DB2 11 for z/OS – Key Topics

June Lundstrom
DB2 Advisor for z/OS – West Region
jelundst@us.ibm.com

Acknowledgements and Disclaimers:

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

DB2 for z/OS Customer Trends

- **Proliferation of mobile and other network-connected devices is driving increases in:**
  - **transaction workloads**
  - **data volumes**
  - **24x7 requirements**
- **Continued focus on cost containment and resource efficiency**
- **Competitive pressures continue to drive an increasing need for innovation, analytics, and data integration**
- **DB2 for z/OS has leading edge capabilities to support these requirements and DB2 11 makes important improvements**

DB2 11 Major Themes

- **Out-of-the-box CPU Savings***
  - Improving efficiency, reducing costs, no application changes
  - Up to 10% for complex OLTP
  - Up to 10% for update intensive batch
  - Up to 40% for queries
  - Additional performance improvements through use of new DB2 11 features
- **Enhanced Resiliency and Continuous Availability**
  - Improved autonomics which reduces costs and improves availability
  - Making more online changes without affecting applications
  - Online REORG improvements, less disruption
  - DROP COLUMN, online change of partition limit keys
  - Extended log record addressing capacity (1 yottabyte)
  - BIND/REBIND, DDL break into persistent threads
- **Enhanced business analytics**
  - Expanded SQL, XML, and analytics capabilities
  - Temporal and SQLPL enhancements
  - Transparent archiving
- **Simpler, faster DB2 version upgrades**
  - No application changes required for DB2 upgrade
  - Access path stability improvements
  - Product quality/stability: support pre GA customer production

*REBIND may be required for best results

DB2 11 Performance Focus

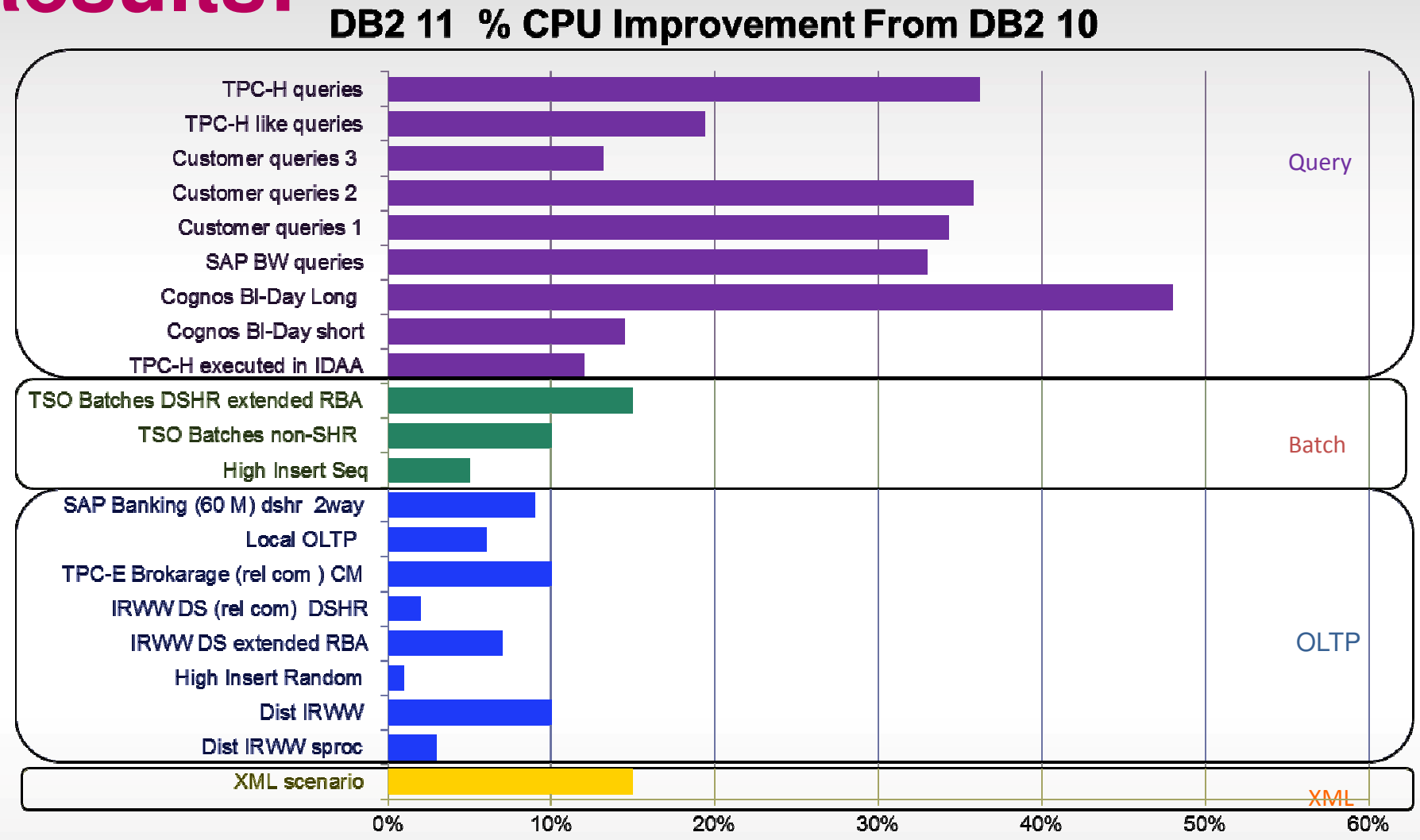
- CPU and cost reduction
- Scalability enhancements
- Focus on customers' pain points
  - Consistent performance with less REORG
  - Less need of performance tuning
  - Access Path Stability

