMS-managed DB2-managed DB2 catalog
- System z10, z9, z890, z990, and above (no z800, z900)
- DB2 Connect 9 FP1, 9.7 FP3 for many 10 functions
- IMS 10 & 11 (not 9) CICS compilers (See announcement)
- SPE PK56922 PK69411 PK61766 PK85956 PM04680 PK87280 PK87281 PM08102 PM08105 PM10227
- Premigration check DSNTIJPA PM04968
- Info APARs II14477 (DB2 9) II14474 (V8)

Items deprecated in earlier versions eliminated: more for V8 mig.

- Private protocol → DRDA (DSNTP2DP, PK92339, PK64045)
- Old plans and packages V5 or before → REBIND
- Plans containing DBRMs → packages PK62876 (9), PK79925 (8) PK85833 (9), PM01821 (All)
  - ACQUIRE(ALLOCATE) → ACQUIRE(USE)
- Old plan table formats → DB2 V8 or 9, Unicode, 59 cols PK85068
- BookManager use for DB2 publications → Info Center, pdf

DB2 9 in IBM Redbooks Publications

1. DB2 9 Technical Overview SG24-7330
2. DB2 9 Performance Topics SG24-7473 update just made
3. DB2 9 Stored Procedures SG24-7604
4. Index Compression with DB2 9 for z/OS redp4345
5. SQL Reference for Cross-Platform Development
6. Enterprise Database Warehouse, SG24-7637
7. 50 TB Data Warehouse on System z, SG24-7674
8. New Tools for Query Optimization SG24-7421
9. LOBs with DB2 for z/OS SG24-7270
10. Deploying SOA Solutions SG24-7663
11. Enhancing SAP - DB2 9 SG24-7239
12. SAP Application on Linux z SG24-6847
13. Best practices SAP BI - DB2 9 SG24-6489-01
14. Data Sharing in a Nutshell, SG24-7322
15. Securing DB2 & MLS z/OS SG24-6480-01
16. Data Sharing: Distributed Load Balancing & Fault Tolerant Configuration redp4449
17. Considerations on Small & Large Packages redp4424
18. Backup and Recovery Considerations redp4452
19. Powering SOA with IBM Data Servers SG24-7259
20. Packages Revisited, SG24-7688
21. Data Studio V2.1 Web Services redp4510
22. Ready to Access Solid-State Drives redp4537
23. Distributed Functions SG24-6952
24. Buffer Pool Monitoring & Tuning redp4604
25. Securing & Auditing Data SG24-7720
26. Serialization and Concurrency SG24-4725-01 new
27. Utilities SG24-6289-01 newest
28. Integrated Data Management Service Level Objectives, SG24-7769