

DB2 for z/OS V10 Overview

Kevin Harrison

Sr. Systems Engineer/Sr. IT architect

IBM Silicon Valley Lab

kharrison@us.ibm.com



Disclaimer and Trademarks

Information contained in this material has not been submitted to any formal IBM review and is distributed on "as is" basis without any warranty either expressed or implied. Measurements data have been obtained in laboratory environment. Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. The use of this information is a customer responsibility.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, DATABASE 2, DB2, Enterprise Storage Server, FICON, FlashCopy, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, System z, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, zSeries

The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95

For additional information see ibm.com/legal/copytrade.phtml

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.

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
 - Can be altered to higher precision.
 - Will fail with `SQLCODE -190` if lowered.
- Example: `TS(6)` to `TS(10)`
Example: `TS(10)` to `TS(6)`



Functions Compatibility

- **Functions Modified for Competitive Compatibility**
 - **See PM40724**
 - **Enabling APAR PM38326**
 - **Functions added / modified**
 - **Modified**
 - LTRIM
 - RTRIM
 - REPLACE
 - ROUND
 - TRUNCATE or TRUNC
 - **Added**
 - TRIM
 - **PM70455 available for DB2 9**
 - **Begins using IFCID 366 to report functions & timestamp usage**
- 5 **considered incompatible in DB2 10**

Stored Procedure Enhancements

▪ **PM37668**

- Stored Procedure RETURN TO CLIENT cursor support
 - Returns result set to client from a nested stored procedure
 - Result set is invisible to intermediate stored procedures
- For NSPs RETURN TO CLIENT cursors
 - Duplicate cursor instance support (see APAR for details)
 - Becomes a result set cursor

▪ **PM53243**

- Stored procedure monitoring improvements
- Begin / End for each SP or UDF invocation (IFCID 233)
- Execution details (IFCID 380 & 381)
- Statement level details (IFCID 497, 498, 499)

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](#)

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
- **Invoked via UPDATE...**
- **Requires MV format**
- **One updater at a time for a document**
 - Concurrency control by the base table
 - Document level lock to prevent UR reader

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

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
 - PM31314 (9/2011) allows the use of TIMESTAMP WITH TIMEZONE
 - 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

- **History Table**

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

Benefits of Inline LOBs

- **Reduces auxiliary table storage, CPU, and virtual storage potential**
- **Inline portion of LOB compressible even though LOBs are not compressed**
- **Inline portion of LOB can be used in index on expression**
 - For example, when using spatial data the index portion of the spatial object can reside in the row
- **Inline LOBs stored in the base table can achieve similar performance characteristics as VARCHAR columns**

Access Path Stability Improvements...

▪ APCOMPARE / APREUSE

- Options to get stabilize access paths across BIND or REBIND
- The replaced package must be from DB2 9 or later
- APCOMPARE
 - Structural comparison of the access path across the BIND / REBIND
 - When differences are discovered (for existing statements), the outcome depends on the option used
 - WARN: Package is created with an RC=04
 - ERROR: No package is created with an RC=08
- APREUSE
 - Hints all statements for the new package using the access path from the package to be replaced
 - When differences are discovered (for existing statements), no package is created
 - ERROR is the only option
- **The previous (pre-BIND / REBIND) package must have been bound / rebound at the DB2 9 level or later**
- Starting in DB2 9 an Explain Data Block (EDB) is kept in the package in SPT01

Access Path Stability Improvements...

▪ APCOMPARE / APREUSE...

- BIND / REBIND & REBIND TRIGGER PACKAGE options
 - NO/NONE, WARN (APCOMPARE only), ERROR
- Uses the pre-BIND / REBIND current package for comparison or reuse to determine the next package
- The previous (pre-BIND / REBIND) package must have been bound / rebound at the DB2 9 level or later
- Starting in DB2 9 an Explain Data Block (EDB) is kept in the package in SPT01
 - Internal hint
 - A compressed, internal representation of the PLAN_TABLE
- PM25679 adds this new function
- PM30425 corrective maintenance to enhance package reusability
 - Packages must have been bound / rebound on this maintenance before expecting this reuse
 - Ex. Information about Virtual Tables and corrected COLUMN_FN_EVAL

Access Path Stability Improvements

▪ APCOMPARE / APREUSE

- Success or failure is at the package level
- Comparisons are done at the statement level
- Feedback is placed in the PLAN_TABLE if **EXPLAIN YES** or **ONLY** specified
 - HINT_USED contains 'APREUSE' when hint succeeds
 - REMARKS used for reuse or comparison failure information
 - These are placed on the 'new' PLAN_TABLE rows
 - If an 'old' PLAN_TABLE row is unmatched in the 'new' set of rows
 - A new row is created to show the REMARKS
- These are not “sticky”; must specify for each BIND/REBIND
- AUTOBIND defaults to **NO**