"Reminder notice" –

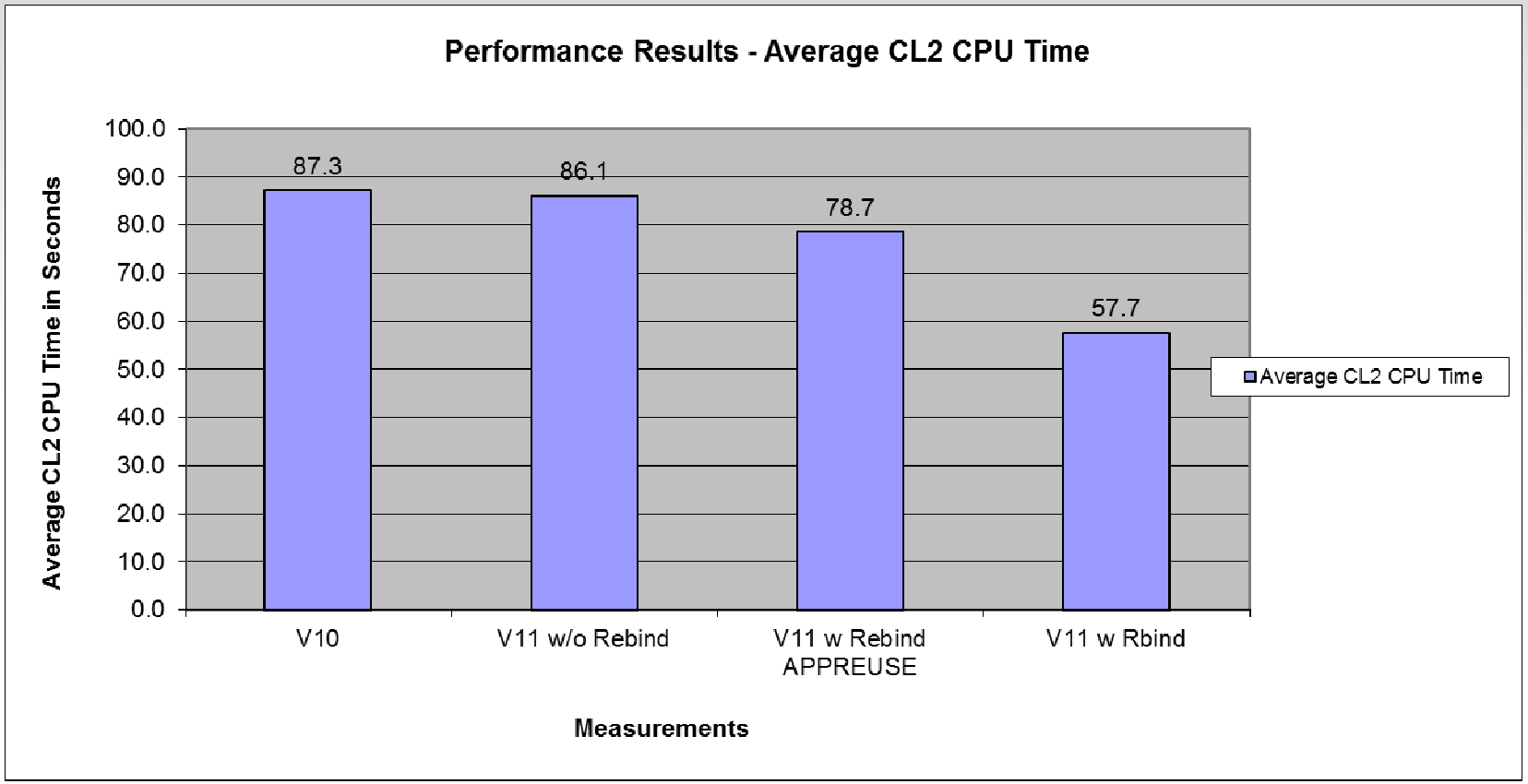
DB2 10 for z/OS EOM: July, 2015

DB2 10 for z/OS EOD: September, 2017

Impressive DB2 11 Performance Results!



TPC-H Using Static SQLPL



-10% out-of-box improvement with DB2 11 when rebinding with APREUSE
- 34% improvement in DB2 11 when rebinding to obtain DB2 11 AP

Static SQLPL TPCH Workload

44 SQLPL Stored Procedures, each executes 1 TPCH query with degree any

22 queries using Literals, 22 queries using SQL Variables

An application program written in C calls the Stored Procedures

Performance Improvements
no REBIND needed – Partial List

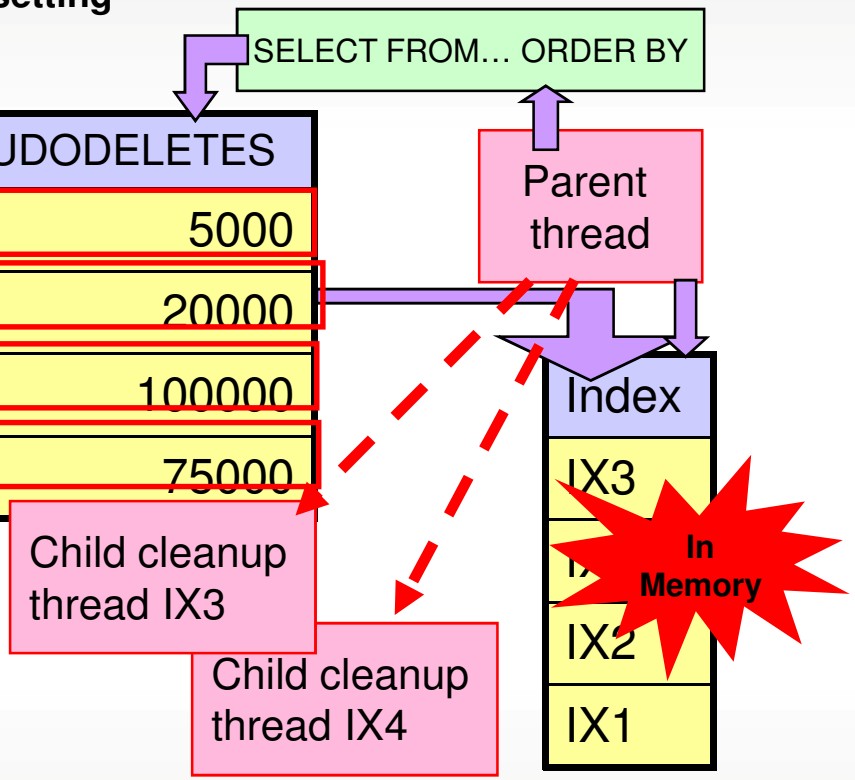
- DDF performance improvements
  - Reduced SRB scheduling on TCP/IP receive using new CommServer capabilities
  - Improved autocommit OLTP performance
- INSERT performance
  - Latch contention reduction
  - CPU reduction for Insert column processing and log record creation
  - Data sharing LRSN spin avoidance
- Automatic index pseudo delete cleanup
- IFI 306 filtering capabilities to improve Replication capture performance
- DGTT performance improvements
  - Avoid incremental binds for reduced CPU overhead
- Utilities performance improvements
- Java stored procedures: multi threaded JVMs, 64-bit JVM – more efficient

