



DB2 for z/OS V10 Topics

Kevin Harrison
Sr. IT Architect / Sr. Software Engineer
DB2 for z/OS Development
kharrison@us.ibm.com



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*	DataPower*	IMS	Rational*	Tivoli*
IBM (logo)*	CICS*	Lotus*	System Storage	WebSphere*
ibm.com*	DB2*	POWER7	System x*	XIV*
AIX*	DS4000*	ProtectTIER*	System z*	zEnterprise
BladeCenter*	FICON*	RACF*	System z10	z/OS*
				z/VM*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

InfiniBand is a trademark and service mark of the InfiniBand Trade Association.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput 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 improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Disclaimer

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.

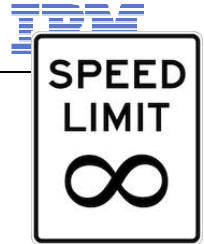
DB2 10 for z/OS



- **Base Pre-requisites:**
 - zSeries z890, z990, z9, z10, z196 or later
 - z/OS V1.10 or above
 - Defined shared memory objects (V9)
- **Catalog changes:**
 - Additions for new features
 - Hashes and links removed
 - Many tables changed to:
 - Single table, table spaces (UTS, PBG)
 - Row level locking
 - Using Inline LOBs
- **64-bit ODBC driver for z/OS**
- **DB2 Connect**
 - V9 FP1
 - V9.7 FP3a for new features
 - PM24292 for Sysplex Workload Balancing
- **Migration Process:**
 - From Version 8 or 9 NFM
 - Data sharing coexistence in CM8 or CM9
 - DSNTIJPA in V8 or V9 with PM04968

DB2 10 for z/OS Skip Level Migration

- Allows a customer to migrate from DB2 for z/OS V8 NFM to DB2 10
- Is this right for you?
- Considerations for skip level migration
 - **Compare the maturity of the versions (time in the field)**
 - **Support Margin: V8 EOS date is yet unknown. When announced, will there be enough time overlap for a comfortable migration timeframe?**
 - **Accumulated incompatibilities that may add to the V10 migration**
 - **Magnitude of the migration vs. the migrations costs**
 - **Some “unsung” V9 benefits which could add value now**
 - **Shared Memory Objects enhancing distributed communications (happens in CM).**
 - **Major financial company reported a 20% reduction in distributed application CPU**
 - **Native Stored Procedures**
 - **Easy development, debugging, and deployment**
 - **zIIP enabled via DRDA**
 - **Good performance boost opportunity**
 - **Many Utility CPU savings (happens in CM)**



DB2 10 Sample Performance Improvements

■ DB2 10 CMx with REBIND

- Run time CPU reductions 5% - 10%
- 1 MB page size 0% - 4% z10, z196
- Page fix buffers 0% - 8% since V8
- Release deallocate 0% - 15% short trans, batch
- Virtual storage constraints 0% - 5% memory, latches
- Data sharing fewer members 1% for each 2 members
- Insert 0% - 40% high volume insert
- Predicate evaluation 0% - 60% complex predicates
- Increased use of zIIP 0% - 3% IO, RUNSTATS, parallelism
- Utilities (from V8) 3% - 20%

■ DB2 10 NFM

- Improved dynamic SQL cache 0% - 20% literals
- Access: hash, index include 0% - 5% access improved



SQL Enhancements

- **Access Currently Committed**
- **Extended Indicator Variables**
- **More OLAP Specifications**
- **Greater Timestamp Precision**
- **Timestamp with Time Zone**

Access Currently Committed...

■ **SELECT not blocked by INSERT:**

- Pre-DB2 10 SELECT waits and eventually a row returned or times out.
- DB2 10 can see row being inserted is not committed, and would immediately skip row.
- Relation to SKIPUNCI ZParm for uncommitted INSERTs:
 - Bind option or PREPARE attribute value controls outcome when specified
 - SKIPUNCI value controls outcome when not defined

■ **SELECT not blocked by DELETE:**

- Pre-DB2 10 SELECT waits and eventually no row is returned or times out.
- Currently committed row, including any LOB or XML data is returned, until DELETE is committed.

Extended Indicator Variables

■ Pre-DB2 10 Problem/Requirements:

- No way to tell DB2 what value to use in an INSERT, UPDATE or MERGE when values for all the columns are not specified.
- Need to define all possible SQL combinations... very tedious
- Very low use of dynamic statement cache.

■ DB2 10 Solution:

- DB2 10 NFM introduces extended indicator variables to specify what value should be used for columns when a value is not provided in an INSERT, UPDATE or MERGE.
 - New values used for existing indicator variables.
- Simplifies application development... no need to define all possible SQL combinations
- Improves overall performance

Extended Indicator Variables

■ Implementation:

- EXTENDEDINDICATOR YES/NO option on BIND/REBIND PACKAGE
 - NO default for BIND.
 - Existing setting is default for REBIND
 - New EXTENDEDINDICATOR column added to SYSPACKAGE
- WITH EXTENDED INDICATORS or WITHOUT EXTENDED INDICATORS attribute string on PREPARE
 - Default - WITHOUT EXTENDED INDICATORS
- Various methods available for implementing support with JDBC, SQLJ and ODBC applications.
- Extended indicator variable values:
 - -5 specifies the DEFAULT value – set column to the default value
 - -7 specifies the UNASSIGNED value - treat as column was not specified

More OLAP Specifications

■ Moving Sum and Moving Average

- New DB2 10 NFM specifications (aggregate functions) that compute a single value for the current row based on some or all the rows in a defined group.
- Supports cumulative sums and moving averages by using a window.
- A window can specify three optional components:
 - Partitioning of the result table using PARTITION BY clause (in DB2 9).
 - Ordering of rows within a partition using ORDER BY clause (in DB2 9).
 - Aggregation group by using ROW or RANGE clause
 - Two mandatory components need to be specified for an aggregation group.
 - Keyword ROWS to indicate a physical group ... or
 - Keyword RANGE to indicate a logical group ... and
 - The starting and ending row of the aggregation group
- Cannot be used within an XMLQUERY, XMLEXISTS, or as an argument of an aggregate function.
- New -20117 SQL code to indicate an invalid window specification.

Greater Timestamp Precision

- **Number of digits for the fractional second in a timestamp extended**
 - The DB2 9 default of 6 digits remains
 - `TIMESTAMP` is the same as `TIMESTAMP(6)`
 - String representation:
`yyyy-mm-dd-hh.mm.ss.nnnnnn`
 - Range supported in DB2 10 NFM is 0 to 12 digits
 - E.g. `TIMESTAMP(12)` is the maximum
 - String representation:
`yyyy-mm-dd-hh.mm.ss.nnnnnnnnnnnn`
 - Other capabilities like timestamp duration and `CURRENT_TIMESTAMP` extended to support new precision capability



Timestamp with Time Zone

- **New data type `TIMESTAMP WITH TIME ZONE`**

- `SET SESSION TIME ZONE` or rely on `IMPLICIT_TIMEZONE (DECP)`
- Year.month.day.hour.minutes.seconds +-timezone

```
CREATE TABLE TABLE1 (C1 TIMESTAMP WITH TIME ZONE, C2 INTEGER);  
INSERT INTO TABLE1 VALUES ( '2007-11-05-08.00.00-08:00', 1);  
INSERT INTO TABLE1 VALUES ( '2007-11-05-11.00.00-05:00', 2);  
INSERT INTO TABLE1 VALUES ( '2007-11-05-11.00.00', 3);
```

- **Comparison takes place after converting internally to UTC**

- '2009-11-05-08.00.00 -08:00' is converted to '2009-11-05-16.00.000000'
- '2009-11-05-11.00.00 -05.00' is converted to '2009-11-05-16.00.000000'
 - `SELECT C2 FROM TABLE1 WHERE C1 = '2009-11-05-10.00.00 -06:00';`
 - would retrieve all 3 of the above
 - '-6:00' is the timezone, US Central which is UTC -6, except during daylight savings time

64-bit ODBC driver for z/OS

- Provides the ability for 64-bit ODBC applications to run on z/OS, and take advantage of the new addressability.
 - In DB2 9 with PK83072.
- New driver benefits:
 - runs in AMODE(64)
 - reduces virtual storage constraint
 - can accept 64-bit user data pointers
 - access user data above the 2GB bar in the application address space
 - XPLINK only
 - Shipped in addition to the 31-bit ODBC driver.
- 31-bit ODBC driver is XPLINK and non-XPLINK. The APIs for existing 31-bit applications have not been changed, and continue to work using the 31-bit ODBC driver.
- Consider migrating 31-bit applications to 64-bit, if the application can take advantage of more than 2GB of memory.
 - **This is a migration**
[Migrating an ODBC 31-bit application to a 64-bit application](#)
- The XPLINK driver is recommended to enhance performance, but only if your ODBC application uses XPLINK code exclusively.
- For more information: [DB2 ODBC application requirements](#)

pureXML, The Sequel

- **Multi-Versioning Support**
- **Subdocument Update**
- **Validation Improvements**
- **Native SQL Procedures & UDFs**
- **XML Date and Time**

Multi-Versioning (MV) Support...

- **Requires the base table to be a Universal Tablespace (UTS)**
- **Existing tables with XML columns must be dropped and recreated after DB2 10 NFM for MV format.**
- **MV format required for several V10 features:**
 - XMLMODIFY
 - Temporal data (“**AS OF**”)
 - Currently Committed
 - SELECT FROM OLD TABLE (V9 feature)
- **The MV format reduces locking**
 - V9
 - XML data is not kept with base row data in the work files
 - Therefore a lock is maintained on the XML row until it is processed after work file
 - MV eliminates the need for this lock except for UR Readers (DOCID lock). Avoids a UR Reader from getting an incomplete document.

Subdocument Update with XMLMODIFY...

- **DB2 9 pureXML supported the updating of an XML column**
 - But the entire column needed to be replaced
- **DB2 10 delivers the XMLMODIFY function to:**
 - Insert nodes within an existing document
 - Replace existing nodes of a document
 - Delete nodes from an existing document

- **Requires MV format**

```
DSNT408I  SQLCODE = -4730, ERROR:  INVALID SPECIFICATION OF XML COLUMN  
XML_MV_TEST_FROMV10.XMLCOL IS NOT DEFINED IN THE XML VERSIONING  
FORMAT, REASON 1
```

XML Date & Time Support

- **DATE and TIME are not supported types in DB2 9 XPath**
- **DB2 10 provides**
 - xs:dateTime
 - xs:time
 - xs:date
 - xs:duration
 - xs:yearMonthDuration
 - xs:dayTimeDuration
 - Several functions (fn) for comparison, value extraction, arithmetic operations
- **Implicit time zone will be UTC if not specified**
- **These types can be used in the creation of XML Value Indexes**

What is temporal data?