Access Path Stability Improvements

- **APCOMPARE (NO/NONE, WARN, ERROR)**
 - Determine if the static SQL statements have an access path change
 - This is an EDB structural comparison
 - There could be some details of the access path that are different
 - Examples:
 - MERGE_JOIN_COLS=3; Specific columns & order???
 - Only explainable statements can be compared
 - Compares statements that are exactly the same
 - Statement order in the package does not matter
 - If differences are found:
 - WARN results in an RC=04 and continues
 - Recommend the use of a Plan Management Policy with REBIND because a new package (and access path) will be created
 - ERROR results in an RC=08 and that specific BIND/REBIND is terminated
 - Use **EXPLAIN YES** or **ONLY** to get failure reasons

Access Path Stability Improvements

▪ APCOMPARE

– DSNT285I lists

- # of statements successfully compared
- # of statements with unsuccessful comparison
- # of statements that could not be compared
 - Examples include:
 - > Bound prior to V9
 - > VALIDATE(RUN)
 - > New statement during a BIND

– The 'old' package for comparison is found based on

- REPLACE or COPY..COPYVER
 - LOCATION, COLLID, NAME, VERSION
- ADD
 - LOCATION, COLLID, NAME
 - DSNT294I reports the VERSION found for comparison
- If Plan Management is in use, it always uses the 'current' copy

Access Path Stability Improvements

▪ **APREUSE (NO/NONE, ERROR)**

- Applies a hint from the EDB for the new package
- Attempts to avoid an access path change
 - Not guaranteed due to:
 - Missing objects
 - Version incompatibilities
 - Hint ambiguities (ex. Columns of a merge join)
- Can get the same errors as when using OPTHINTs
 - +395 / Reason Code
 - Reason Code 50 added for comparison failure
- ERROR & unable to reuse an explainable statement results in RC=08
 - That particular BIND / REBIND fails

Access Path Stability Improvements

▪ APREUSE

– Matches:

- LOCATION, COLLID, NAME, VERSION
- If no match, then LOCATION, COLLID, NAME
- If no match, DSNT292I and operation continues

– For packages that bind / rebind successfully

- SYSPACKSTMT ACCESSPATH='A'
- SYSPACKAGE & SYSPACKCOPY APREUSE column set
- Hints are not enforceable 100% of the time
- DB2 10 may merge query blocks or decorrelate subqueries

▪ Migration and APCOMPARE / APREUSE

- Consider using REBIND...EXPLAIN(YES)...APREUSE(ERROR)
- Then analyze the REBIND failures or use APCOMPARE(WARN)

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 is 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...

- **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
 - DSN_PGROUP_TABLE, RANGEKIND = 'R'

Universal Table Spaces

▪ What kind of Table Space will be created? (* optional)

CREATE TABLESPACE...	SEGSIZE	NUMPARTS	MAXPARTITIONS	Comments
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 *DPSEGSZ 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 – 4GB)
 - PM43175 adds DSSIZE 128 and 256
 - Proportionally decreases the number of partitions
 - 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)
 - With PM57001 MAXPARTITIONS can be altered lower if removed part(s) are not allocated
 - BUFFERPOOL PGSTEAL NONE
 - LOB INLINE LENGTH LOB