Pseudo-deleted Index Cleanup

- **Cleanup done under system tasks, run as enclave SRBs and zIIP eligible**
  - **Parent thread (one per DB2 member) reads through RTS to find candidates**
    - **Runs every 5 minutes**
  - **Eligible indexes sorted based on number of pseudo deleted rids to delete highest first**
  - **Child threads assigned based on ZParm setting**

SYSIBM.SYSINDEXSPACESTATS

NAME	…	NPAGES	…	REORGPSEUDODELETES
IX1	n	100	X	5000
IX2	n	1000	X	20000
IX3	n	500	X	100000
IX4	n	2000	X	75000



- Child cleanup thread only started if Index already open for INSERT, UPDATE or DELETE
  - 'X'-type P-lock already held

SYSIBM.SYSINDEXCLEANUP Example

- All index spaces in DB_1234 are enabled for cleanup on Sundays from 4:30 AM until noon, except
- Index space IX_9876 is always disabled for cleanup

SYSIBM.SYSINDEXCLEANUP

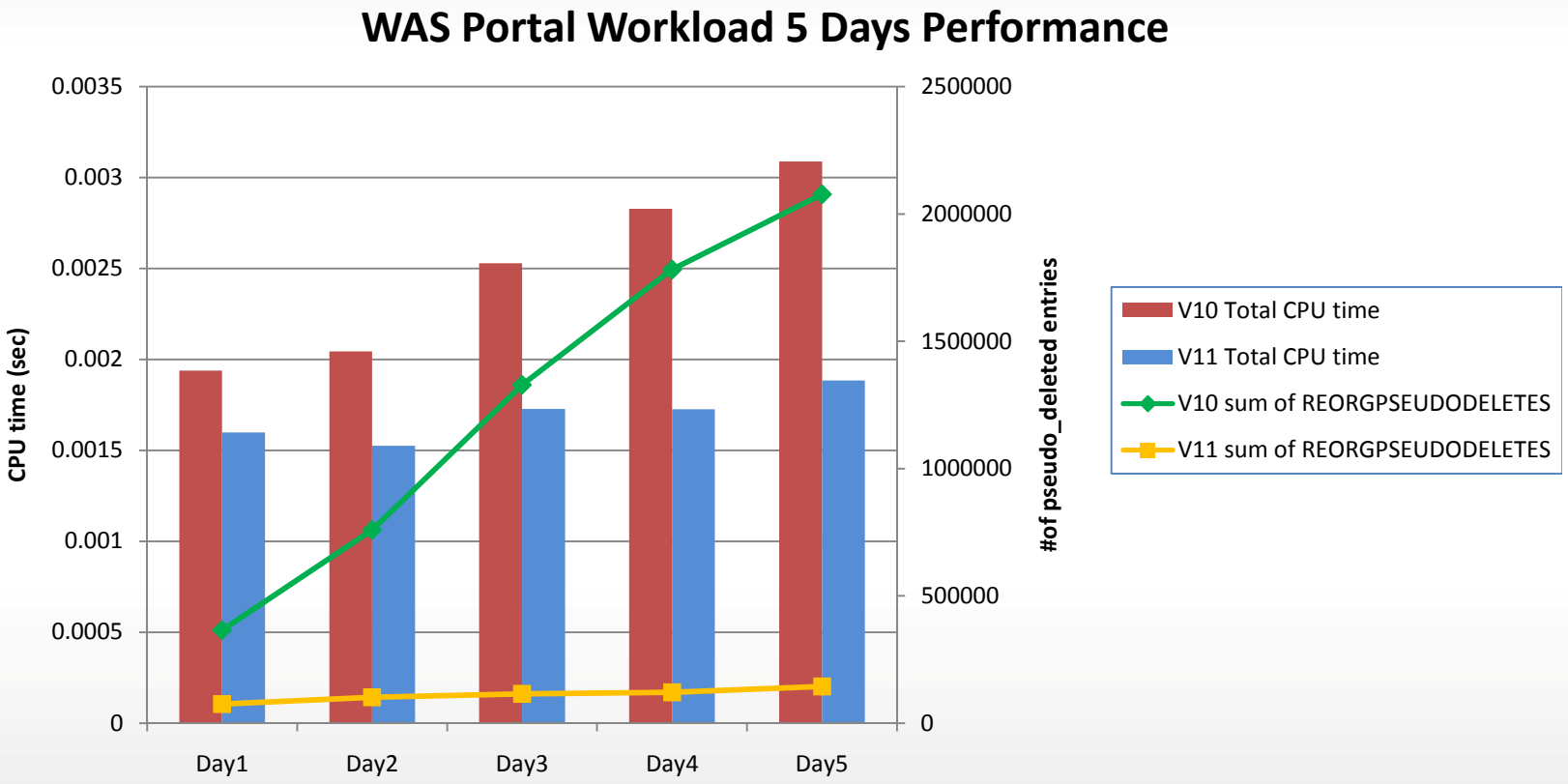
DBNAME	INDEX-SPACE	ENABLE_DISABLE	MONTH_WEEK	MONTH	DAY	START_TIME	END_TIME
DB_1234	NULL	E	W	NULL	7	043000	120000
DB_1234	IX_9876	D	NULL	NULL	NULL	NULL	NULL

- DB2 checks SYSIBM.SYSINDEXCLEANUP table at 10 min intervals
  - Enforcement of new row may be delayed up to 10 min
- RECOMMENDATION: Use rows in SYSIBM.SYSINDEXCLEANUP only to define exceptions to default index cleanup behavior
  - Define time windows at system or database levels, rather than specific indexes when possible
  - Remove unneeded or conflicting rows

