

## **DB2 11 for z/OS Technical Overview**

**John J. Campbell and Joachim Limburg**

*DB2 for z/OS Development*

Session Code: A01/B01 and A02/B02

Monday 14<sup>th</sup> October 2013 10:45am – 1:00pm | Platform: DB2 for z/OS





## Abstract

- This session will provide a technical overview of the functions and features of the new DB2 11 for z/OS release. New functions and features related to performance, availability, utilities, security, analytics will be introduced and discussed.
  - Introduce and discuss major themes of new release
  - Introduce and discuss performance improvements and expectations
  - Introduce and discuss improvements related to query performance and management, availability, utilities, security, analytics



**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.*

**Performance Disclaimer:**

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



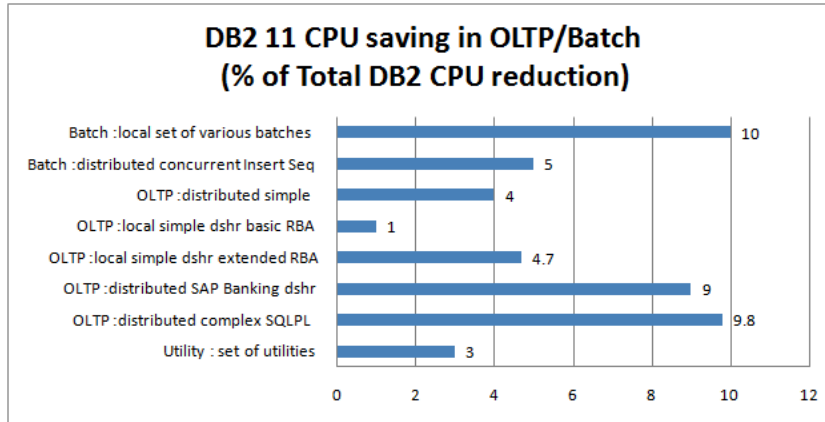
## Agenda

- General Performance
- Query Performance and Management
- Availability, Autonomics and Data Sharing
- Utilities
- Analytics
- Application
- Easier system upgrade
- Summary



## Performance Expectations for OLTP and Batch

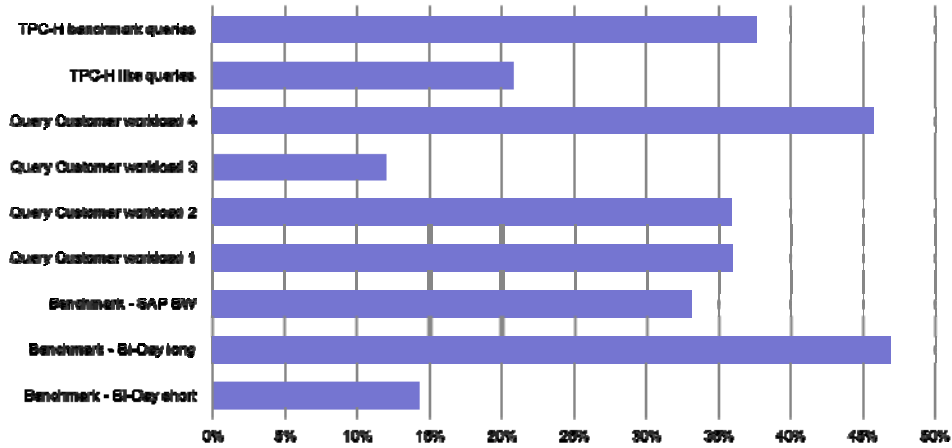
- Performance expectations vary depending on many factors, including
  - Access path selections, Read/Write rate, Number of rows returned
  - Number and type of columns returned, Number of partitions touched
  - Schema - Number of partitions defined, DPSI, etc
  - RELEASE option on BIND, Table Compression





## Performance Expectations for Queries

**DB2 11 Query Workloads - After REBIND w/o APREUSE**  
**% of DB2 Class 2 CPU Reduction from DB2 10**



- Most performance improvements are also available with APREUSE
- New and improved access path choices may be available without APREUSE



## Performance “Sweet Spots”

- Write Intensive Batch
- Queries
  - With compressed tables
  - With access path improvement
  - With sort intensive workload
  - Accessing multiple DPSI partitions
  - IDAA with large result sets
- Online transactions
  - Write intensive transactions
  - With large # of partitions (>200 partitions ) with RELEASE(COMMIT)
  - With large buffer pools
  - With queries returning a large number of columns
  - Chatty DDF applications with z/OS Communications Server PM80004 / UK92097
- Cost saving from zIIP eligible address space SRB time
  - DBM1 in data sharing
  - MSTR address space for update intensive workloads