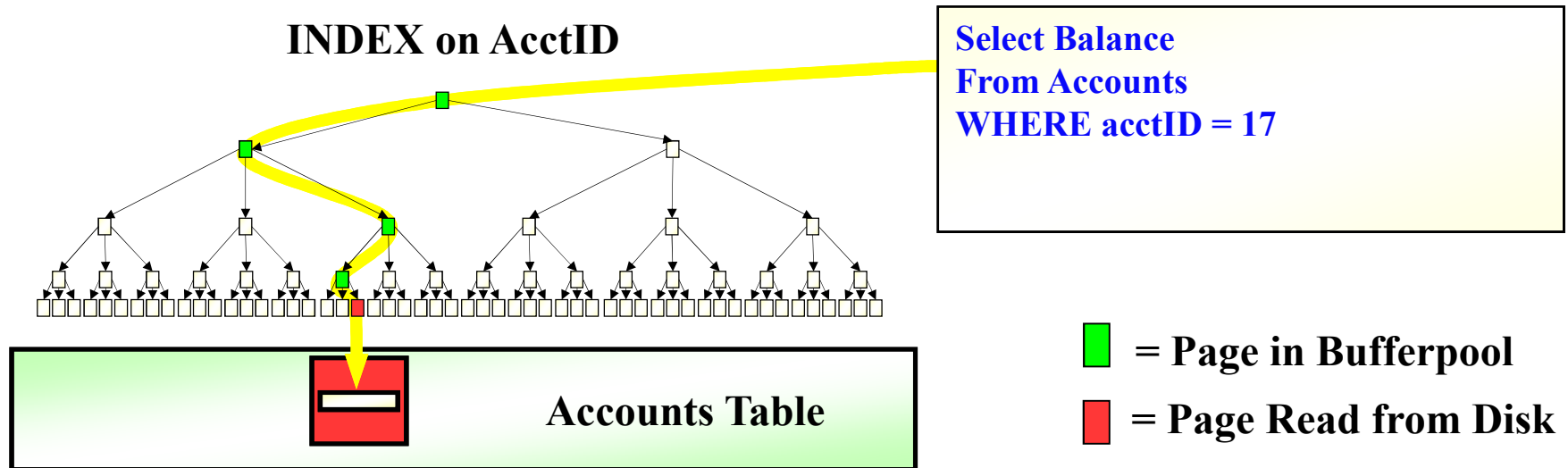
Compress on INSERT

- **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**

Hash Access

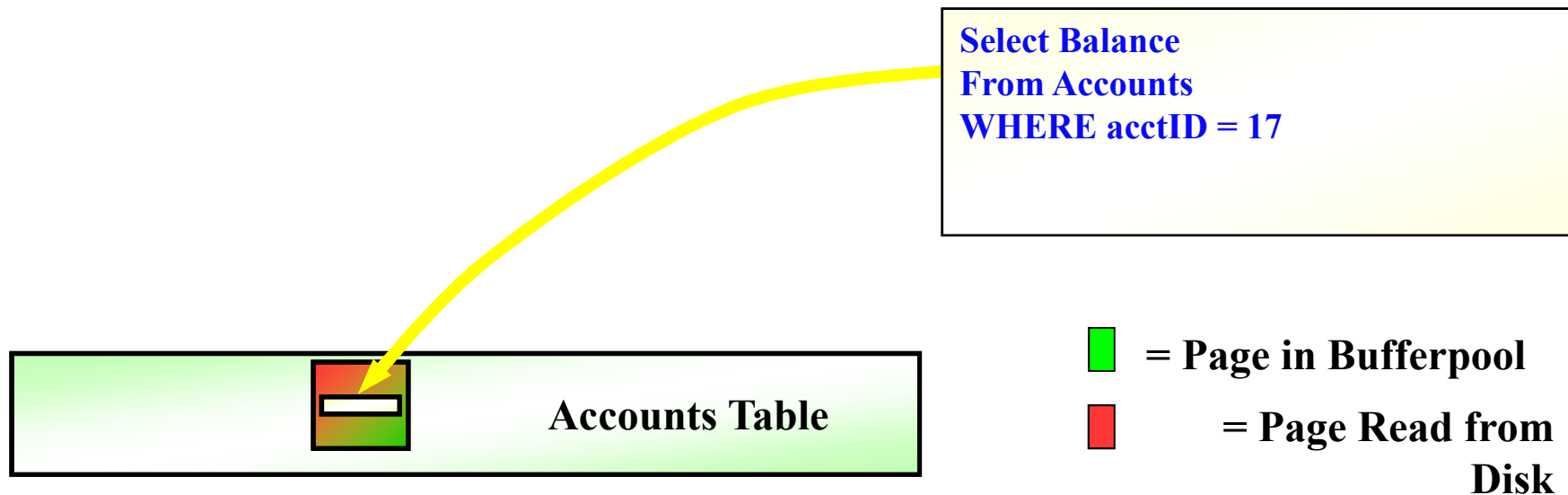
- **Hash access vs. Index only access and index lookaside**
 - Competes against index only access and index lookaside
 - Advantage that index only access still provides for clustered data access
 - Can now have unique index with INCLUDE columns
 - Reduce number of indexes required for performance reasons
 - Improve insert, update and delete performance
 - Need to find the sweet spot
 - High NLEVELS in index (≥ 3)
 - Purely direct row access by primary key
 - Truly random access
 - Read intensive, not volatile
 - No range queries
 - Many rows per page etc
 - Space allocation of fixed hash space is key to control overflow
 - Too small will lead to rows in overflow
 - Too large will lead to random IO