- **One of the major improvements in DB2 10 will be the ability for the database to reduce the complexity and amount of coding needed to implement “versioned” data, data that has different values at different points in time.**
- **Data that you need to keep a record of for any given point in time**
- **Data that you may need to look at for the past, current or future situation**
- **The ability to support history or auditing queries**
- **Supporting Business Time and System Time**

Basic Temporal Concepts

- **Business Time (Effective Dates, Valid Time, From/To-dates)**
 - Every row has a pair of TIMESTAMP(6) or DATE columns [set by Application](#)
 - Begin time : when the business deems the row valid
 - End Time : when the business deems row validity ends
 - Constraint created to ensure Begin time < End time
 - Query at current, any prior, or future point/period in business time
- **System Time (Assertion Dates, Knowledge Dates, Transaction Time, Audit Time, In/Out-dates)**
 - Every row has a pair of TIMESTAMP(12) columns [set by DBMS](#)
 - Begin time : when the row was inserted in the DBMS
 - End Time : when the row was modified/deleted
 - Every base row has a Transaction Start ID timestamp
 - Query at current or any prior point/period in system time
- **Times are inclusive for start time and exclusive for end times**

Basic Temporal Concepts

■ **Bi-temporal**

- Inclusion of both System Time and Business Time in row

■ **Temporal Uniqueness**

- PK or Unique Key with BUSINESS_TIME WITHOUT OVERLAPS
- Support for a unique constraint for a point in time
- This is optional, however without it:
 - Unique constraints will likely return errors due to multiple rows per key
 - If there's no key and System Time is used
 - All history table rows are considered new additions
 - Not associated with previous rows with same column values

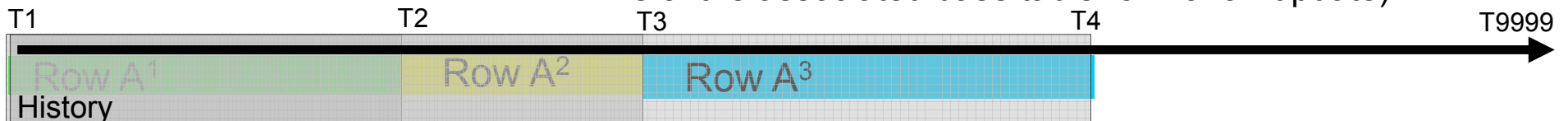
■ **History Table**

- Table to save “old” rows when using System Time

Row Maintenance with System Time

- **No temporal syntax for System Time maintenance**
 - Use regular Update, Delete, Insert statements
- **If the modification impacts existing base table rows**
 - Insert or Update
 - The base table row(s) are created / updated with a current System Start Time and high value System End Time.
 - Delete
 - Remove the base table row.
 - Update or Delete
 - Create a “before-image” copy of all qualified base table rows in the History Table.
 - If a previous copy of a row exists in the History table
 - Limit the System timeframe of the previous copy
 - Set previous copy’s End System Time = copied row’s Begin System Time
 - The newly created History row(s) are added with a System End Time equal to the current time (System Start Time of the associated base table row for an update)

Delete Row A @ T4



Row Maintenance with Business Time

- **Temporal syntax is used for Business Time maintenance**
 - FOR PORTION OF BUSINESS_TIME FROM x TO y
- **If the modification impacts existing base table rows**
 - Insert
 - If a PK includes Business Time check for overlaps for the same PK of different base table rows
 - 803 returned if overlaps are found
 - Update / Delete
 - Insert the base row with the specified Begin & End Business Times
 - Check the specified Business Time against existing qualified rows
 - Rows **contained within** the specified Business Time range are updated / deleted
 - row Business Time remains unchanged for the update
 - Rows that **span the specified From OR To** Business Time are
 - Updates: split into two rows, and updates applied to the portion of Business Time within the From and To
 - Deletes: The Begin or End Business Time is updated so no portion of the specified range remains
 - Row that **span the specified From AND To** are split into:
 - Updates: three rows, and updates applied to the portion of Business Time within the From and To
 - Deletes: two rows representing the remaining Business Time on either end of the specified range

In a bi-temporal implementation any changes to existing rows would also go through the System Time steps on the prior slide

System Period Versioning Information...

- **Base and History tables must be RECOVERed as a set**
 - VERIFYSET NO can override the need to RECOVER together
- **No utility operations that deletes data from base table**
 - LOAD REPLACE
 - REORG DISCARD
 - CHECK DATA DELETE YES
- **No CHECK utilities that invalidate AUX/LOB/XML**
- **Cannot ALTER the schema while versioning**
- **The Base and History table, table spaces must be single table**
- **No temporal SELECT, UPDATE, or DELETE against the History**
- **Cannot be an MQT**
- **Cannot have a Clone Table, Column Mask, Row Permission, or Security Label column**

System Period Versioning Information...

- **Cannot TRUNCATE**
 - INSERT, UPDATE, DELETE, and MERGE are accepted
- **To find the Base / History Tables**

- **System Time can be altered on an existing table**
 - See Information Center topic:
http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.admin/db2z_addingsystime.htm
- **QUIESCE of the Base or History**
 - Will cause a quiesce against all tables in the versioning relationship, including auxiliary spaces

System Period Versioning Information

- **REPORT TABLESPACESET** identifies versioning relationships in the system-maintained temporal table space or history table space
- **CURSORs & VIEWs** referencing system temporal table with a period specification will be **READ ONLY**








Business Period Versioning Information

- **Business Time can be altered on an existing table**
 - See Information Center topic:
http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2z10.doc.admin/db2z_alteringtableaddbusitime.htm
- **Consider the implications of non-temporal UPDATE & DELETE statements**
 - These statements are allowed
- **SQLERRD(3) does not reflect rows added due to a temporal UPDATE / DELETE**
 - Consistent with RI handling
- **It is possible to have contiguous Business Time ranges with the same non-temporal data in the row**
- **FOR PORTION OF BUSINESS_TIME must specify valid Date/Time values in the FROM and TO**
 - Some documentation would indicate that the TO can be omitted
 - MINVALUE / MAXVALUE returns errors

Advanced Design Options

- **Table Space history**
- **Online Schema Evolution**
- **Implicit Casting**
- **Hash Access**

Universal Table Spaces

CREATE TABLESPACE...	SEGSIZE	NUMPARTS	MAXPARTITIONS	Comments
<p>■ What kind of Table Space will be created? (* optional)</p>				
Segmented	 *			*SEGSIZE is optional. Default for explicitly created TS & implicitly created TS for CM8. SEGSIZE defaults to 4.
UTS PBG	 *	Optional to indicate # of initial partitions		Default for CM9 and NFM with implicitly created TS. Single table TS. *SEGSIZE will default to 32.
UTS PBR	 *			Single table TS *SEGSIZE will default to 32.
Classic Partitioned TS	 *			Partitioning TS prior to V9 *SEGSIZE 0 will create classic partitioned and CM8 behavior is same as V8 NFM

Improved ALTER...

- **Pending changes materialized with an online REORG apply to UTS**
 - SEGSIZE – no other pending ALTERs can be done before this is materialized
 - DSSIZE – no other pending ALTERs can be done before this is materialized (IMPDSSIZE ZParm for default)
 - MEMBER CLUSTER – new for UTS
 - MAXPARTITIONS
 - If other pending changes are involved, or changing table space type: it is pending
 - Otherwise it is immediate
 - Page Size (BUFFERPOOL)
 - Can be done with REORG TABLESPACE (for indexes and tables) or REORG INDEX for only index changes
- **Other ALTERs are immediate**
 - The above statements if TS or IX not defined
 - MAXPARTITIONS (unless changing TS)
 - BUFFERPOOL PGSTEAL NONE
 - LOB INLINE LENGTH

Online Schema – Details on Execute ALTER Statement

- **Statement is validated**
 - Semantic checking against effective catalog definition
- **Assuming all checks out ok:**
 - Statement is put on pending list
 - Table space is placed in Advisory-REORG pending :**AREOR** (non-restrictive)
 - Not to be confused with REORG-pending advisory (**AREO***) which says access path could be degraded
 - Statement completes with SQLCODE +610 to advertise the advisory state

SYSIBM.SYSPENDINGDDL:

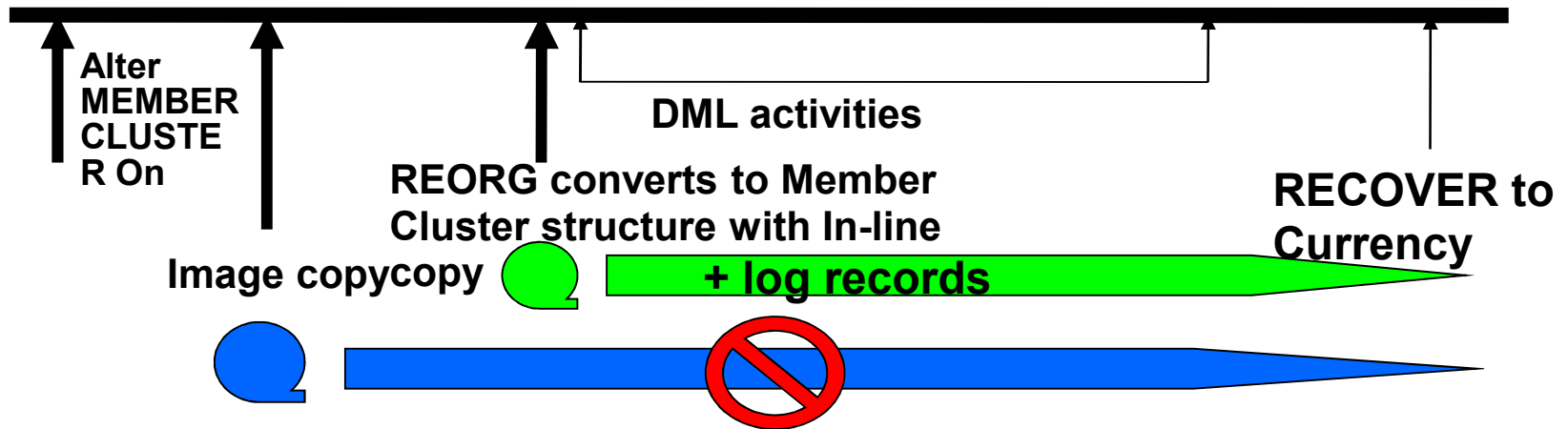
DBNAME	TSNAME	DBID	PSID	OBJSCHEMA	OBJNAME	...	OPTION_Keyword	OPTION_Value	...	STATE_Ment_Text

Online Schema – Details on Online REORG

- **Pending DDL is materialized - DSNU1163I**
 - Catalog & Directory are updated with the new attributes
 - Data sets are updated with the new attributes
 - Materialized SYSPENDINGDDL entries are removed
- **Stats are collected**
- **SYSCOPY entries show inability to recover object prior to changes**
- **AREOR state is reset**

Online Schema...