DB2 11 Auto Pseudo Delete Cleanup

WAS Portal Workload

- Up to 39% DB2 CPU reduction per transaction in DB2 11 compared to DB2 10
- Up to 93% reduction in Pseudo deleted entries in DB2 11
- Consistent performance and less need of REORG in DB2 11



Performance Improvements
REBIND required – Partial List

- Query transformation improvements – less expertise required for performant SQL
- Continuous Block fetch
- In-memory techniques
  - In-memory, reusable workfile
  - Sparse index (limited hash join support)
  - Non-correlated subquery using MXDTCACH
  - Correlated subquery caching
- Select list do-once
  - Non column expressions in the select list can be executed once rather than per-row
    Example: CURRENT TIMESTAMP - MONTH
- Column processing improvements
  - Xproc (generated machine code) for column processing, moved above the bar
- DPSI performance improvements
- Data de-compression optimizations

Performance Improvements
Sysprog, DBA, or application effort required – Partial List

- Suppress-null indexes
  - Index entries not created when all values for indexed columns are NULL
  - Reduced index size, improved insert/update/delete performance, compatibility with other DBMSes
  - Improved utility and CREATE INDEX performance
- New PCTFREE FOR UPDATE attribute to reduce indirect references
- DGTT performance improvements
  - Non logged DGTTs
- Extended optimization - selectivity overrides (filter factor hints)
  - Improve optimizer's ability to find the cheapest access path
  - Collect filter factors for predicates in a Selectivity Profile
- Open data set limit raised to 200K

Summary of Utilities Improvements

- Over 40 new enhancements!
- Availability
  - Online data repartitioning
    - REORG REBALANCE SHRLEVEL(CHANGE)
    - Online ALTER of limit keys
  - Online REORG availability improvements
    - SWITCH phase reduction
    - Improved drain processing
  - Part level inline image copies for REORG
- Usability
  - Online REORG automated mapping tables
  - REORG delete unused PBG data sets
  - System cloning improvements
- CPU reduction
  - More zIIP offload for LOAD and RUNSTATS
- Performance
  - Faster LOAD processing
  - Inline statistics improvements, reduced need for RUNSTATS
  - Optimizer input to statistics collection
  - REORG option to avoid sorting data for clustering
  - DSNACCOX performance

Key utilities performance numbers

- Up to 81% zIIP-eligible CPU with RUNSTATS COLGROUP
- Up to 40% zIIP-eligible CPU in REORG & LOAD with inline distribution stats
- REORG SWITCH phase outage reduced by up to 91%
- Up to 71% elapsed time reduction for REORG of subset of partitions
  - SORTNPSI option retrofitted to V9 & V10
- RECOVER from part-level image copies reduced CPU by up to 50%, elapsed by up to 40%
- LOAD from single input data set elapsed time reduced by up to 70%
- Crossloader support for FETCH CONTINUE for LOB & XML data
  - 28% CPU reduction

Application Compatibility ...

- Occasionally SQL functions change behavior
  - Usually in support of family compatibility and SQL standards
  - This introduces an application incompatibility which can
    - Delay version migrations
    - Potentially create a single version charge issue
  - These are documented in the Release Incompatibilities

- Example DB2 10 incompatibilities & resolutions
  - CHAR function results (also for VARCHAR and CAST of these data types)
    - Leading zeroes no longer returned when there is a decimal point
    - ZParm BIF_COMPATIBILITY was introduced to reverse the V10 behavior
    - IFCID 366 introduced to report on use of V9 code path while in V10 ZParm enabled
  - Strong data typing for .NET stored procedures
    - ZParm DDF_COMPATIBILITY reversed this behavior
  - Acceptance of unsupported Timestamp formats was also reversed by BIF_COMPATIBILITY
  - Others required application changes starting in Conversion Mode

Application Compatibility ...

- DB2 11 fences DML behavior change beginning in CM with APPLCOMPAT
  - Does not fence DDL or DCL
  - Separates the application migration from the system migration
    - Application migration can begin after the system migration is complete
    - Or the application migration can be delayed for up to 2 future DB2 versions
- APPLCOMPAT ZParm *and* Bind Parameter
  - V10R1
    - DML behaves as it did for DB2 10
    - Must use V10R1 until NFM
    - Attempting to use new features under V10R1 results in SQLCODE -4743

```
DSNT408I SQLCODE = -4743, ERROR:  ATTEMPT TO USE A FUNCTION WHEN THE
         APPLICATION COMPATIBILITY SETTING IS SET FOR A PREVIOUS LEVEL
```

  - V11R1
    - Requires the subsystem / group to be in NFM
    - New DML behavior is introduced
    - Also required for new features of V11

Application Compatibility ...

- ZParm APPLCOMPAT is the default for the BIND / REBIND parameter
  - Has no effect on existing Packages OR REBIND PACKAGE w/ APPLCOMPAT is set
  - Defaults to V11R1 for installations and V10R1 for migrations
- BIND / REBIND parameter
  - Defaults to ZParm APPLCOMPAT
    - After the previous setting for REBIND (Application Compatibility level already set)
  - V11R1 will cause an error before NFM
    - Once NFM, can BIND / REBIND with V10R1 or V11R1 irrespective of the ZParm
  - REBIND existing application when ready for *new behavior*
  - REBIND existing application when wanting to use *new features*
    - WARNING: Must also be ready for potential behavioral changes
- CURRENT APPLICATION COMPATIBILITY special register for dynamic SQL
  - Defaults to Package APPLCOMPAT
  - Cannot be set to V11R1 until NFM
    - Once NFM, can bet set to V10R1 or V11R1 irrespective of the Package or ZParm settings
  - Can SET via a System Profile with PM93658

Application Compatibility

- Instrumentation
  - When using the V10R1 compatibility level, 3 IFCIDs are produced to identify application at risk at the V11R1 level
    - IFCID 239 enhanced to report on Packages that use the V10 code path
    - IFCID 366 (also in V10) identifies Statements in Packages that use V10 code path
    - IFCID 376 reports the same as 366, but attempts to eliminate duplicates
- Considerations
  - Behavioral change is possible in the down level compatibility level when needing to conform to SQL Standards
  - This does not fence DDL or DCL, in NFM (system) / V10R1 (package)
    - A Global Variable can be created (DDL) and have authority granted (DCL)
    - But applications cannot SET or reference the Global Variable until V11R1
  - Build your NFM plan to adopt V11R1 for applications