 - REORG AUTOESTSPACE(YES) but still some rows in overflow
 - Degraded LOAD and REORG utility performance

Index to Data Access Path



- **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

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
 - Data size changes frequently
 - 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
 - PM25652 adds REORG recommendations to DSNACCOX

I/O Parallelism for Index Inserts...

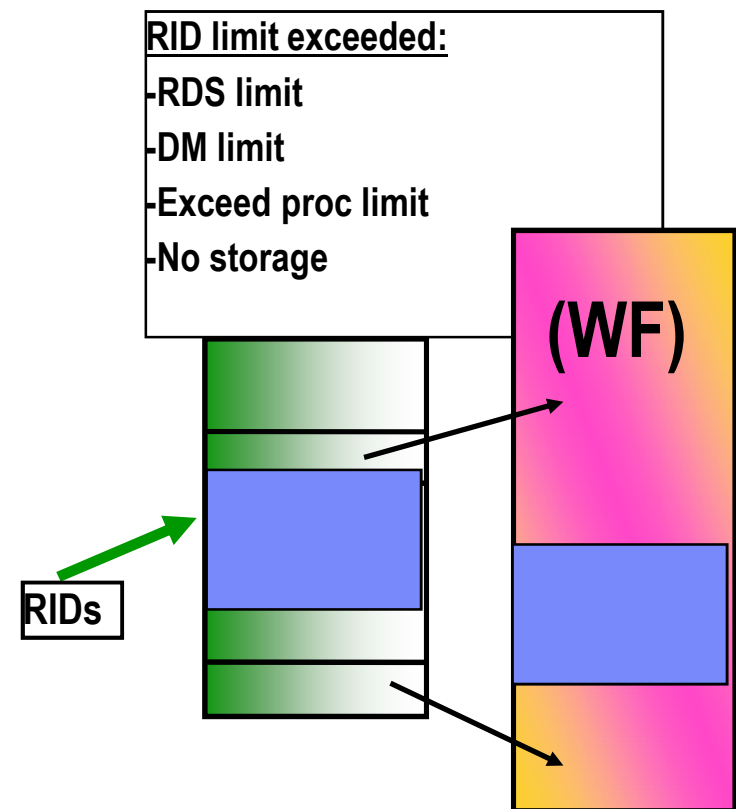
- **Transactions that perform inserts into tables with many indexes defined on the table previously may have had high response times due to index I/O wait time.**
 - DB2 executes index updates sequentially
- **I/O parallelism : overlap the I/Os against non-clustering indexes**
 - Utilized if there are 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.
- **New ZParm INDEX_IO_PARALLELISM with default YES**
- **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