- **Restrict RECOVER across materializing REORGs**



- **Plans and packages are invalidated if changing table space type**

- When changing the MAXPARTITIONS attribute of a simple or segmented table space to convert it to a partition-by-growth universal table space
- The SEGSIZE attribute of a partitioned table space is changed to convert the table space to a range-partitioned universal table space

Table space enhancements

- You can now ROTATE any partition to last
- Also ALTER TABLE ADD PARTITION for partition by growth TS

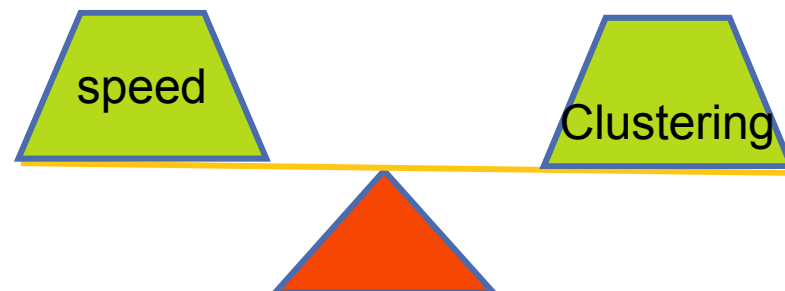
```
ALTER TABLE TAB1 ROTATE PARTITION 3  
TO LAST ENDING AT (2009 JUL') RESET;
```

logical	ts	pi	
Partition 1	2009 Jan		A001
Partition 2	2009 Feb		A002
	2009 Mar		A003
Partition 3	2009 Apr		A004
Partition 4	2009 May		A005
Partition 5	2009 Jun		A003

Compress on INSERT

- **Data compression occurs when a dictionary exists**
- **Prior to DB2 10**
 - Dictionary not built on a table space with COMPRESS YES attribute until:
 - REORG or
 - LOAD utility was executed
 - For some customers, REORG or LOAD are not executed frequently
 - LOAD COPYDICTIONARY offered in DB2 9
- **DB2 10 NFM allows for build of compression dictionary on:**
 - INSERT
 - MERGE
 - LOAD utility with REPLACE, RESUME NO, or RESUME YES SHRLEVEL CHANGE, and without KEEPDICTIONARY
- **Eliminate need for REORG or LOAD needed to build compression dictionary**

Member Cluster



■ MEMBER CLUSTER added to UTS

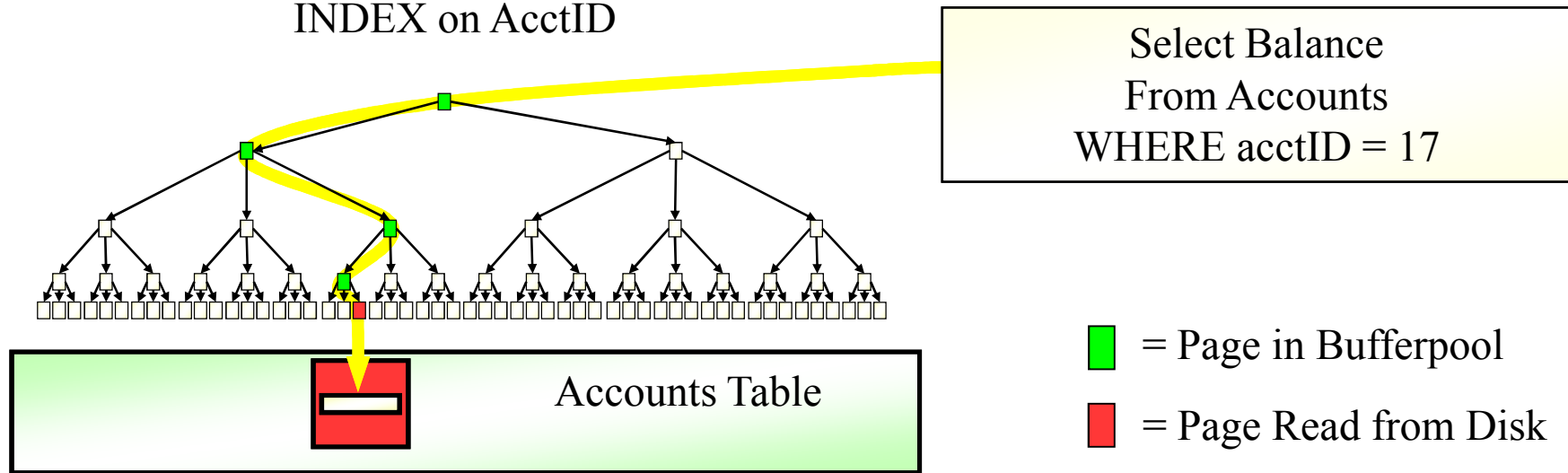
- Avoid hotspots, P-lock, and page latch contention
 - Usually only applicable for data sharing, concurrent access
 - Each member is assigned a set of space map pages
 - Ignores clustering index on insert, REORG to get clustering back
 - Space map covered 199 data pages in DB2 9
 - Space map covers 10 segments in DB2 10
- New column SYSTABLESPACE.MEMBER_CLUSTER
- Very important for APPEND tables (DB2 9)

Hash Access Candidates

- Candidate Tables
 - For queries needing single row access via the unique key
 - Queries having equal predicates on keys
 - With known and static approximate size of data
 - Having large N-level indexes
- Not for Tables
 - Needing sequential processing
 - Frequently updated
 - Either using BETWEEN or > and <
- Follow-up
 - Run REBIND with EXPLAIN option and query the PLAN_TABLE to check access path
 - SYSTABLESPACESTATS.REORGHASHACCESS
 - Number of times data is read using hash access in the last time interval
 - Check LASTUSED & REORGINDEXACCESS on overflow and other indexes to validate HASH access

Index to Data Access Path

INDEX on AcctID





- **Traverse down Index Tree**
 - Typically non-leaf portion of tree in the bufferpool
 - Leaf Portion of the tree requires I/O
 - Requires searching pages at each level of the index
- **Access the Data Page**
 - Typically requires another I/O
- **For a 5 Level Index**
 - 6 GETP/RELPS, 2 I/O's, and 5 index page searches

Hash Access

- Hash Access provides the ability to directly locate a row in a table without having to use an index
- Single GETP/RELP in most cases
- 1 Synch I/Os in common case
 - 0 If In Memory Table
- Greatly reduced Search CPU expense

```
Select Balance  
From Accounts  
WHERE acctID = 17
```



-  = Page in Bufferpool
-  = Page Read from Disk

Optimization Evolution

- **Literal Replacement**
- **Access Path Stability Improvements**
 - Optimization Hints
 - Plan Management Policies
- **Index Usage Improvements**
- **SQL Pagination (Complex OR)**
- **Dynamic Row Level Sequential Detection**
- **Parallelism**
- **Statistics Enhancements**
- **BIND / REBIND Changes**
- **EXPLAIN Changes**

Literal Replacement...

- **Dynamic SQL statements with literals can now be made more reusable with literal replacement**
- **Literals are replaced with a “&” and stored in DSC**
- **Enabled via:**
 - PREPARE: CONCENTRATE STATEMENT WITH LITERALS in the ATTRIBUTES
 - ODBC: LITERALREPLACEMENT specified in the initialization file
 - JCC Driver: enableLiteralReplacement='YES'
- **Requires 2 DSC lookups**
 - A DSC match for the original statement (with literals) happens first
 - If no exact statement match occurs, the literals are replaced and a second lookup occurs
 - This must match another statement with literal replacement, not parameter markers
 - Also requires:
 - Matching criteria similar to parameter markers (SQL text and auth ID)
 - The Literal Reusability Context must be the same.
 - The data type and lengths of the literal replacements must be compatible

Literal Replacement...

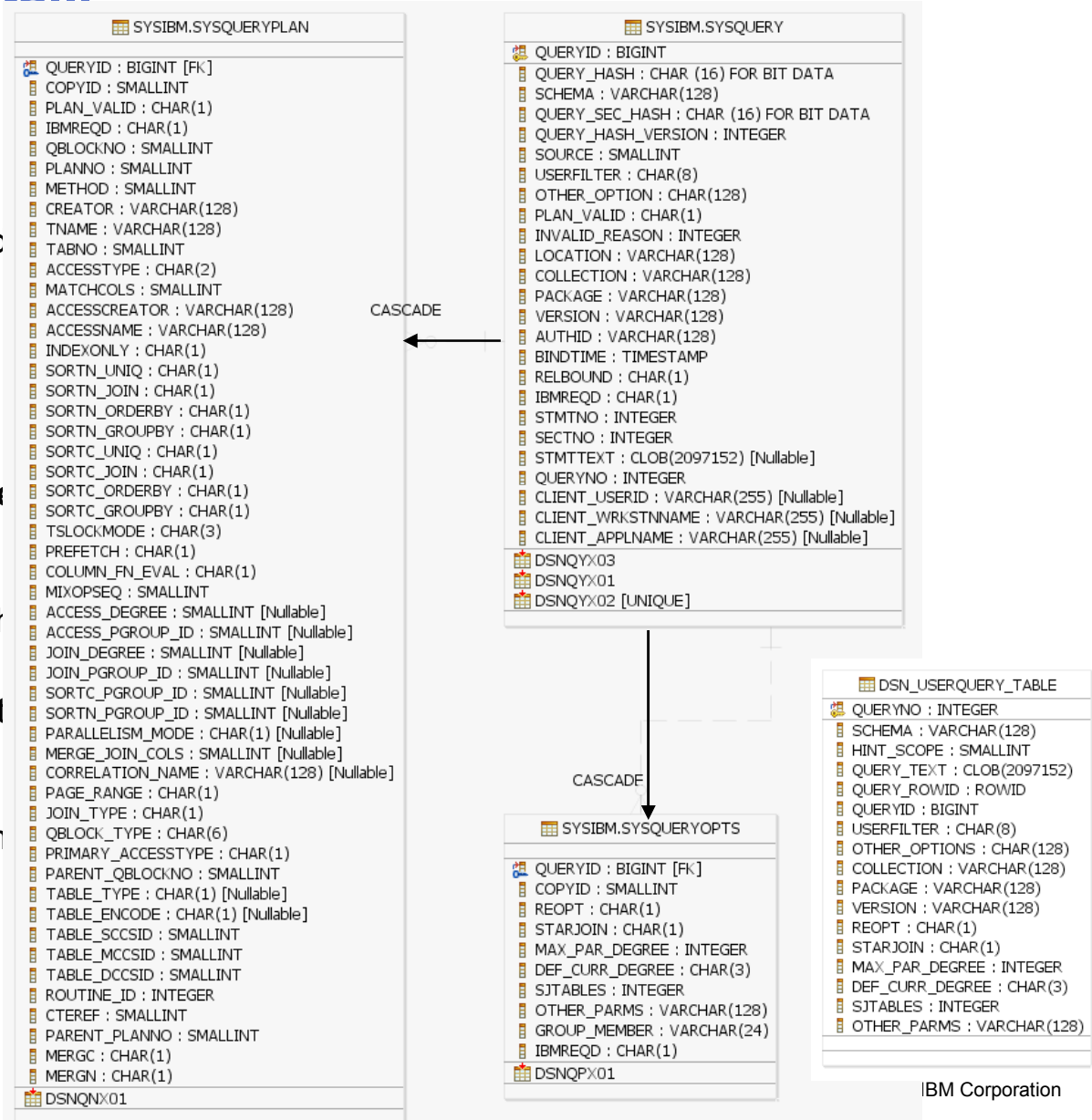
- **Statements that benefit by including literal values will run better:**
 - Without literal replacement
 - Or with REOPT (all REOPT options supported with literal replacement)
 - These include statements that benefit from distribution or correlation statistics
- **Better suited statements are likely smaller workload SQL statements with literal that could benefit from DSC**
- **EXPLAIN STATEMENT CACHE ALL will populate DSN_STATEMENT_CACHE**
 - The LITERAL_REPL column will indentify literal replacement SQL
- **Cannot have literal replacement and parameter markers**
 - For cases with parameter markers and literals
 - The literals are considered constants that do not change and the values are provided to the optimizer for a better filtering estimate

Access Path Stability Improvements (APSI)...