Extended RBA Problem Statement

- DB2's Relative Byte Address (RBA) for logging is 6 bytes
  - Gives 256TB of log record addressing capacity per DB2 subsystem/member
- With heavy sustained logging rates, DB2 can exhaust the 6-byte RBA
  - DSNJ032I and DSNJ033E warning messages
    - Alert-level = 'WARNING' when RBA reaches x'F00000000000'
    - Alert-level = 'CRITCIAL' when RBA reaches x'FFFF00000000'
  - Manual recovery actions are needed
    - Data Sharing: shut down the affected member and start a new member in its place
    - Non Data Sharing: reset all PGLOGRBA values back to zero (extended outage)
    - Documented in the DB2 Administration Guide
  - If alert-level reaches 'CRITICAL' then DB2 terminates to protect data integrity and force recovery actions
    - Reason code 00D10251
    - ACCESS(MAINT) restart allowed to prepare for recovery actions

Extended LRSN Problem Statement

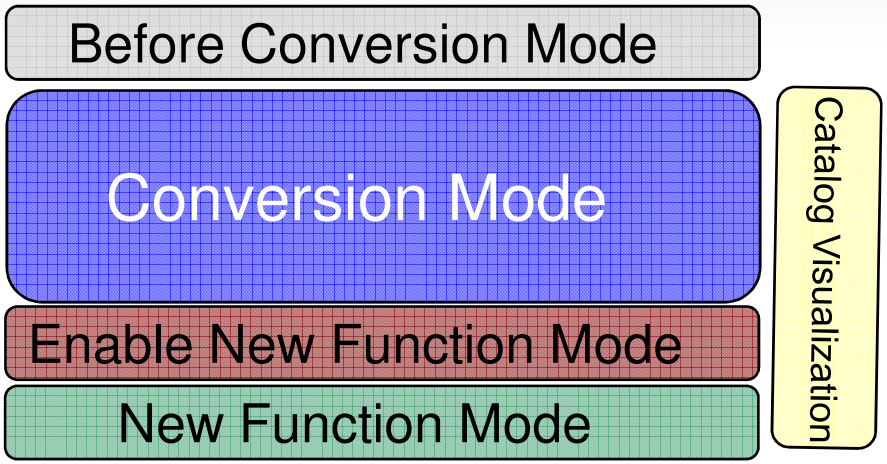
- The data sharing Log Record Sequence Number (LRSN) is derived from the 8-byte time-of-day clock which hits end of range in 2042
- However, some data sharing groups have a non-zero LRSN “delta” which gets added to the TOD clock
  - If a non-zero “delta” exists, then the LRSN will hit end of range prior to 2042
  - Use DSNJU004 to determine if you have a non-zero LRSN delta value
  - A “delta” value could be set when data sharing is enabled or re-enabled
    - Whenever the end-of-log RBA of the enabling member is past the TOD clock
- Some non data sharing customers have enabled data sharing to circumvent RBA nearing end-of-range
  - This would cause a non-zero LRSN delta, so LRSN hits end of range before 2042
- 6-byte LRSN value has precision to only 16 microseconds
  - Can cause LRSN ‘spinning’ which burns extra cpu and aggravated log latch contention
  - V9 NFM addresses most LRSN spin situations, and V10 NFM enhances further. But some spins still exist due to the 16 usec granularity (log latch not held, page latches are)

DB2 11 for z/OS RBA/LRSN Solution

- Expand the RBA and LRSN to 10 bytes
  - RBA addressing capacity of 1 yottabyte (2**80)
  - LRSN extended on left by 1 byte, on the right by 3 bytes
    - >30,000 years and 16Mx more precision
  - 8 bytes is not sufficient to solve LRSN issues and may not give sufficient capacity for the longer term
- NFM only (6 byte RBA/LRSN continues to be used in CM)
- Once in NFM, DB2 continues to use 6-byte values until you take action to convert
- Two conversion tasks:
  - Convert BSDSes to new format to enable logging with larger RBAs/LRSNs
  - Convert pagesets to new page format
- These tasks are optional
  - If you don't care about larger RBAs/LRSNs then you don't have to convert
  - But performance will be better if you convert BSDSes (avoid internal conversion overhead on log write)
- BSDSes can be converted without converting pagesets
- Pagesets can be converted in a piecemeal fashion
  - Expectation is that most customers will roll the conversion over a period of days/weeks/months

Migration Process

- **Preparations**
- **Customization / Tailoring**
- **Migration**
  - **Before CM**
  - **Migration to CM**
  - **Fallback to V10**
  - **Remigration to CM**
  - **Conversion to ENFM**
  - **Conversion to NFM**
  - **Application Compatibility**
  - **Extended RBA / LRSN**
- **Testing**



Note: These steps will iterate

Application Compatibility ...

10 NFM	11 CM	11 ENFM	11 NFM
	• ZParm V10R1 – Can set to V11R1 but will not operate that way • BIND/REBIND – Must be V10R1 • CREATE/ALTER – Must be V10R1 • SET CAC* not available • IFCID 239/366/376	**Same as CM**	• ZParm • V10R1 or V11R1 • BIND/REBIND • V10R1/V11R1 available • BIND • Defaults to ZParm • REBIND & Autobind • Defaults to previous Catalog value first • Zparm second • CREATE/ALTER • V10R1 or V11R1 • SET CAC* available • New Features • Require V11R1

***CAC = CURRENT APPLICATION COMPATIBILITY**

APREUSE

- DB2 10 APCOMPARE / APREUSE stability across BIND & REBIND
  - APREUSE options were NONE/NO or ERROR
- DB2 11 provides APREUSE(WARN)
  - Attempts to hint the same access path for matched statements
  - When the access path cannot be maintained, a new one is calculated for those specific matching statements
  - Effectively operates at the statement level
  - The same considerations from DB2 10 carry forward
    - Uses the Explain Data Block, as does APREUSE(ERROR) and APCOMPARE(WARN / ERROR)
    - Some PLAN_TABLE columns are not hint-able (MATCHCOLS)
      - APREUSE(WARN) will allow the BIND / REBIND to continue
- DB2 10 APREUSE(ERROR) ... EXPLAIN(ONLY) may represent invalid plan
  - In DB2 11, this will rolled backed

```
TSTDB206.DSN8BH10.DSN8BC3,
           USE OF APREUSE RESULTS IN:
           0 STATEMENTS WHERE APREUSE IS SUCCESSFUL
           0 STATEMENTS WHERE APREUSE IS EITHER NOT SUCCESSFUL
             OR PARTIALLY SUCCESSFUL
           4 STATEMENTS WHERE APREUSE COULD NOT BE PERFORMED
           0 STATEMENTS WHERE APREUSE WAS SUPPRESSED BY OTHER HIN
```

```
DSNT232I  DB1R SUCCESSFUL REBIND FOR
           PACKAGE = TSTDB206.DSN8BH10.DSN8BC3.()
```

Statistics Feedback ...