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



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

Reduce risk by minimizing use of SYSADM

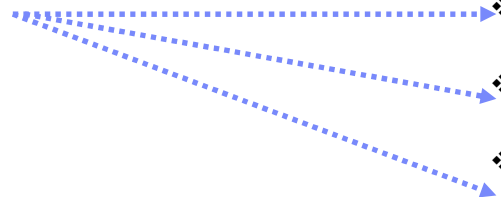
New granular system authorities

Prior to DB2 10

- ❖ SYSADM
- ❖ DBADM
- ❖ DBCTRL
- ❖ DBMAINT
- ❖ SYSCTRL
- ❖ PACKADM
- ❖ SYSOPR

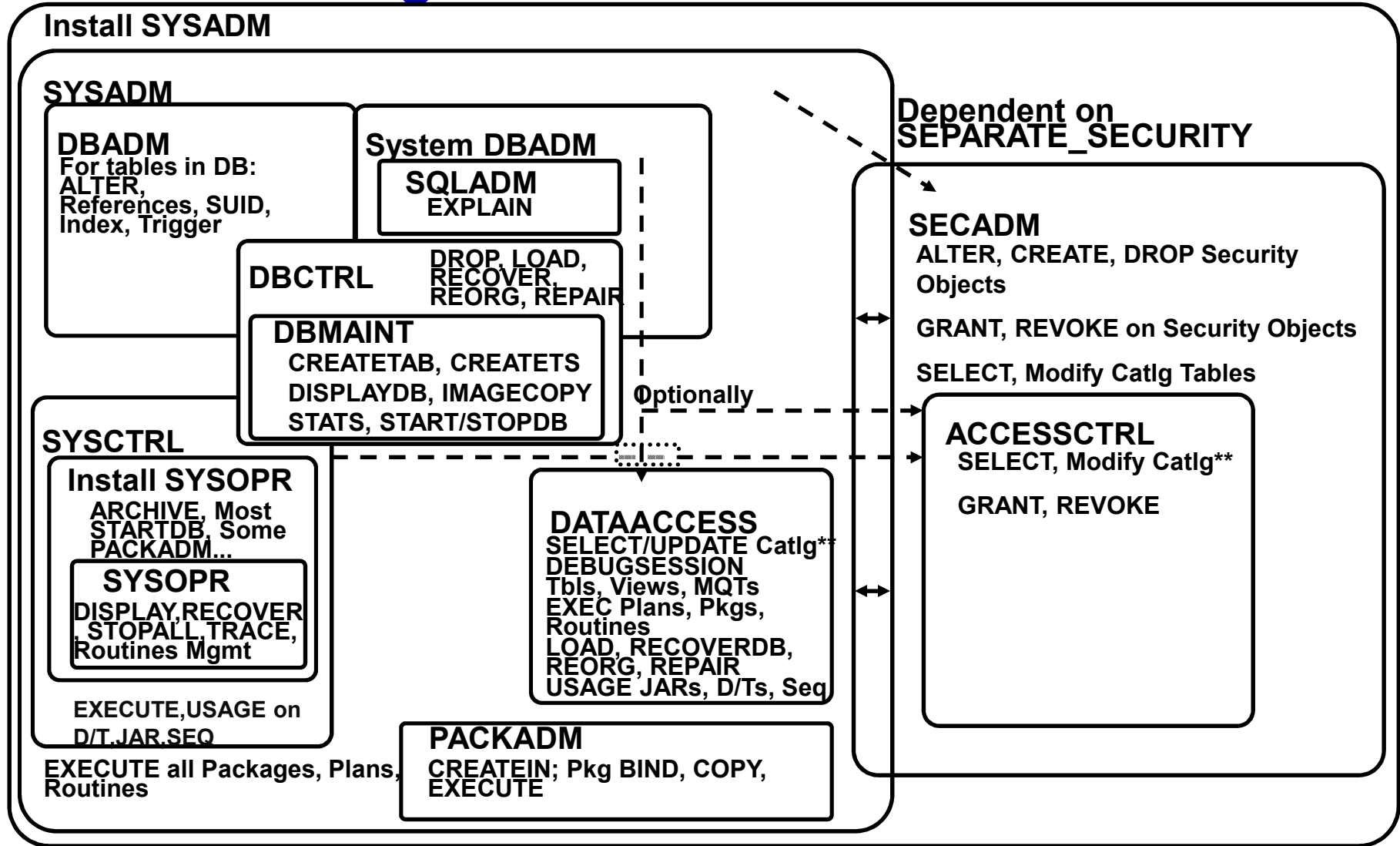
New in DB2 10

- ❖ System DBADM
- ❖ ACCESSCTRL
- ❖ DATAACCESS
- ❖ SECADM
- ❖ SQLADM
- ❖ EXPLAIN



Authorities Diagram

** Modify Catlg w/o SYSAUDITPOLICIES



Audit policy categories

➤ Audit policy supports eight categories.

<u>Categories</u>	<u>Mapping IFCIDs</u>
❖ CHECKING	❖ IFCID 83 (only authentication failures), IFCID 140
❖ VALIDATE	❖ IFCIDs 55, 83, 87, 169, 269, 319
❖ OBJMAINT	❖ IFCID 142
❖ EXECUTE	❖ IFCIDs 143, 144, 145
❖ CONTEXT	❖ IFCIDs 23, 24, 25
❖ SECMAINT	❖ IFCIDs 141, 270, 271
❖ SYSADMIN	❖ IFCID 361 (Audits installation SYSADM, installation SYSOPR, SYSOPR, SYSCTRL, SYSADM)
❖ DBADMIN	❖ IFCID 361 (Audits DBMAINT, DBCTRL, DBADM, PACKADM, SQLADM, system DBADM, DATAACCESS, ACCESSCTRL, SECADM)

Example: Dynamic auditing of tables

- **Audit all the tables that start with 'PAY' in EMPLOYEE schema**
 - Does not require AUDIT clause to be specified during table definition
 - IFCID 145 trace record contains full SQL statement text and unique statement ID
 - IFCID 143 and 144 trace records contain the unique statement ID that can be used to identify the SQL statement in IFCID 145 record.

```
INSERT INTO SYSIBM.SYSAUDITPOLICIES (AUDITPOLICYNAME, OBJECTSCHEMA,  
OBJECTNAME, OBJECTTYPE, EXECUTE)  
VALUES ('TABADT1','EMPLOYEE','PAY%', 'T','A');  
-STA TRACE (AUDIT) DEST (GTF) AUDTPLCY(TABADT1);
```

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

Performance and Scalability

- **High Performance DBATs (Hi-Perf DBATs) – new type of distributed thread**
 - Must be using CMTSTAT=INACTIVE so that threads can be pooled and reused
 - Packages must be bound with RELEASE(DEALLOCATE) to get reuse for same connection and -MODIFY DDF PKGREL(BNDOPT) must also be in effect
 - When a DBAT can be pooled after end of client's UOW
 - DBAT and client connection will remain active together
 - Still cut an accounting record and end the enclave
 - After the Hi-Perf DBAT has been reused 200 times
 - DBAT will be purged and client connection will then go inactive
 - All the interactions with the client will still be the same in that if the client is part of a sysplex workload balancing setup, it will still receive indications that the connection can be multiplexed amongst many client connections
 - IDTHTOIN will not apply if the Hi-Perf DBAT is waiting for the next client UOW
 - If Hi-Perf DBAT has not received new work for POOLINAC time
 - DBAT will be purged and the connection will go inactive
 - If # of Hi-Perf DBATs exceed 50% of MAXDBAT threshold
 - DBATs will be pooled at commit and package resources copied/allocated as RELEASE(COMMIT)
 - Hi-Perf DBATs can be purged to allow DDL, BIND, and utilities to break in
 - Via -MODIFY DDF PKGREL(COMMIT)

High Performance DBAT...

- 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
 - Must be using `CMTSTAT=INACTIVE` so that threads can be pooled and reused
 - Packages must be bound with `RELEASE(DEALLOCATE)` to get reuse for same connection and `-MODIFY DDF PKGREL(BNDOPT)` must also be in effect