- **Static SQL provides many benefits, including:**
 - Performance
 - Security
 - Predictability / Stability
- **Optimization Hints**
 - Pre- DB2 10 offered User Level Optimization Hints
 - DB2 10 introduces Statement Level Optimization Hints
 - Access Path Repository
 - Statement level hints with **BIND QUERY** for static or dynamic SQL
 - Can enforce an existing access path from a PLAN_TABLE (hints) or
 - Can customize optimization options for a statement

APSI / Optimization Hints...

- **The Access Path Repository (APR) implemented during the ENFM mock migration**
 - SYSIBM.SYSQUERY
 - SYSIBM.SYSQUERYPLAN
 - SYSIBM.SYSQUERYOPTS
- **The new input table, DSN_USERQUERY_TABLE, is under schema of a userid**
- **SYSQUERY**
 - Contains data for all optimization hints
- **SYSQUERYPLAN**
 - Contains data for optimization hints and an access path
- **SYSQUERYOPTS**
 - Contains data for hints that customize optimization parameters



Access Path Comparison

- Users sometimes need to **BIND/REBIND** static SQL packages and yet maintain the exact same access paths.
- **BIND PACKAGE** and **REBIND PACKAGE** - two new options:
 - **APREUSE**
 - **YES/NO**: The default is **APREUSE(NO)**.
 - **APREUSE(YES)** - DB2 attempts to reuse existing access paths.
 - Enforcement is not guaranteed in all cases.
 - **APCOMPARE**
 - **WARN/ERROR/NONE**:
 - With **WARN** and **ERROR**, DB2 compares the old and new access paths.
 - If the access paths are structurally dissimilar, **DSNT285I** message produced.
 - **WARN** - DB2 will continue processing.
 - **ERROR** - DB2 will terminate the processing of the package.
 - **NONE** - DB2 does not perform access path comparison. Default is **NONE**.
 - **APREUSE/APCOMPARE** – supported on packages created **V9** or higher

SQL Pagination targets 2 classes of OR queries:

- **Cursor scrolling SQL**
 - Retrieve next n rows
 - Common in COBOL/CICS and any screen scrolling application
- **Complex OR predicates against the same index**
 - Common in SAP

- **In both cases:**
 - The OR (disjunct) predicate refers to a single table only.
 - Each OR predicate can be mapped to the same index.
 - Each disjunct has at least one matching predicate.

SQL Pagination

- Prior to DB2 10 OR predicates often result in multi-index access with list prefetch
- DB2 10 supports single index access without list prefetch
- Range List Index Scan requires
 - Every OR predicate
 - has at least one matching predicate
 - is mapped to the same index
 - Therefore this isn't just for positioning or paging.
- **ACCESSTYPE="NR" (iN list Range)**
 - At runtime, the ordering of the access path steps will be determined
 - PLAN_TABLE represented the order coded in the query
 - Ordering will be based on literal values and the order of the index chosen

```

SELECT *
FROM EMP
WHERE
  (
    LASTNAME = ?
    AND FIRSTNME = ?
    AND MIDINIT > ?
  )
OR
  (
    LASTNAME = ?
    AND FIRSTNME > ?
  )
OR
  LASTNAME > ?
ORDER BY LASTNAME
        ,FIRSTNME
        ,MIDINIT
;
    
```

EMPX1 Index (LASTNAME, FIRSTNME, MIDINIT)

MIXOPSEQ	METHOD	TABLE_NAME	TABNO	ACCESSTYPE	MATCHCOLS	INDEX_NAME
1	0	EMP	1	NR	3	EMPX1
2	0	EMP	1	NR	2	EMPX1
3	0	EMP	1	NR	1	EMPX1

Simple scrolling – Index matching and ORDER BY

- Scroll forward through the PHONEBOOK to obtain the next 20 rows from the current position – JONES, WENDY
 - Assumes index is available on (LASTNAME, FIRSTNAME)
- WHERE clause may appear as:

```
WHERE (LASTNAME='JONES' AND FIRSTNAME>'WENDY')  
      OR (LASTNAME>'JONES')  
ORDER BY LASTNAME, FIRSTNAME;
```

```
WHERE ((LASTNAME='JONES' AND FIRSTNAME>'WENDY')  
      OR (LASTNAME>'JONES')  
      AND (LASTNAME >= 'JONES'))  
ORDER BY LASTNAME, FIRSTNAME;
```

Extra predicate required in v9 for matching index access

Multiple OR ranges against same index

Assume an index on (LASTNAME, FIRSTNAME).

Given the following complex WHERE clause:

```
WHERE (LASTNAME=' JONES ' AND FIRSTNAME=' WENDY ' )  
      OR (LASTNAME=' SMITH ' AND FIRSTNAME=' JOHN ' ) ;
```

SQL pagination will allow single matching index access for this type of OR condition – without list prefetch.

V9 requires multi-index access with list prefetch for matching index access.

Dynamic Prefetch: Row Level Sequential Detection (RLSD)...

- **Problem: Dynamic sequential prefetch could work poorly when the number of rows per page is large**
- **Solution: DB2 10 has row level sequential detection (RLSD)**
 - Count rows, not pages
 - Unclustered row is less likely to cause DB2 to fall out of prefetch
- **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
 - Progressive prefetch also applies to indexes (but still counts pages)

Parallelism...

- **Support parallelism for multi-row fetch**
 - Pre DB2 10
 - Parallelism is disabled for the last parallel group in the top level query block
 - if there is no more tables to join after the parallel group
 - and there is no GROUP BY clause or ORDER BY clause
 - Example:- `SELECT * FROM CUSTOMER`
 - There is no parallel group in the query and there are no table joins
 - There is no GROUP BY clause
 - There is no ORDER BY clause
 - So NO PARALLELISM will be used
 - Only effective if CURSOR is DECLARED as READ ONLY
- **Allow parallelism if a parallel group contains a work file**
 - View or table expression materialization results in a work file
 - This type of work file is not shared among child tasks prior to DB2 10
 - Hence parallelism is disabled
 - DB2 10 will make the work file shareable
 - only applies to CP mode parallelism and no full outer join case

Parallelism...

- **Previous releases of DB2 divide the number of keys or pages by the number representing the parallel degree**
 - One task is allocated per degree of parallelism
 - The range is processed and the task ends
 - Tasks may take different times to process due to uneven distribution/skew

- **DB2 10 may use the Straw Model workload distribution method**
 - More key or page ranges will be allocated than the number of parallel degrees
 - The same number of tasks as before are allocated (same as degree)
 - Once a task finishes its smaller range it will process another range
 - Skewed data has the opportunity to be divided into a smaller number of pieces

Parallelism...

- **Dynamic record range partitioning**
 - Intermediate results are divided into ranges
 - Equal number of records
 - Division doesn't have to be on the key boundary
 - Unless required for group by or distinct function
 - Record range partitioning is dynamic
 - No longer based on the key ranges decided at bind time
 - Now based on number of
 - Composite side records and
 - Workload elements
 - Not impacted by
 - Data skew,
 - Out of date statistics
 - Will attempt to use in-memory work file for the materialization

Indexing Enhancements

- **I/O Parallelism**
- **Include Columns**
- **RID List**
- **Sequential Inserts**
- **Index Cost Reduction**

I/O Parallelism for Index Inserts...

- **Transactions that perform inserts into tables with many indexes defined on the table previously may have had high response times due to index I/O wait time.**
 - DB2 executes index updates sequentially
- **I/O parallelism : overlap the I/Os against non-clustering indexes**
 - Utilized if there are more than 3 indexes defined on the table and one of them is a clustering index, or 2 indexes if neither is a clustering index
 - DB2 can prefetch pages from different indexes defined on the same table into buffer pool in parallel for insert operation.
- **For a table with 6 indexes and 2000 inserts, elapsed time improvements of up to 50% are measured**
- **This functionality is enabled for DB2 10 Conversion mode**

Additional Non-key Columns In An Index

- **Indexes are used to enforce uniqueness constraints on tables**
- **To achieve index only access on columns not part of the unique constraint, additional indexes are often created for the non-unique columns**
 - Slower DB2 transaction time
 - Increased storage requirements
- **In DB2 10 Additional Non-key Columns can be defined in a unique index to reduce total amount of needed indexes**
- **Indexes that participate in referential integrity (RI) will support additional columns, but INCLUDE(d) columns will not be used to enforce RI**
- **Improves:**
 - insert performance as less indexes need to be updated
 - space usage
 - Can stabilize access path as optimizer has fewer similar indexes to choose from

```
CREATE UNIQUE INDEX ON t1(c1,c2,c3) INCLUDE (c4,c5)
```

More Index Enhancements

- **RID list overflows to workfile instead of relational scan (MAXRBLK was 8MB, now 400MB)**
 - Eliminate RID list failures from all four causes
 - DB2 9 had it for pair-wise join
 - MAXTEMPS_RID new Zparm
- **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

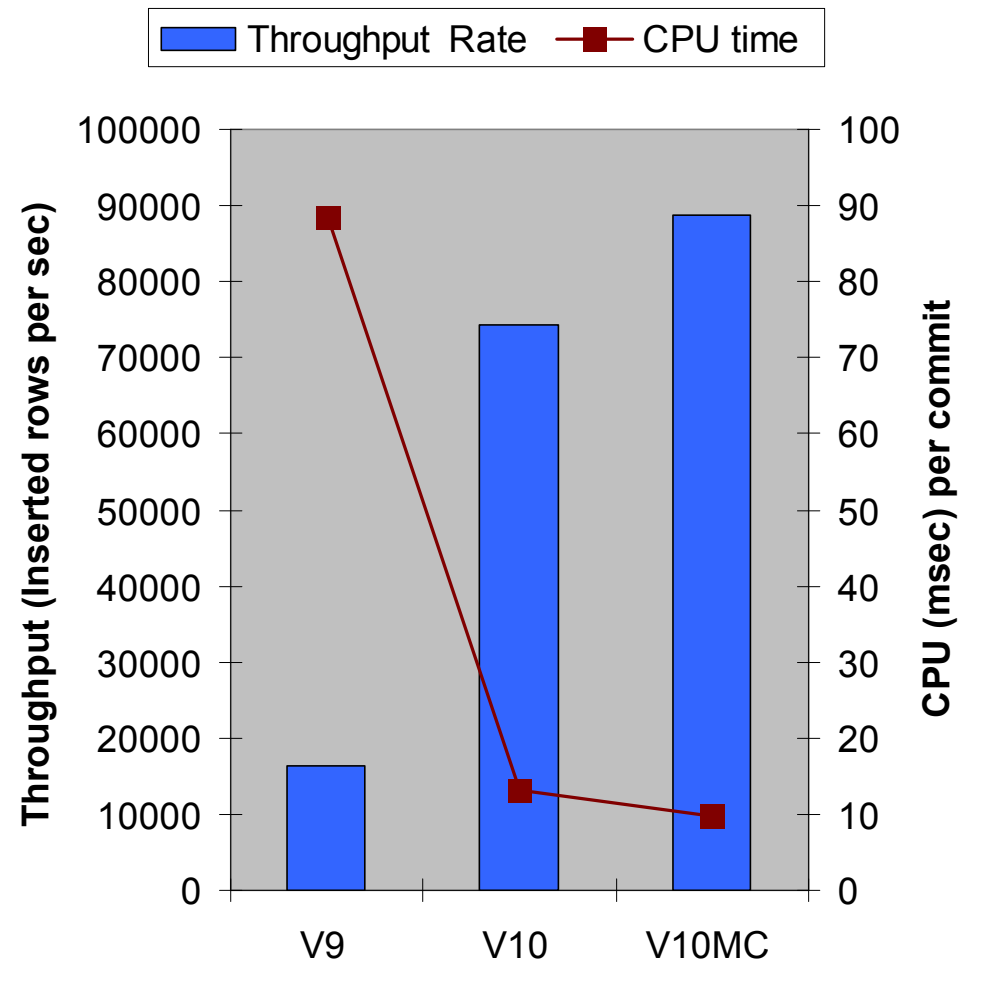
RID limit exceeded:

- RDS limit
- DM limit
- Exceed proc limit
- No storage

Sequential Inserts

- **Performance improvement for sequential inserts into the middle of a cluster index**
 - Optimize when index manager picks the candidate RID during sequential insert
 - Significant space search improvement in sequential insert
 - Reduced latch class 006, and 245
 - Higher chance to find open space and avoid a search
- Test case: Sequential key insert into 3 partitioned TS from 240 clients in 2way data sharing. Multi row insert.

PBR Sequential Insert



Member Cluster is only shown as a comparison point

MC = member cluster

Cost reduction in Index Evaluation

- **Optimization in index predicate evaluation process**
 - Applicable in any workload but query with many predicate shows higher improvement
 - Preliminary measurements shows average 18% CPU reduction (1% thru 70%) from TPC-H like workload using 100 queries
- **Reduced need for REORG**
 - Index scan using disorganized index
 - Index manager switches from dynamic prefetch to list prefetch for a disorganized index
 - Index scan can be 2 to 6 times faster, CPU time is less too
 - REORG, CHECK, and RUNSTATS performance improvement
 - Sequential INSERT and DELETE improvement
 - Maximum elapsed time benefit occurs if:
 - Cache hit ratio is high or index is on SSD
 - Index is striped

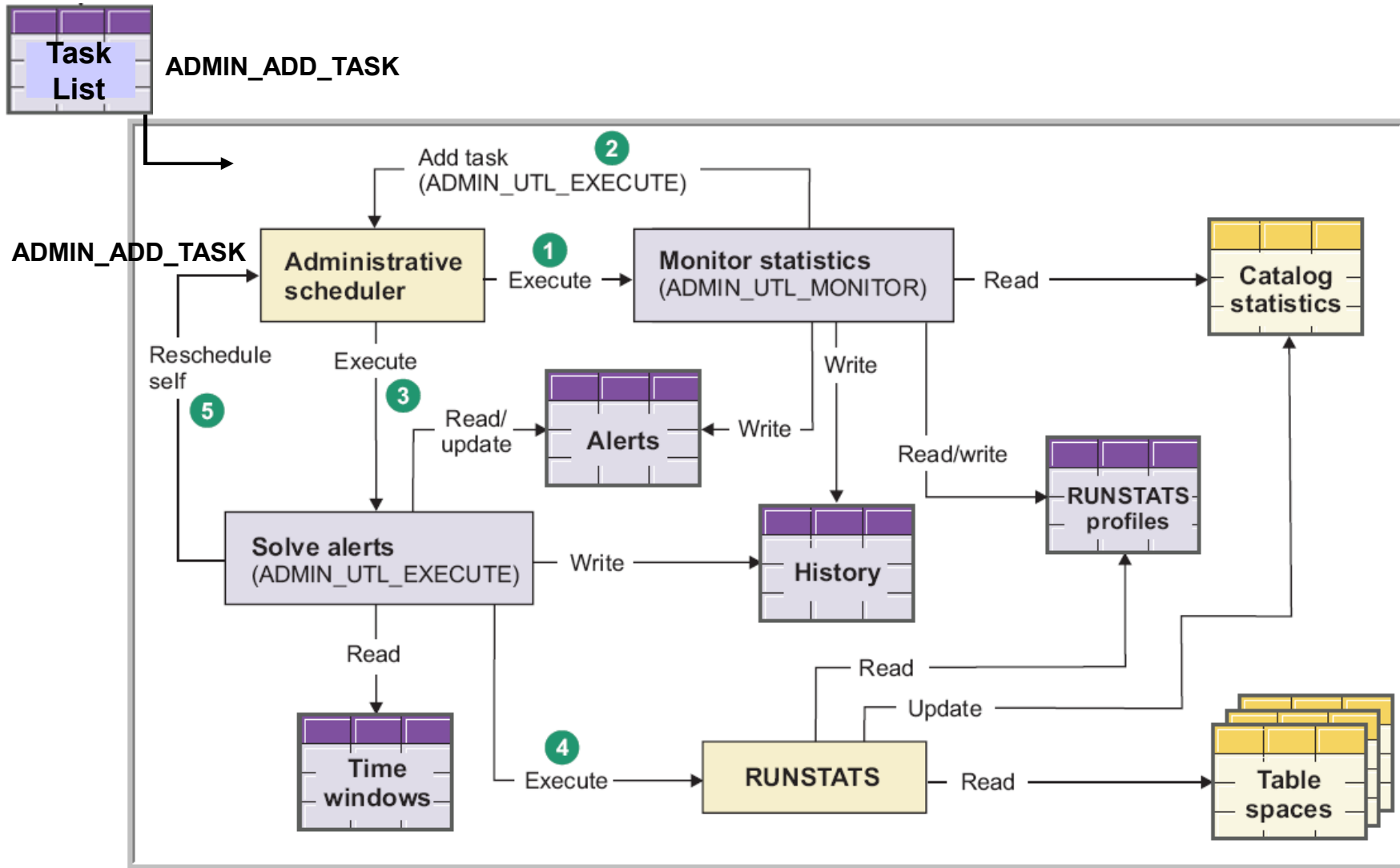
Statistics Enhancement... AutoStats

- **Autostats is implemented through a set of Stored Procedures**
 - Stored procedures are provided to enable administration tools and packaged applications to automate statistics collection.
 - **ADMIN_UTL_MONITOR:** Look for statistics needing update
 - **ADMIN_UTL_EXECUTE:** Execute RUNSTATS
 - **ADMIN_UTL_MODIFY:** Clean up the Autostats Repository
 - **ADMIN_ADD_TASK:** Called by MONITOR/EXECUTE & start up
 - STATS ADVISOR (Data Studio)
- **Run automatically honoring specified time windows**
- **Working together, these procedures**
 - Determine what stats to collect
 - Determine when stats need to be collected
 - Schedules and Performs the stats collection
 - Records activity for later review

See Chapter 11 "Designing DB2 statistics for performance" in the DB2 10 for z/OS Managing Performance Guide for details on how to configure autonomic monitoring directly within DB2.

Also see the ADMIN_UTL_MONITOR definition for criteria of Missing, Out of Date, or Inconsistent statistics.

Statistics Enhancement... AutoStats



EXPLAIN Changes...

- **EXPLAIN tables must be in:**
 - V8 or later format (Otherwise -20008, RC 2 is returned)
 - Unicode encoding scheme
 - Is preferred for V8 & V9 formats
 - Required for V10 format (Otherwise SQLCODE -878 returned)
- **EXPLAIN CAPTURE Special Register (defaults to NO)**
 - Set to YES
 - Explainable SQL statements run normally and explain information is captured after each statement is prepared and executed
 - Set to EXPLAIN
 - Explainable SQL statements do not execute but explain information is captured during prepare
 - Note that application logic may fail when expecting successful execution
 - Requires a PLAN_TABLE and DSN_STATEMENT_CACHE_TABLE
- **The DSNAEXP stored procedure is deprecated**
 - This procedure allowed the EXPLAIN of a SQL statement with the authorization for the statement
 - The EXPLAIN privilege or SQLADM should be used

Separation of Duties ...

■ **New ZParm – SEPARATE_SECURITY**

– Specified on DSNTIPB

- YES – Users with SYSADM can not perform GRANTS on objects created by others
- NO – Users with SYSADM can administer security for all objects
- Available in CM Mode
- Users with INSTALL SYSADM can still perform GRANTS for other users

■ **SYSADM/INSTALL SYSADM data access remains unchanged**

– Future direction is to only use SECADM for security and INSTALL SYSADM for install activities

– Users with SYSADM or INSTALL SYSADM can still view all data within tables

– Recommendation:

- Limit the use of INSTALL SYSADM and SYSADM to only when needed

Separation of Duties ...

- New SECADM authority to manage security of data
 - Does not have access to the data
 - Is able to manage GRANTS on all objects
- New Administrative authorities
 - DATAACCESS – to control who can see data
 - ACCESSCTRL – to control who can govern access to the data
- New DBADM ON SYSTEM authority
 - Gives the individual or role the ability to
 - Manage all user tables in the subsystem
 - With or without DATAACCESS – default is with DATAACCESS
 - With or without ACCESSCTRL - default is with ACCESSCTRL
- Administrative authorities can be divided among individuals without overlapping responsibilities
- Allows security administrator to grant the minimum privilege to a user to perform a specific task

SQLADM

- **Designed to be used by Performance analyst**
- **This will allow performance analyst to do all performance work, except access data**
- **What can a person with SQLADM do?**
 - Issue SQL EXPLAIN statement
 - Issue START, STOP and DISPLAY PROFILE commands
 - Perform actions involving
 - EXPLAIN privilege
 - STATS privilege on all user databases
 - MONITOR2 privilege
 - Execute DB2 supplied stored procedures and routines
 - Ability to SELECT, INSERT, UPDATE, DELETE on DB2 catalog tables
 - CAN NOT access data, perform DDL or EXECUTE plans or packages

Row and Column level access ...

- **What is the purpose of row level security?**
 - Filter rows out of answer set
 - Policy can use session information like SQL ID is in what group or user is using what role to control when row is returned in result set
 - Applicable to SELECT, INSERT, UPDATE, DELETE & MERGE
 - Defined as a row permission:

```
CREATE PERMISSION policy-name ON table-name  
FOR ROWS WHERE search-condition  
ENFORCED FOR ALL ACCESS ENABLE
```

- Optimizer inserts search condition in all SQL statements accessing table. If row satisfies search-condition, row is returned in the answer set

Row and Column level access ...

■ What is the purpose of column level security?