## ESP Customer Performance Evaluations

- Several ESP customers sent DB2 11 vs. 10 performance data to SVL for analysis
- DB2 11 measurements are looking favorable
  - Majority of comparable workloads are batch jobs
  - Range of 5 to 20% CPU reduction in batch and OLTP workloads
  - Increase in zIIP usage in DBM1 in data sharing
  - zIIP usage in MSTR address space





## Performance Improvements – no REBIND needed

- DDF performance improvements
  - Reduced SRB scheduling on TCP/IP receive using new CommServer capabilities
  - Improved autocommit OLTP performance
  - DRDA package based continuous block fetch
- xProcs above the 2G bar
  - 31-bit Vstor relief enabled by RMODE 64 support in z/OS 1.13 and above
  - Enables other internal performance improvements
- zIIP enablement for all SRB-mode DB2 system agents that are not response time critical
- Avoid cross-memory overhead for writing log records
- Data decompression performance improvement
- INSERT performance
  - Latch contention reduction for classes 6, 14, 19
  - CPU reduction for Insert column processing and log record creation
  - Data sharing LRSN spin avoidance
  - Page fix/free avoidance in GBP write

9

Suppress-null indexes: EXCLUDE NULL KEYS on CREATE INDEX. Will prevent Index entries being created when **ALL** values for indexed columns are actually NULL

N244 DDF performance.

N2645 xProcs above the bar. z./OS 1.13 and above supports RMODE 64. Moving more control data into the SPAB (DN1607) to reduce the cost of moving a section from the EDM Pool (or Global Statement Cache) to thread storage can only be done if the xPROCs can be moved ATB.

N2895 DSCF performance.

N4504 General Performance improvements:

260 Reduce section storage movement (DN1607).

N265 – Insert space search improvements.

Insert: DB2 V10 has had a significant Insert performance enhancement, both in terms of CPU reduction and elapsed time, ie scalability enhancement, in general. DB2 V11 continues the same trend with contention and/or wait time reduction for latch class 6, 14, 19, and CPU reduction for Insert column processing, log record creation, page fix/free avoidance in GBP write, etc.

DSCF/IFC enhancements: Move WWFR, CCB above the bar



## Performance Improvements – no REBIND needed ...

- Automatic index pseudo delete cleanup
  - For fine-tuning, DBA work would be required
- ODBC/JDBC type 2 performance improvements
  - Stored procedure invocation
- Java stored procedure multi-threading improvements
- Sort performance improvements
- DPSI performance improvements for merge
- Performance improvements with large number of partitions
- XML performance improvements
- Optimize RELEASE(DEALLOCATE) execution so that it is consistently better performing than RELEASE(COMMIT)
  - Monitor # parent locks and cleanup internal structures when threshold is hit
- IFI 306 filtering capabilities to improve Replication capture performance
- Utilities performance improvements

10

Suppress-null indexes: EXCLUDE NULL KEYS on CREATE INDEX. Will prevent Index entries being created when **ALL** values for indexed columns are actually NULL

N244 DDF performance.

N2645 xProcs above the bar. z./OS 1.13 and above supports RMODE 64. Moving more control data into the SPAB (DN1607) to reduce the cost of moving a section from the EDM Pool (or Global Statement Cache) to thread storage can only be done if the xPROCs can be moved ATB.

N2895 DSCF performance.

N4504 General Performance improvements:

260 Reduce section storage movement (DN1607).

N265 – Insert space search improvements.

Insert: DB2 V10 has had a significant Insert performance enhancement, both in terms of CPU reduction and elapsed time, ie scalability enhancement, in general. DB2 V11 continues the same trend with contention and/or wait time reduction for latch class 6, 14, 19, and CPU reduction for Insert column processing, log record creation, page fix/free avoidance in GBP write, etc.

DSCF/IFC enhancements: Move WWFR, CCB above the bar



## Performance Improvements – no REBIND needed ...

- ACCESS DATABASE command performance
- DGT performance improvements
  - Avoid incremental binds for reduced CPU overhead
- P-procs for LIKE predicates against Unicode tables
- Improved performance for ROLLBACK TO SAVEPOINT
- zEC12 exploitation:
  - Pageable 1M size frames for buffer pool control structures
  - 2G page size frame size
  - 1M page size frames for DB2 code
    - Requires z/OS 2.1 or above and zEC12 Flash Express
- Latch contention reduction and other high n-way scalability improvements
- Data sharing performance improvements
  - LRSN spin reduction with extended LRSNs
  - Castout performance
  - GBP write-around
  - Index split performance

11

N269 Castout Improvements. Overlapping RFCO for pageset castout. Avoid sending page lists for GBP structure threshold and GBP checkpoint.

N207 – reduce data sharing index log force

ACCESS DATABASE: S10724 ACCESS DB runs under a separate service task so it doesn't cause queueing for other DB commands. S2927 ACCESS DB command to run in parallel

N254 Temp table performance/usability: not logged DGTTs – new NOT LOGGED option for DGTTs, default is LOGGED. Keep static SQL statements that use DGTTs prepared across commit; Explain on incrementally bound queries; Additional stats collection on temp tables.