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

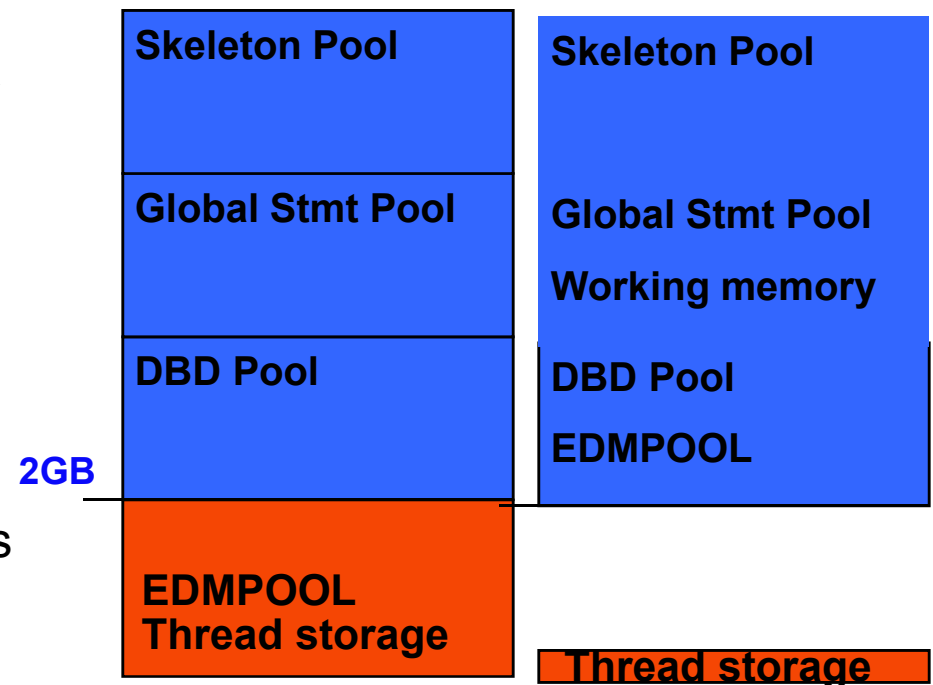
Plan Ownership Authorization Elimination

- **Plan Ownership Authorization is a Private Protocol security semantic that has been allowed with DRDA for previous releases of DB2 DB2 z/OS Client to DB2 z/OS Server**
 - Plan Owner must have EXECUTE authority for SQL at the DB2 Requester
- **With Private Protocol now eliminated in DB2 10, Plan Ownership Authorization is being removed from DRDA**
 - PM17665 removes this behavior in DB2 V8 and DB2 9 with PRIVATE_PROTOCOL=NO
 - PM17665 removes this behavior in DB2 10
 - The Primary Authorization ID must have EXECUTE authority of the package at the server
- **PM17665 introduces a potential release incompatibility in DB2 10**
- **PM37300**
 - Introduces PRIVATE_PROTOCOL=AUTH to DB2 V8, 9, and 10
 - This allows Private Protocol Authorization without allowing Private Protocol
 - For DB2 V8 and 9 this defaults to YES
 - For DB2 10 this defaults to NO
 - PRIVATE_PROTOCOL=NO supports secondary authorization IDs for SQL EXECUTE authority at the DB2 Server

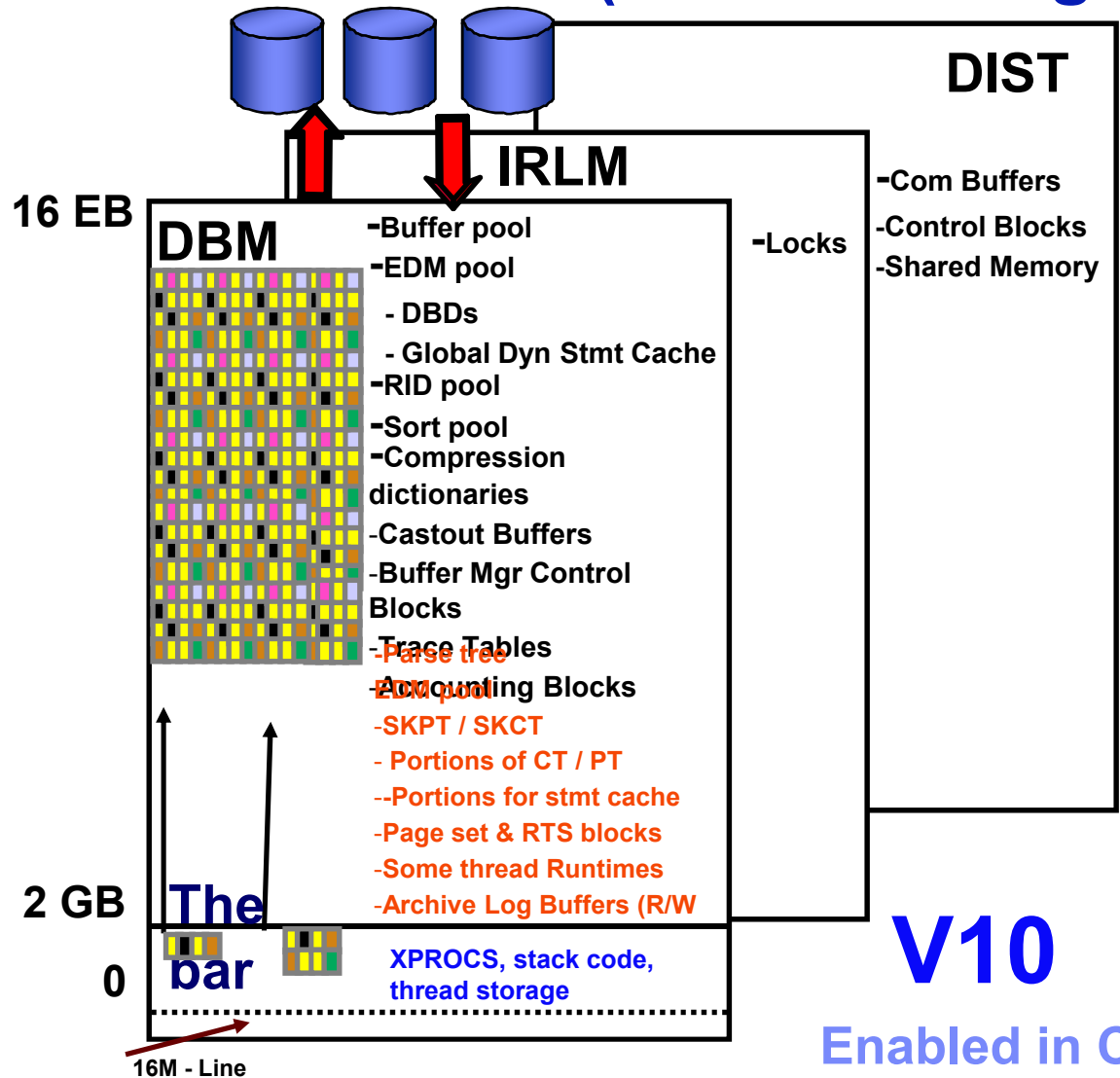
DB2 10: 64 bit Evolution (Virtual Storage Relief)