- Mask column values in answer set
- Applicable to the output of outermost subselect
- Defined as column masks:

```
CREATE MASK mask-name ON table-name FOR COLUMN  
column-name RETURN CASE expression ENABLE;
```

Optimizer inserts case statement in all SQL accessing table to determine mask value to return in answer set

Row and Column level access ...

■ Define a column or row policy based on who is accessing the table

– SESSION-USER

- Primary authorization ID of the process

– CURRENT SQLID

- SQL authorization ID of the process
- SET CURRENT SQLID = some authorization id

– VERIFY_GROUP_FOR_USER (new BIF)

- Get authorization IDs for the value in SESSION_USER
 - Gets both primary and secondary auth ids
 - Return 1 if any of those auth IDs are in the argument

```
WHERE  
  VERIFY_GROUP_FOR_USER(SESSION_USER,'MGR','PAYROLL') = 1
```

– VERIFY_ROLE_FOR_USER (new BIF)

- Get the role for the value in SESSION_USER
- Return 1 if the role is in the argument list

```
WHERE  
  VERIFY_ROLE_FOR_USER(SESSION_USER,'MGR','PAYROLL') = 1
```



Row and Column level access – Banking example...

- Create a column mask on INCOME for customer service rep

```
CREATE MASK INCOME_COLUMN_MASK ON CUSTOMER
FOR COLUMN INCOME RETURN
  CASE WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER, 'CSR') = 1)
    THEN CASE WHEN (INCOME > 150000) THEN 3
              WHEN (INCOME > 75000) THEN 2
              WHEN (INCOME > 25000) THEN 1
            ELSE 0
          END
        ELSE 0
      END
ENABLE;
```

Row and Column level access – Banking example...

- **Activate Row-level and column-level access control**

```
ALTER TABLE CUSTOMER
  ACTIVATE ROW      ACCESS CONTROL
  ACTIVATE COLUMN  ACCESS CONTROL;
```

- **What Happens in DB2?**

- A default row permission is created implicitly to prevent all access to table customer (WHERE 1=0)
- All packages and cached statements that reference table CUSTOMER are invalidated

Row and Column level access – Banking example...

```
SELECT ACCOUNT, NAME, INCOME, PHONE, BRANCH  
FROM CUSTOMER  
WHERE BRANCH IN ('A', 'B');
```

ACCOUNT	NAME	INCOME	PHONE	BRANCH
2222-3333-4444-5555	Bob	1	222-2222	B
3333-4444-5555-6666	Louis	2	333-3333	B

INCOME is automatically masked by DB2

If the user is not a member of the CSR group, then no rows at all will be returned

Business value of DDF Enhancements

- **Lessen CPU consumption of remote threads**
 - High performance DBAT can reuse threads very efficiently
 - Continued improvement of applications using JCC driver to attach locally
 - Recent addition of zAAP eligibility for type 2 connections

High Performance DBAT...in 2 steps

- High Performance DBATs reduce CPU consumption by
 - Supporting RELEASE(DEALLOCATE) to avoid repeated package allocation/deallocation
 - xPROCs, CTs and PTs, lookaside and prefetch will not have to be re-initialized
 - Bigger CPU reduction for short transactions
- High Performance DBAT behavior ... **STEP 1 package must be RELEASE(DEALLOCATE)**
 - DBAT will stay active with connection when DBAT is about to go be pooled and there is at least one RELEASE(DEALLOCATE) package still existing
 - Normal idle thread time-out detection will be applied to these DBATs.
If DBAT is in completed unit-of-work status, Connection will turn inactive instead of being canceled (POOLINAC value controls this)
 - Connections will turn inactive after 200 times (not changeable) to free up DBAT (same as in previous releases)
- **IBM plans to change default bind option for client packages from RELEASE(COMMIT) to RELEASE(DEALLOCATE)**

High Performance DBAT...

- New -MODIFY DDF PKGREL(BNDOPT/COMMIT) command ... **STEP 2**
 - to alter DDF's inactive connection processing which is activated via the ZPARM, CMTSTAT=INACTIVE
 - Display command shows DSNL106I message with PKGREL= BNDOPT or COMMIT
 - 2 options
 - PKGREL(BNDOPT) honors package bind option
 - PKGREL(COMMIT) forces package bind option RELEASE(COMMIT)
 - Same as v6- DB2 9 inactive connection behavior
 - Will allow BIND and DDL to run concurrently with distributed work

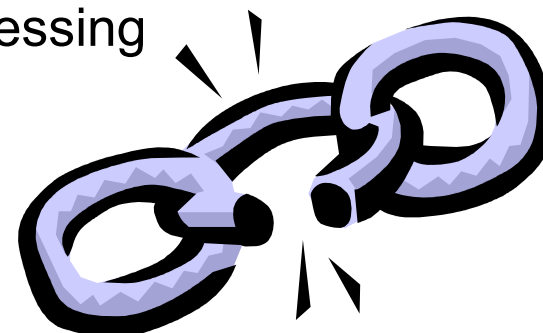
Handling of Private Protocol Requests in V10

- **Jobs to analyze and prepare for private protocol elimination**
 - PK64045 – delivered DSNTDP2DP update, as well as DSNTPPCK
- **Ability to Enable or Disable Private protocol (PK92339)**
 - Via ZPARM PRIVATE_PROTOCOL in DSN6FAC Macro
 - Enables testing. PRIVATE_PROTOCOL=NO ZParm will mimic DB2 10 behavior in V8/V9
- **DB2 10 will respond to a Private protocol response as follows, from a v9 or prior system**
 - Reject request
 - VTAM sense code '10086021'
 - Requestor will receive
 - -904 Reason Code '00D31026' Type '1001'
 - DB2 10 DDF will reject a BIND with Private protocol
 - SQLCODE -30104
- **Attempt to load/execute object with DBPROTOCOL column = 'P' will fail with SQLCODE -904, reason code 00E3001E, saying it needs to convert to DRDA, except the following case**
 - If a package with PRIVATE PROTOCOL accesses local data only it is allowed. If it attempts to access remote data it will fail

Utility Enhancements

Utility Enhancements

- **Unless explicitly stated, new utility functions are only available in NFM**
- **REORG SHRLEVEL(CHANGE) for complete catalog/directory**
 - Improved access to SYSUTILX by removal of the UTSERIAL lock
 - Hashes and links are removed during ENFM processing
 - Row level locks are relied upon
 - No changes to utility jobs are necessary
 - Greatly improves access to catalog/directory
- **Improves concurrency for online REORG**
 - Update of catalog statistic information for inline REORGs occurs after access to the table is allowed



REORG...

- **REORG TABLESPACE or INDEX**
 - SHRLEVEL CHANGE/REFERENCE
 - Materializes pending changes
 - Table space type, DSSIZE, SEGSIZE, data page size, index page size, member cluster
 - REORG SHRLEVEL(NONE)
 - Will not materialize pending DDL changes, but will proceed

REORG TABLESPACE...

- **Performance improvement for partition-level REORG with NPI's and REORG INDEX**
 - Index list prefetch results in up to 60% ET reduction
- **Messaging to estimate log apply phase time to completion and SWITCH phase begin is estimated to miss the DEADLINE**
 - SHRLEVEL REF or CHANGE only
 - DSNU383I in addition to DSNU105I
- **REORG of multiple non-contiguous partitions allowed**
 - Changes behavior from DB2 9
 - Customers who perform multi-part REORGs can now specify multiple parts in a single command
 - Ex: REORG TABLESPACE PARTS (1,5,10:15, 20:23)
 - Retro-fitted into v9 via PM22336

REORG TABLESPACE...

- **REORG can CANCEL threads on last RETRY processing**
 - Ensures REORG can break in on applications
 - Allows REORG to cancel threads that hold claims on objects that prevent drains
 - FORCE NONE|READERS|ALL
- **New Advisory REORG Pending status – AREOR**
 - Indicates that there are pending alters that require a table space level REORG
- **Statistics**
 - Now defaults to STATISTICS TABLE ALL INDEX ALL UPDATE ALL HISTORY ALL
 - In DB2 9 the default is the value specified in the STATISTICS HISTORY field in the DSNTIPO panel which by default is NONE

REORG INDEX/REBUILD INDEX...

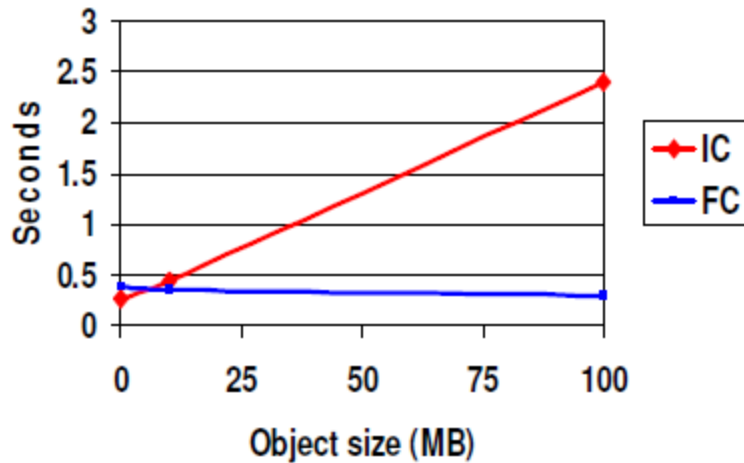
- **REORG INDEX SHRLEVEL(REFERENCE or CHANGE)**
 - Will materialize pending index changes so long as there are no pending table space changes
- **REORG INDEX SHRLEVEL(NONE)**
 - Will not materialize pending index changes
 - DSNU1165I RC=4 will be issued indicating pending changes were not done
 - For indexes defined COPY YES, an entry will be made in SYSCOPY
- **For REBUILD INDEX**
 - If there are pending TS changes, REBUILD proceeds and message DSNU275I RC=4 is issued indicating pending changes were not materialized
 - For indexes defined COPY YES, an entry will be made in SYSCOPY

FLASHCOPY SUPPORT...