- The Optimizer can provide feedback on missing or inconsistent statistics during
  - Access path selection
  - Explain processing
- Access path selection (BIND / REBIND / PREPARE)
  - Externalized to new Catalog table SYSSTATFEEDBACK
    - Table added in CM
    - Feedback is externalized asynchronously starting in NFM
      - On STATSINT interval, or
      - With ACCESS DB command
      - At the Table level
  - Controlled by ZParm STATFDBK_SCOPE (ALL)
    - ALL, DYNAMIC, NONE, STATIC
  - Additional control at the table level
    - SYSTABLES STATS_FEEDBACK which defaults to Y
  - Redundant recommendations are not created
  - RUNSTATS for an object removes these associated recommendations

Statistics Feedback ...

- Explain processing
  - Externalized to new EXPLAIN table DSN_STAT_FEEDBACK (DSNTESC)
  - Controlled by the existence of this table
    - SYSTABLES STATS_FEEDBACK does not control these recommendations
  - Statement level recommendations written synchronously with EXPLAIN
  - History is maintained
  - User managed clean up
- Does not apply to VOLATILE, DGTT, or CGTT
- Not seeing expected feedback in SYSSTATFEEDBACK?
  - Make sure it has been externalized
    - STATSINT interval
    - ACCESS DB(*) SPACENAM(*) MODE(STATS) will externalize on command
- Not directly consumable by RUNSTATS
- Good “Blog Entry” by Willie Favero on this new feature: http://it.toolbox.com/blogs/db2zos/statistics-feedback-in-db2-11-for-zos-59598

Statistics Feedback ...

NUMCOLUMNS

- # of columns in COLGROUP

COLGROUPCOLNO

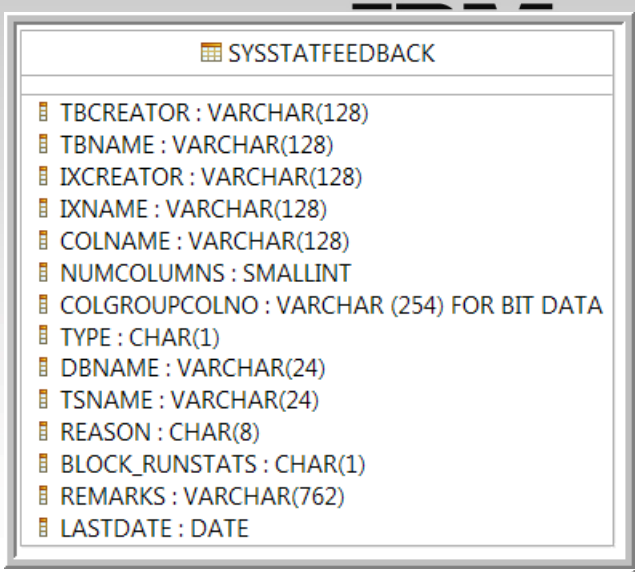
- Hex array of column numbers for tooling use

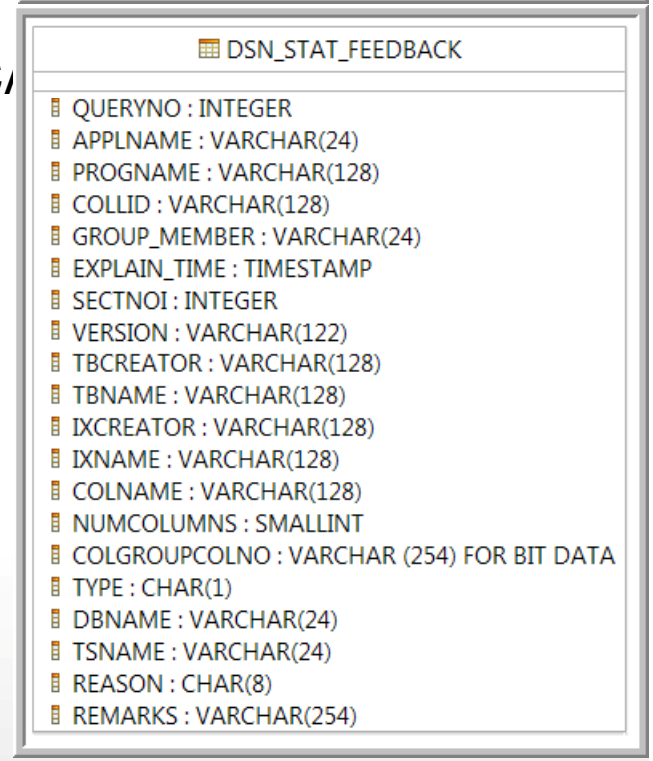
TYPE

- ‘C’ardinality
  - Use the COLUMN or COLGROUP option
- ‘F’requency
  - Use FREQVAL option (COUNT = Floor(10, COLC
- ‘H’istogram
  - Use HISTOGRAM option
- ‘I’ndex
  - Collect basic index stats
- ‘T’able
  - Collect basic table stats

BLOCK_RUNSTATS

- For use by tooling
- Not used by DB2





Statistics Feedback

REASON

- BASIC
  - TABLE(ALL) INDEX(ALL) will resolve, currently -1
- KEYCARD
  - Index column cardinalities missing
- LOWCARD
  - Low cardinality, likely skew
- NULLABLE
  - No distribution stats available, likely skew
- DEFAULT
  - Predicate likely on a default value
- RANGEPRD
  - No histogram on a range predicate
- PARALLEL
  - Could be improved by balancing data
- CONFLICT
  - Collect consistent stats at the same point in time
- COMPFFIX
  - Multi-column cardinality stats are needed