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

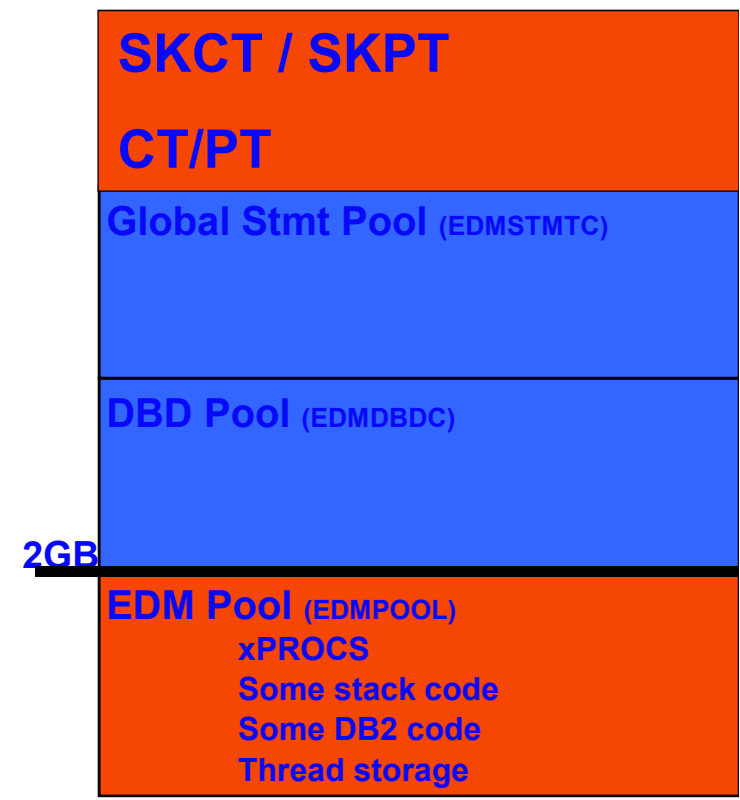


64 bit Evolution (Virtual Storage Relief)



V10
Enabled in CM

Environmental Descriptor
Manager Pool



In Memory Table Spaces...

- **DB2 can cache an entire tablespace into a buffer pool.**
 - Data will be preloaded into buffer pool 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

DGTT changes ...

- **V9 Consolidated temp data base and work file data base**
 - DGTT and other work file usage (sort work) share the work file database
- **APARS/PTFs introduced to alleviate contention issues**
 - Separation by workload
 - Secondary Quantity = 0 for Sort Work
 - Secondary Quantity >0 for DGTT
 - See <http://www.ibm.com/support/docview.wss?uid=isg1I14587> for details
- **V10 supports PBG table spaces in the work file**
 - DGTT can get beyond the 64GB limitation
 - MAXPARTITIONS allowed for table spaces in work file, as well as DSSIZE and Numparts (CREATE ONLY / no ALTER)
 - SYSTABLESPACESTATS will record work file data at the partition level for PBG tables spaces
 - New partitions will remain through DB2 restart