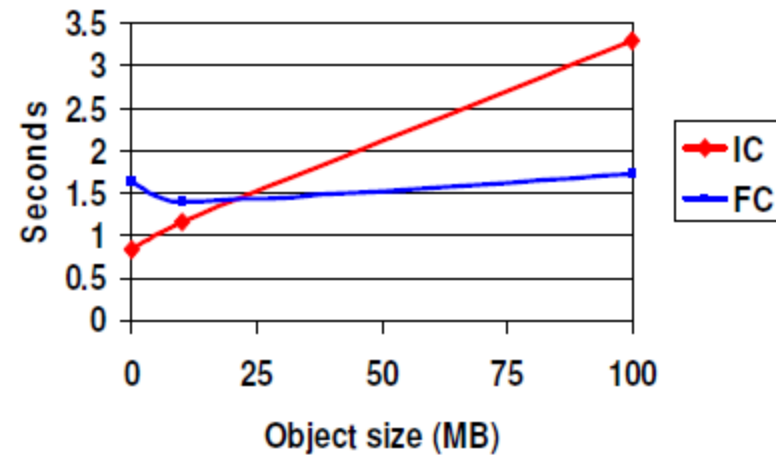
■ Dataset-level Flashcopy support

- For COPY, RECOVER, REORG, LOAD, REBUILD INDEX, REORG INDEX, CHECK
- Significant CPU and elapsed time reductions vs image copy

CPU time per object (z10)



Elapsed time per object (z10)



COPY

- **Create transaction consistent image copies from a COPY SHRLEVEL CHANGE run using Flashcopy**
 - DB2 10 uses the Flashcopy copy to make a transaction consistent image copy.
 - Don't have to wait for blockers. No application outage and no quiesce.
 - Uncommitted units of work are rolled back against the Flashcopy
 - Image copy is then taken from the Flashcopy
 - Objects must be SMS managed residing on FC v2 volumes
- **Improved dataset management & performance**
 - Will not allocate incremental copy dataset unless copy taken
 - &ICTYPE now matches actual image copy
 - Regardless of the type specified on a template i.e. TYPE=C
 - Utilize RTS for CHANGELIMIT performance

RECOVER

▪ Point-in-time recovery with BACKOUT

- Prior to DB2 10 recovered from a backup image then applied logs to reach the desired point in time.
- DB2 now provides an option on utilities that specifies if the recovery should be performed by processing the log backwards or in the traditional restore and roll forward through the log process.
- Changes are backed out from the current state. IC not used.
- Customer decides if BACKOUT is desirable option to use
 - IBM Recovery Expert can assist in making the best Recovery decision
- In order to work successfully, indexes must be marked COPY YES since DB2 needs the SYSCOPY records for non-recoverable events to know whether the RECOVER can proceed or not.
 - PM30991 HIPER should be on to disallow a BACKOUT recovery when certain unlogged events exist.

```
RECOVER TABLESPACE myschema.mytablespace BACKOUT YES  
TOLOGPOINT x'00000518DFEF'
```

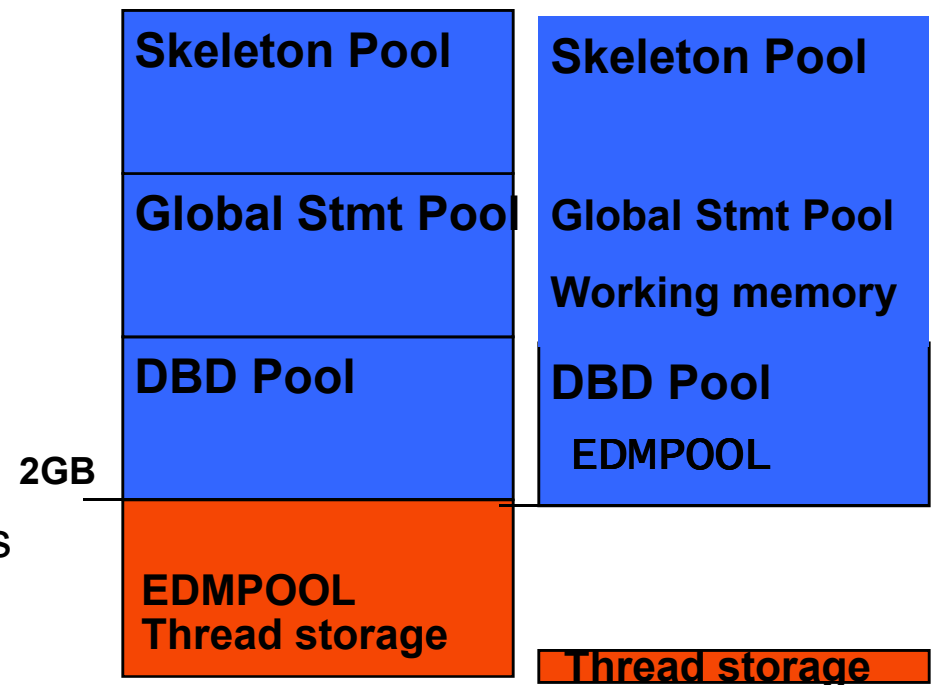

Virtual Storage Improvements

- **More Storage Above the Bar**
 - Ability to run more threads
- **In-memory Buffer pool objects**
- **Workfile Enhancements**
- **Additional support for 64 bit runtime**
- **1 MB Page size**
- **System Latching changes**
- **On Demand Buffer Pool storage**
- **Log Buffer Page Fix**

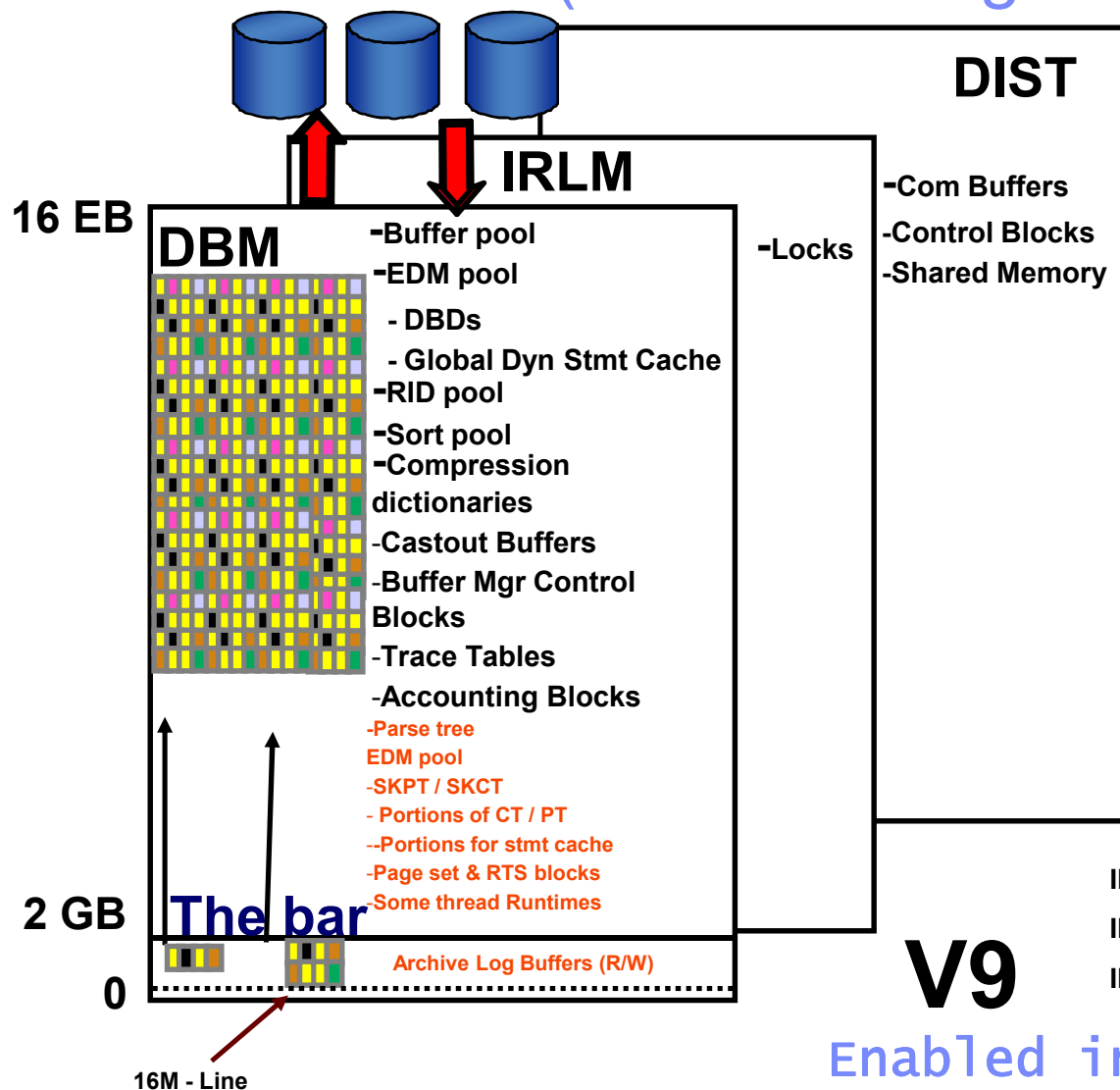
DB2 10: 64 bit Evolution (Virtual Storage Relief)

Scalability: Virtual storage constraint is still an important issue for many DB2 customers, until DB2 10

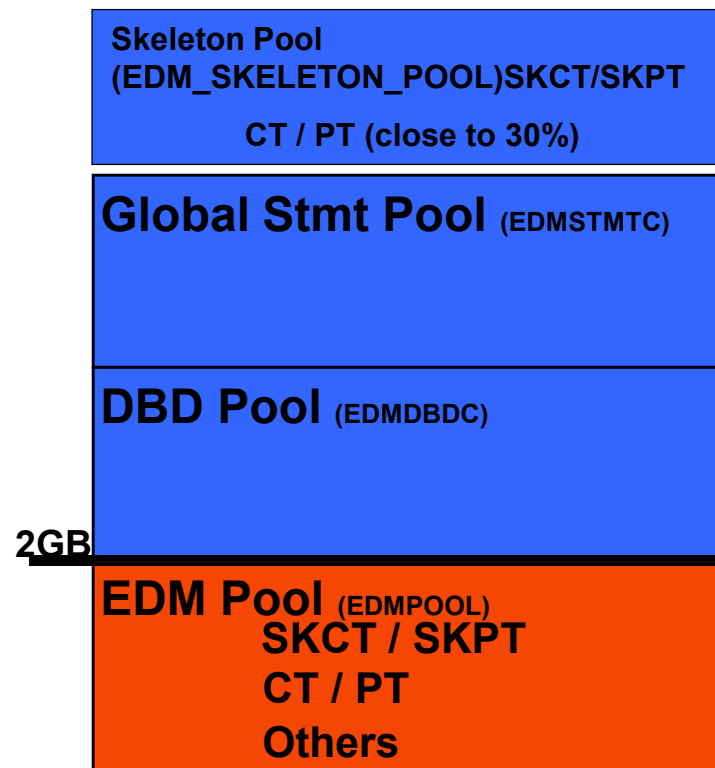
- **DB2 10 supports 5-10x more active threads, up to 20,000 per member**
 - 80-90% of thread storage moved above the bar
 - More concurrent work
 - Reduce need to monitor
 - Consolidate members and LPARs
 - Reduced cost, easier to manage, easier to grow
 - REBIND required to get most of the savings



64 bit Evolution (Virtual Storage Relief)