Rollback to savepoint: in prior releases, performance degradation occurred due to increasing # of log records being scanned with each rollback request. In V11 we remember the point in the log where the previous rollback completed.

1M for DB2 code. Reduces TLB misses. 1.8% improvement for IRWW. Requires z/OS 2.1 or above.

Removed the latch contention for PB directory access (apar is opened to retrofit to V10)

Removed the latch contention for fast path get page (used by index mgr) to locate a root page



## Performance Improvements – REBIND required (with or without APREUSE)

- Query transformation improvements – less expertise required to write performant SQL
  - Enhanced query rewrite to improve predicate indexability
    - New situations where non-indexable predicates can be rewritten by Optimizer to be indexable
    - Convert some common stage 2 predicates to indexable (YEAR(), DATE(), SUBSTR(col,1,x), value BETWEEN COL1 AND COL2)
    - Improved indexability for OR COL IS NULL predicates
    - Push complex predicates inside materialized views/table expressions
  - Enhanced pruning of "always true" and "always false" predicates
- Enhanced duplicate removal
  - Lots of queries require duplicate removal: e.g. DISTINCT, GROUP BY, etc.
  - Dup elimination via sorting can be expensive
  - New techniques: Index duplicate removal, early out
  - Will not show in Explain table, need to look at IXSCAN\_SKIP\_DUPS column in DSN\_DETCOST\_TABLE to determine if sort avoided

12

Story 10699 – convert correlated subquery to non-correlated to exploit index access for Union/Union All (e.g. temporal) and some others

N2366 Index skipping: skipping over duplicates when an index provides order for DISTINCT or GROUP BY.

Expression evaluation: Optimized CASE expression evaluation, Sharing of repeat expressions (from view merge), Avoid repeat evaluation of non-column expressions

Rewritten version of predicate written to DSN\_PREDICATE\_TABLE (original predicate used if internally rewritten predicate is not valid syntax)



## Performance Improvements – REBIND required (with or without APREUSE) ...

- In-memory techniques
  - In-memory, reusable workfile
  - Sparse index (limited hash join support)
  - Non-correlated subquery using MXDTCACH
  - Correlated subquery caching
- Non correlated subquery with mismatched length
- Select list do-once
  - Non column expressions in the select list can be executed once rather than per-row
- Column processing improvements
  - Xproc (generated machine code) for output column processing
  - Optimized machine instructions for input/output column processing

13

“Limited hash join”: In a nutshell, our hash join implementation works well for joins to small tables - especially when very large tables are joined to small tables. It equally applies to non-correlated subqueries (extended memory usage in V11 - as pre-V11 we were always limited to sparse index of 32K), and small code tables, as well as DW style dimension tables.

"SELECT list do-once" refers to non-column expressions in the select list. Pre-V11 we executed these once per row, and now we execute them once. For example:

```
SELECT C1, CURRENT DATE - 1 MONTH FROM T1;
```

In the above example "CURRENT DATE - MONTH" will be evaluated once for the query, rather than once per qualified row (as in V10).

N195: Column processing improvements: xproc for column procedure for output column, it is applicable for distributed output column (select). Also code change on MVCDK for local input/output column processing



## Performance Improvements – REBIND required (with or without APREUSE) ...

- RID overflow to workfile handled for Data Manager set functions
  - DB2 10 added RID overflow to workfile
  - DB2 11 adds support for set functions (COUNT, MAX, MIN etc) which was excluded in DB2 10
- Performance improvements for common operators
  - MOVE, CAST, output hostvar processing, CASE, SUBSTR, DATE, others
- DECFLOAT data type performance improvements
  - Up to 23% CPU reduction for conversion to/from decfloat
  - Approx. 50% cpu reduction in INSERT, FETCH for decfloat columns
  - Helped further by zEC12 hw improvements for decimal floating point



## Performance Improvements – REBIND required (without APREUSE)

- DPSI and page range performance improvements
  - Page range screening for join/correlation predicates
  - Parallelism optimization for DPSI access
- Optimizer CPU and I/O cost balancing improvements
  - Measured results: 3% to >30% performance improvement for query workloads



## Performance Improvements – DBA or application effort required

- 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 CREATE INDEX performance
- New PCTFREE FOR UPDATE attribute to reduce indirect references
- DGT performance improvements
  - Non logged DGTs
- Global variables
  - Easier, more efficient sharing of data between SQL statements

16

Suppress-null indexes: EXCLUDE NULL KEYS on CREATE INDEX. Will prevent Index entries being created when **ALL** values for indexed columns are actually NULL

N244 DDF performance.