1 MB Frame Size

- DB2 takes advantage of the new 1 MB frame size on the z10 and later
- Must specify PGFIX=YES to get 1 MB frame size
- 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 frame size will enable efficiencies in the hardware
- See OA34024 for information about LFAREA sizing
- **DISPLAY VIRTSTOR,LFAREA**
 - To see if LFAREA has been allocated & High Water Mark information
 - Example display shows no 1MB frames available
- **PM85944: Pageable 1MB Frames**
 - Buffer Pool Control Block Only
 - Prereqs
 - EC12
 - Flash Express Memory
 - z/OS V1R13+ with prereq FMIDs & PTFs

```

Display Filter View Print Options Search
-----
SDSF SYSLOG 7962.103 MVSA MVSA 05/03/2012 13W
COMMAND INPUT ==> /DISPLAY VIRTSTOR,LFAREA
RESPONSE=DEMOMVS
IAR019I 06.52.50 DISPLAY VIRTSTOR 184
SOURCE = DEFAULT
TOTAL LFAREA = 0M
LFAREA AVAILABLE = 0M
LFAREA ALLOCATED (1M) = 0M
LFAREA ALLOCATED (4K) = 0M
MAX LFAREA ALLOCATED (1M) = 0M
MAX LFAREA ALLOCATED (4K) = 0M
  
```

1 MB Frame Size

- **When 1 MB Page Frames are in use and none available**
 - Reverts to 4K pages
 - No longer request 1 MB pages and pages in use remain mixed
- **DISPLAY BUFFERPOOL(BP1) SERVICE=4**
 - DSNB999I - how many 1MB size page frames are in use

```
DSNB999I  DSNT DSNB1DBP  SERVICE( 4 )OUTPUT
DSNB999I  DSNT 4K PAGES 246
DSNB999I  DSNT 1M PAGES 0
DSN9022I  DSNT DSNB1CMD  '-DISPLAY BUFFERPOOL' NORMAL COMPLETION
***
```

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
 - Ensure PM51093 is applied before NFM (Red Alert) for recovery and LPL/GRECP
 - Targeting Multirow INSERT operations
 - Internal measurements have shown a 70 – 130% improvement in throughput for workloads constrained by LC19
- **MEMBER CLUSTER for UTS**

Data Sharing Connection Queue Management

- Connections waiting for a DBAT could be redirected to another member if notified that the current member cannot service the request in a timely manner
 - **PM43293 introduced connection queue depth and wait time monitors (off by default)**
 - **MAXCONQN**: Max **number** of inactive or new connection requests waiting for a DBAT
 - **MAXCONQW**: Max **wait time** for a connection request to wait for a DBAT.
 - -DIS DDF DETAIL enhanced to show settings
 - **Only effective for:**
 - DB2 members in a data sharing group
 - Using CMTSTAT=INACTIVE
 - **DSNL030I with Reason Code 00D31053 (QN) or 00D31054 (QW)**
 - **Additionally**
 - Reduces DB2 system health as connections pass 80% & 90% of CONDBAT (with DSNL074I)
 - 50% & 25% of calculated health until condition relieved (with DSNL075I)
 - For KEEP DYNAMIC threads exceeding 1 hour or inactive for 20 minutes and inactive
 - Now produces messaging to indicate termination
- 52 • DSNL027I with 00D3003E or 00D3003F



Data Sharing Improvements...

- **Table and Index Spaces no longer need to be stopped to change the buffer pool**
 - Pre-DB2 10 / Data Sharing, the table space / index space needed to be stopped before the ALTER statement was issued
- **Optimization when removing Group Buffer Pool Dependency**
 - IRLM lock timeouts could occur during CF “delete name” requests when removing inter-DB2 Read Write interest
 - Typically associated with a DB2 member physically separated from the CF
 - Process is changed to only delete CF data entries
 - Avoiding potentially massive amounts XI signaling
 - CF “delete name” happens more quickly
 - Avoid impact to transactions
 - The directory entry will be deregistered when the local buffer pool pages are reclaimed via a “lazy” cleanup process

Active Log Data Sets

- **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*)
 - Changes are pervasive.
- **Data set must be defined with IDCAMS before issuing command.**
- **Recommendations:**
 - Format data set with DSNJLOGF utility before issuing command.
 - Add both copy 1 and 2 for dual logging.
- **New DB2 messages:**
 - DSNJ363I:
COPY*log-copy* LOG DATA SET *dsn* ADDED TO THE ACTIVE LOG INVENTORY
 - DSNJ364I: NEWLOG OPERATION FAILED: *reason*

Autonomic Checkpoint...

- **ZParm changes:**
 - New **CHKTYPE** defines threshold method being used.
 - **SINGLE** – either number of log records or time interval
 - **BOTH** – both methods used
 - New **CHKLOGR** defines log record threshold, when **CHKTYPE** = **BOTH**. Otherwise set to **NOTUSED**.
 - New **CHKMINS** defines time interval threshold, when **CHKTYPE** = **BOTH**. Otherwise set to **NOTUSED**.
 - Existing **CHKFREQ** defines log record or time interval threshold, when **CHKTYPE** = **SINGLE**. Otherwise set to **NOTUSED**.
 - **Note:** The Panel choices that drive these parameter selections are
 - **LOGRECS**
 - **MINUTES**
 - **BOTH**

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

Summary of DB2 10 Business Value

- **Integrated XML / Relational Support**
- **Extending the lead with Security enhancements**
- **Performance**
- **Manageability**
- **Extended Distributed Computing Performance**
- **Virtual Storage Management**
- **Data Warehousing**
- **Time Travel Queries with Temporal (Versioned) Data support.**

THANK
YOU