SYSSTATFEEDBACK

- TBCREATOR : VARCHAR(128)
- TBNAME : VARCHAR(128)
- IXCREATOR : VARCHAR(128)
- IXNAME : VARCHAR(128)
- COLNAME : VARCHAR(128)
- NUMCOLUMNS : SMALLINT
- COLGROUPCOLNO : VARCHAR (254) FOR BIT DATA
- TYPE : CHAR(1)
- DBNAME : VARCHAR(24)
- TSNAME : VARCHAR(24)
- REASON : CHAR(8)
- BLOCK_RUNSTATS : CHAR(1)
- REMARKS : VARCHAR(762)
- LASTDATE : DATE

DSN_STAT_FEEDBACK

- QUERYNO : INTEGER
- APPLNAME : VARCHAR(24)
- PROGNAME : VARCHAR(128)
- COLLID : VARCHAR(128)
- GROUP_MEMBER : VARCHAR(24)
- EXPLAIN_TIME : TIMESTAMP
- SECTNOI : INTEGER
- VERSION : VARCHAR(122)
- TBCREATOR : VARCHAR(128)
- TBNAME : VARCHAR(128)
- IXCREATOR : VARCHAR(128)
- IXNAME : VARCHAR(128)
- COLNAME : VARCHAR(128)
- NUMCOLUMNS : SMALLINT
- COLGROUPCOLNO : VARCHAR (254) FOR BIT DATA
- TYPE : CHAR(1)
- DBNAME : VARCHAR(24)
- TSNAME : VARCHAR(24)
- REASON : CHAR(8)
- REMARKS : VARCHAR(254)

Archive Transparency

- What is the purpose of archiving?
  - When you want to delete rows from the table, but need to keep the deleted rows for legal or business purposes
  - To move data to a cheaper storage medium
  - When you do not need to access the old data often, but need to be able to retrieve the data quickly
  - When you do not care about the lineage of a row
    - This means that you do not care about the changes to a row over time
- Do we add extra columns for archiving like we do for system time tables?
  - You do not need extra columns to enable Archive Transparency
- Temporal and Archive Tables are mutually exclusive
  - Can not build an Archive Table on a table that has either Business Time or System Time
- Archive a large amount of data using REORG DISCARD to facilitate migration
  - User would be responsible for loading data from the DISCARD file into the archive table

Archive Transparency Compared to System Time ...

	System Time	Archive Transparency
Schema -- two table approach	current table & history table same column #, column name, column attributes (data type, etc)	archive-enabled table & archive table same column #, column name, column attributes (data type, etc)
ROW BEGIN/ROW END/TRANS ID columns	mandatory	none
Period concept	yes – SYSTEM_TIME period	none, not compatible with STT
Compatible with Business Time tables (ATT)	yes	no
Bind option	SYSTIMESENSITIVE	ARCHIVESENSITIVE
Implicit union all query transformation	controlled by CURRENT TEMPORAL SYSTEM_TIME special register	controlled by built-in global variable SYSIBMADM.GET_ARCHIVE
Data propagation to history/archive table	UPDATE and DELETE	DELETE SYSIBMADM.MOVE_TO_ARCHIVE
Implicit Static DMLs	Implicit two section design	Implicit two section design

Archive Transparency

- These settings for BIND will control the sensitivity of the SYSIBMADM.GET_ARCHIVE global variable:
  - ARCHIVESENSITIVE (default YES) – packages (No space between ARCHIVE and SENSITIVE))
    - BIND PACKAGE
    - REBIND PACKAGE
    - REBIND TRIGGER PACKAGE
    - CREATE TRIGGER (implicit trigger package)
  - ARCHIVE SENSITIVE (default YES) – UDFs and Stored Procedures (space between ARCHIVE & SENSITIVE)
    - CREATE FUNCTION (SQL scalar)
    - ALTER FUNCTION (SQL scalar)
    - CREATE PROCEDURE (SQL native)
    - ALTER PROCEDURE (SQL native)
- If you REBIND a package and change ARCHIVESENSITIVE, all copies of the package will be purged
- APREUSE and APCOMPARE are valid options
- Has no affect on the SYSIBMADM.MOVE_TO_ARCHIVE variable

Archive Transparency Global Variables ...

- Built-in Global Variables that impact archival tables & processing
  - Defined as CHAR(1) NOT NULL DEFAULT ‘N’
  - READ authority granted to PUBLIC
  - SYSIBMADM.GET_ARCHIVE
    - Determines if SELECTs against Archive Enabled (Base) Tables automatically UNION ALL the associated Archive Table
    - 'Y‘ includes the UNION ALL to Archive Tables
  - SYSIBMADM.MOVE_TO_ARCHIVE
    - Determines if deleted rows of Archive Enabled Tables are inserted into associated Archive Tables
    - ‘Y’: INSERT and UPDATE not allowed against the Archive Enabled (Base) Tables
    - ‘E’: INSERT and UPDATE allowed against the Base Tables
    - ‘N’: DELETEd rows not inserted into Archive Table

Archive Transparency Example ...

Base Table

```
CREATE TABLE POLICY_BASE
(EMPL VARCHAR(4) NOT NULL,
 TYPE VARCHAR(4),
 PLCY VARCHAR(4) NOT NULL,
 COPAY VARCHAR(4),
 START_DATE   DATE NOT NULL,
 TIMESTAMP1   TIMESTAMP NOT NULL GENERATED ALWAYS
   FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP,
 PRIMARY KEY (EMPL,PLCY));
```

Archive Table

```
CREATE TABLE POLICY_ARCHIVE
(EMPL VARCHAR(4) NOT NULL,
 TYPE VARCHAR(4),
 PLCY VARCHAR(4) NOT NULL,
 COPAY VARCHAR(4),
 START_DATE   DATE NOT NULL,
 TIMESTAMP1 TIMESTAMP NOT NULL GENERATED ALWAYS
   FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP,
 PRIMARY KEY (EMPL,PLCY));
```

OR

```
CREATE TABLE POLICY_ARCHIVE
  LIKE POLICY_BASE
  INCLUDING ROW CHANGE TIMESTAMP;
```

Activate archiving

```
ALTER TABLE POLICY_BASE ENABLE ARCHIVE USE POLICY_ARCHIVE;
```

- Create the base table
- Create the archive table
- Tell DB2 to associate the base table with the archive table

Archive Transparency Example ...