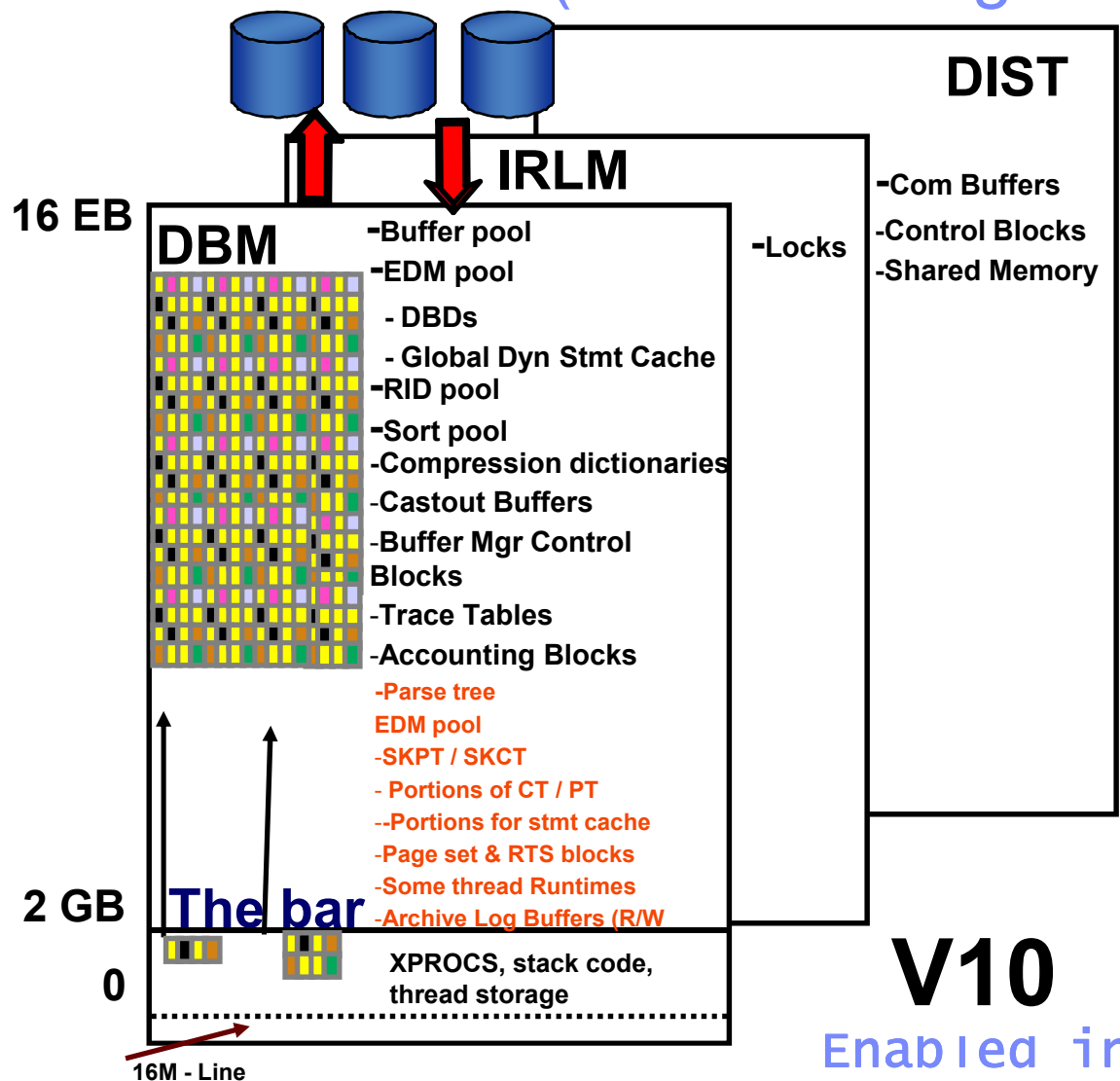


Environmental Descriptor Manager Pool

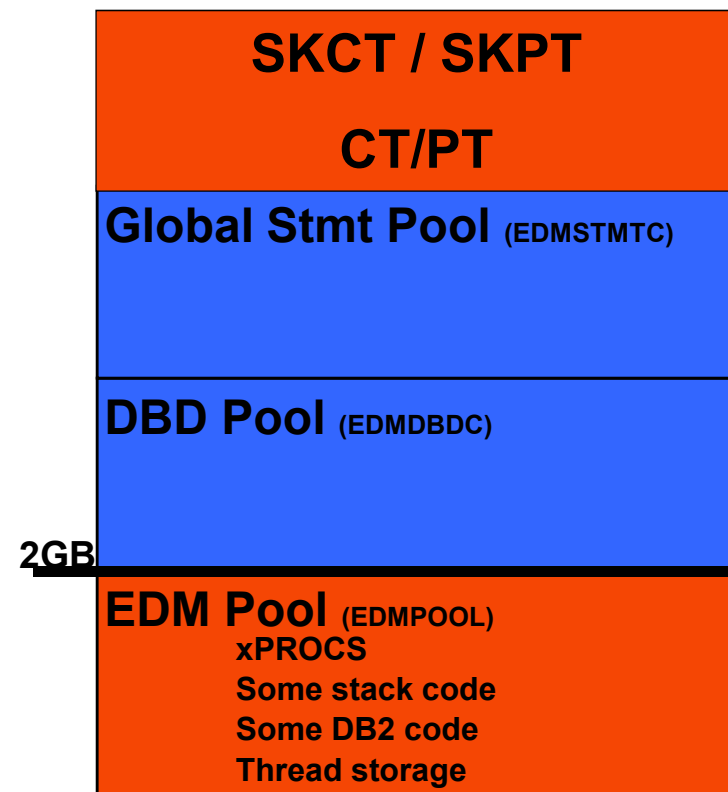


IFCID 217: detailed DBM1 virtual storage health
 IFCID 225: consolidated DBM1 virtual storage health
 IFCID 002: Mapping macro for EDM Pool, DSNDQISE

64 bit Evolution (Virtual Storage Relief)



Environmental Descriptor Manager Pool



V10

Enabled in CM

Bufferpool Enhancements...

- **DB2 may cache an entire tablespace into a bufferpool.**
 - Data will be preloaded into bufferpool when object is opened and will remain until closed.
 - > Specified by PGSTEAL=NONE
 - Excellent for lookup tables and indexes
 - > Small tables that have high I/O rates
 - Avoid LRU chain maintenance and LC14 contention
 - Avoid unnecessary prefetch and LC24 contention
 - IFCID 201 & 202 will be updated to signify this condition
- **On Demand Bufferpool**
 - This gives you the ability to grow your buffer pool over time as opposed to allocate it all at first usage
 - Maximum size will be VPSIZE

1 MB Page Size (Frame size)

- DB2 takes advantage of the new 1 MB page size on the z10 and z196
 - Specify PGFIX=YES to get 1 MB page size in your buffer pools
 - Must be backed by real storage
 - Must allocate space above the bar with LFAREA parm in IEASYSxx in Parmlib
 - Potential for significant performance improvements
- Using the 1 MB page size will enable efficiencies in the hardware

Data Sharing

- **Coexistence**
- **Member Consolidation**
- **LC19 reduction**
- **Buffer Pool Scan Avoidance**
- **MEMBER CLUSTER**
- **Local Subgroup Attach**
- **ACCESS DB**
- **Index Split Improvements**
- **Restart Improvements**
- **Log Buffer Page Fix**
- **ALTER TABLESPACE / INDEXSPACE enhancement**
- **Optimization when removing GBP Dependency**
- **New ZParms**

Data Sharing Sysplex Workload Balance Migration

- **PM24292 addresses the DB2 server reporting mixed levels of Product Identification and DRDA Support**
- **Possible migration issue**
 - Data Sharing groups that remain up during
 - Conversion to NFM
 - Reversion from NFM
 - Running Sysplex Workload Balancing
 - Different members of a Data Sharing group could report different DRDA support levels during migration to / from NFM
 - Remote applications could encounter error codes -4212, -4499
- **To allow for a smooth, seamless migration**
 - Review II14619
 - Review PM24292 & apply UK65312 to DB2 10 (the target version)
 - IBM Data Server Client support in V9.7 FP 3A

Data Sharing Improvements...

- **Coexistence possible with V8 (Mode CM8) or V9 (Mode CM9)**
 - Private Protocol only support with V8 / V9 members during CM8 / CM9
- **Possible Member Consolidation**
 - Reduction in virtual storage below the bar
 - CTHREAD increases
 - Reduced resource contention
 - UTSERIAL lock removed
 - LC19 reduction with LRSN “spin” enhancement
- **LC19 reduction**
 - LRSNs do not have to be unique for INSERTs
 - Internal measurements have shown a 70 – 130% improvement in throughput for workloads constrained by LC19
- **MEMBER CLUSTER for UTS**
- **Subgroup Attach for Local Connections**
 - Gives customers the flexibility to separate workloads.

Data Sharing Improvements...

- **Restart Lite for DDF Indoubt URs**
- **ACCESS DB command wildcarding**
- **Index split recording**
 - IFCID 359 reports index splits when performance class 4 is active
 - Also reports if index was group buffer pool dependent when the split occurred
- **Group restart avoidance of lock resolution problems**
 - Occasional lock resolution problems have been resolved by a lock structure rebuild
 - DB2 10 IRLM detects waits and rebuilds the structure automatically
- **Page fixing the log buffers (not specific to data sharing)**
- **Table and Index Spaces no longer need to be stopped to change the buffer pool**
- **Optimization when removing Group Buffer Pool Dependency**
 - IRLM lock timeouts could occur during CF “delete name” requests when removing inter-DB2 Read Write interest
 - Process is changed to only delete CF data entries

Data Sharing Improvements...

- **DSN_STATEMENT_CACHE table has a GROUP_MEMBER column**
 - EXPLAIN STMTCACHE output can distinguish between cache of different members
- **ZParm DEL_CFSTRUCTS_ON_RESTART**
 - Old CF structures have caused DR test difficulties
 - Defaults to NO
 - If YES
 - Attempts to delete CF structures on group restart
 - Recommended to be the same across members
 - Could be set to YES in the DR ZParm
 - Only the first member to access the structures will trigger the delete
 - Added via PM28925
 - IRLM PM31807 needed to delete Lock Structure

Logging..

- **Ability to add new active log data sets without having to recycle DB2.**
- **NEWLOG and COPY keywords added to the –SET LOG command.**
SET LOG NEWLOG(*dsn*) COPY(*log-copy*)
 - *dsn* = name of newly defined active log data set
 - *log-copy* = “1” for copy-1 or “2” for copy-2
 - Changes are pervasive.
- **DB2 10 enhances checkpoints thresholds to be scheduled on either or both.**
 - Provides ability to use log records when subsystem is busy, and time interval when subsystem is relatively inactive.

Enhanced Monitoring Support

- **DB2 10 enhanced monitoring support is provided as two independent functions.**
 - Statement level
 - System level
- **At the statement level this supports performance monitoring and problem determination for both static and dynamic SQL.**
 - Uses IFI to capture and externalize information for consumption by tooling.
 - Introduces a unique statement execution identifier (STMT_ID).
- **At the system level this provide increased granularity of monitoring system level activities.**

Statistics Management

■ SMF Compression

- Via the new **SMFCOMP** ZParm
- Accounting records may compress 80-90%
- CPU Overhead ~1%
- When **ACCUMACC** ZParm is also set (default to 10) can compress up to 99%
 - DB2 10 rollup accounting provides more detail
- PM27872
 - Adds job DSNTSMFD to decompress SMF records

■ Statistics interval changes

- STATIME default reduced to 1 minute
 - IFCIDs 0105, 0106, 0199, and 0365
- IFCIDs 0001, 0002, 0202, 0217, 0225, and 0230 are no longer controlled by STATIME
 - Corresponding trace records are written at fixed, one-minute intervals

Questions