- **Archive all rows where START_DATE less than December 31, 2010**

```
EMPL   TYPE   PLCY   COPAY   START_DATE   TIMESTAMP1                    EMPL_LAST_NAME
---------+--------+--------+--------+--------+--------+--------+--------+--------
A207   HMO    P667   $10     2007-01-01   2013-07-30-20.07.33.136488   --------------
A208   HMO    P667   $10     2008-01-01   2013-07-30-20.07.33.137805   --------------
A209   HMO    P667   $10     2009-01-01   2013-07-30-20.07.33.139949   --------------
A210   HMO    P667   $10     2010-01-01   2013-07-30-20.07.33.141584   --------------
A211   HMO    P667   $10     2011-01-01   2013-07-30-20.07.33.144117   --------------
A212   HMO    P667   $10     2012-01-01   2013-07-30-20.07.33.153135   --------------
```

Archive-enabled table has 6 rows

- We set the Global variable MOVE_TO_ARCHIVE to 'Y' and then issue the DELETE command where the START_DATE is prior to December 31, 2010

```
SET SYSIBMADM.MOVE_TO_ARCHIVE = 'Y';
DELETE FROM POLICY_BASE WHERE START_DATE < '2010-12-31';
```

- The rows that were deleted from the base table are inserted into the archive table
- The Timestamp in the Archive Table has the time the row was archived, not the time in the base table

```
SELECT * FROM POLICY_BASE;

EMPL   TYPE   PLCY   COPAY   START_DATE   TIMESTAMP1                    EMPL_LAST_NAME
---------+--------+--------+--------+--------+--------+--------+--------+--------
A211   HMO    P667   $10     2011-01-01   2013-07-30-20.07.33.144117   --------------
A212   HMO    P667   $10     2012-01-01   2013-07-30-20.07.33.153135   --------------

SELECT * FROM POLICY_ARCHIVE;

EMPL   TYPE   PLCY   COPAY   START_DATE   TIMESTAMP1                    EMPL_LAST_NAME
---------+--------+--------+--------+--------+--------+--------+--------+--------
A207   HMO    P667   $10     2007-01-01   2013-07-30-20.07.33.216716   --------------
A208   HMO    P667   $10     2008-01-01   2013-07-30-20.07.33.227317   --------------
A209   HMO    P667   $10     2009-01-01   2013-07-30-20.07.33.227768   --------------
A210   HMO    P667   $10     2010-01-01   2013-07-30-20.07.33.227787   --------------
```

Microseconds are greater in the archive table than the base (archive-enabled) table

Archive Transparency Global Variables ...

To SELECT data from both the base and archive tables

- Set the built-in global variable SYSIBMADM.GET_ARCHIVE to ‘Y’
- All subsequent SQL statements including those from invoked functions, stored procedures, and triggers will access both the base and archive table to retrieve the data via DB2 adding a UNION ALL predicate for the two tables
- Package must be bound with ARCHIVESENSITIVE(YES)

To Access data from only the base table

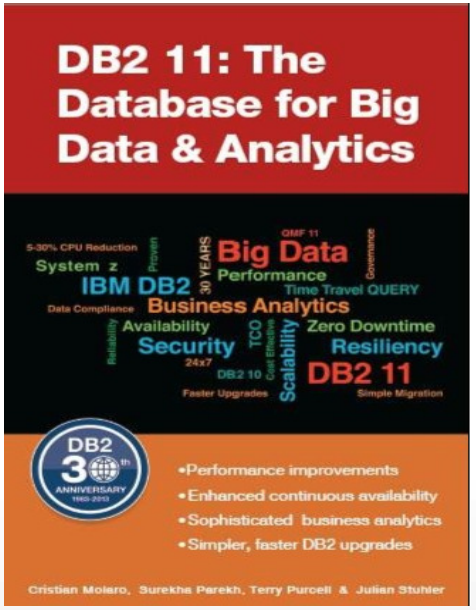
- Set the built-in global variable SYSIBMADM.GET_ARCHIVE to ‘N’ (Default)
- All subsequent SQL statements including those from invoked functions, stored procedures, and triggers will access data from only the base tables

DB2 11 Planning

- Dual mode migration (CM, ENFM, NFM)
- DB2 10 is the platform for migration
- z/OS 1.13 or above. z10 or above.
- No pre-V9 bound packages
- DB2 Connect V10.5 FP2 is the recommended level for V11
  - This level is required to exploit most new V11 features
  - Any in-service level DB2 Connect supports V11
- Sysplex query parallelism support is removed
- DB2 11 Migration Planning Workshop (MPW)
  - Free, 1-day education
  - DB2 11 MPW Community on DeveloperWorks

DB2 11 Resources

- Knowledge Center
- DB2 11 Technical Overview Redbook (SG24-8180)
  - Draft version available, final version coming soon.
- DB2 11 links: https://www.ibm.com/software/data/db2/zos/family/db211/
  - Link to DB2 11 Announcement Letter
  - Links to webcasts
  - Customer case studies
  - Whitepaper: "DB2 11 for z/OS: Unmatched Efficiency for Big Data and Analytics"
  - Whitepaper: "How DB2 11 for z/OS Can Help Reduce Total Cost of Ownership"
- DB2 11 Migration Planning Workshop
  - http://ibm.co/IlJxw8
- Free eBook available for download
  - http://ibm.co/160vQgM



DB2 Cypress Themes

- In-memory processing
  - HW/SW integration into the future on z
  - Out-of-the-box performance improvement
- “Mobile-scale” data bases
  - More schema flexibility
  - Extreme scale tables, indexes
  - Higher data ingest rates
- Cloud enablement
  - Developer self-service, cloud-based provisioning, deployment
  - Self-optimizing system
    - More transparent SQL optimization
    - Temporal catalog for powerful problem diagnosis capabilities
    - Easier management of large tables
- Analytics and BigData
- Extend System z leadership for continuous availability

Thank You