N2645 xProcs above the bar. z./OS 1.13 and above supports RMODE 64. Moving more control data into the SPAB (DN1607) to reduce the cost of moving a section from the EDM Pool (or Global Statement Cache) to thread storage can only be done if the xPROCs can be moved ATB.

N2895 DSCF performance.

N4504 General Performance improvements:

260 Reduce section storage movement (DN1607).

N265 – Insert space search improvements.

Insert: DB2 V10 has had a significant Insert performance enhancement, both in terms of CPU reduction and elapsed time, ie scalability enhancement, in general. DB2 V11 continues the same trend with contention and/or wait time reduction for latch class 6, 14, 19, and CPU reduction for Insert column processing, log record creation, page fix/free avoidance in GBP write, etc.

DSCF/IFC enhancements: Move WWFR, CCB above the bar





## Performance Improvements – DBA or application effort required ...

- Optimizer externalization of missing/conflicting statistics
  - Identify missing statistics during bind/prepare/explain
  - DBA or tooling to convert output to RUNSTATS input
- 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
  - Selectivity Profile is populated via BIND QUERY
- Open dataset limit raised to 200K

17

Suppress-null indexes: EXCLUDE NULL KEYS on CREATE INDEX. Will prevent Index entries being created when **ALL** values for indexed columns are actually NULL

N244 DDF performance.

N2645 xProcs above the bar. z./OS 1.13 and above supports RMODE 64. Moving more control data into the SPAB (DN1607) to reduce the cost of moving a section from the EDM Pool (or Global Statement Cache) to thread storage can only be done if the xPROCs can be moved ATB.

N2895 DSCF performance.

N4504 General Performance improvements:

260 Reduce section storage movement (DN1607).

N265 – Insert space search improvements.

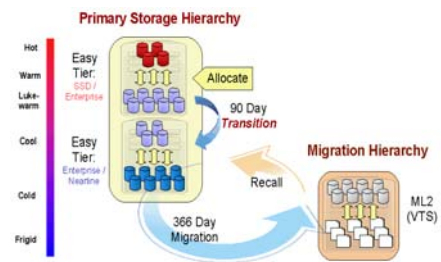
Insert: DB2 V10 has had a significant Insert performance enhancement, both in terms of CPU reduction and elapsed time, ie scalability enhancement, in general. DB2 V11 continues the same trend with contention and/or wait time reduction for latch class 6, 14, 19, and CPU reduction for Insert column processing, log record creation, page fix/free avoidance in GBP write, etc.

DSCF/IFC enhancements: Move WWFR, CCB above the bar



## DB2 Related enhancements in z/OS 2.1

- Enhancing DB2 BACKUP SYSTEM solution
  - Enable recovery of single tablespace from DB2 system-level backup even if original volume does not have sufficient space
  - Enable exploitation of FlashCopy consistency group for DB2 BACKUP SYSTEM
  - Enable restore of a tablespace from tape onto a different target volume
  
- Reduce latency between distributed application servers and DB2
  - Support for RDMA (Remote Direct Memory Access) to reduce TCP/IP stack (will require new hardware)
  
- z/OS DFSMS StorageTiers
  - Optimizes disk placement on SSD and HDD





## Buffer Pool Enhancements

- Extended MRU management for utilities sequential reads
  - Improves buffer hit ratios (reducing I/O) by avoiding the displacement of “useful” pages
  - COPY utility used MRU in V9, now extended to other utilities
- Enhanced classification of pages as “sequential” vs. “random” access
  - Dynamic and list prefetch plus sequential format writes classified as sequential
  - Random getpage which accesses a previously sequentially accessed buffer will now be reclassified as random
  - Improves accuracy of “random hit ratio” statistics
  - Should improve buffer pool hit ratios in many cases



## Buffer Pool Enhancements ...

- New real storage page frame size options
  - Larger page frame size improves performance by increasing TLB efficiency
  - FRAMESIZE(2G)
    - Requires new z/OS and new EC12 hardware support for 2G size frame size
  - New keywords to control the growth of buffer pool with AUTOSIZE enabled
    - VPSIZEMIN specifies minimum number of buffers to allocate
    - VPSIZEMAX specifies maximum number of buffers to allocate



## Query Performance and Management Improvements

- Optimizer externalization of missing statistics
- Query transformation improvements
  - Convert some common stage 2 predicates to indexable (YEAR(), DATE(), SUBSTR(col,1,x), value BETWEEN COL1 AND COL2)
  - Improved indexability for OR COL IS NULL predicates
  - Enhanced pruning of "always true" and "always false" predicates
- Index duplicate skipping
- Expression evaluation improvements
- Plan management improvements
  - APREUSE(WARN) support
- Selectivity overrides
  - Improve optimizer's ability to find the cheapest access path
  - Collect filter factors for predicates in a Selectivity Profile
  - Selectivity Profile is populated via BIND QUERY

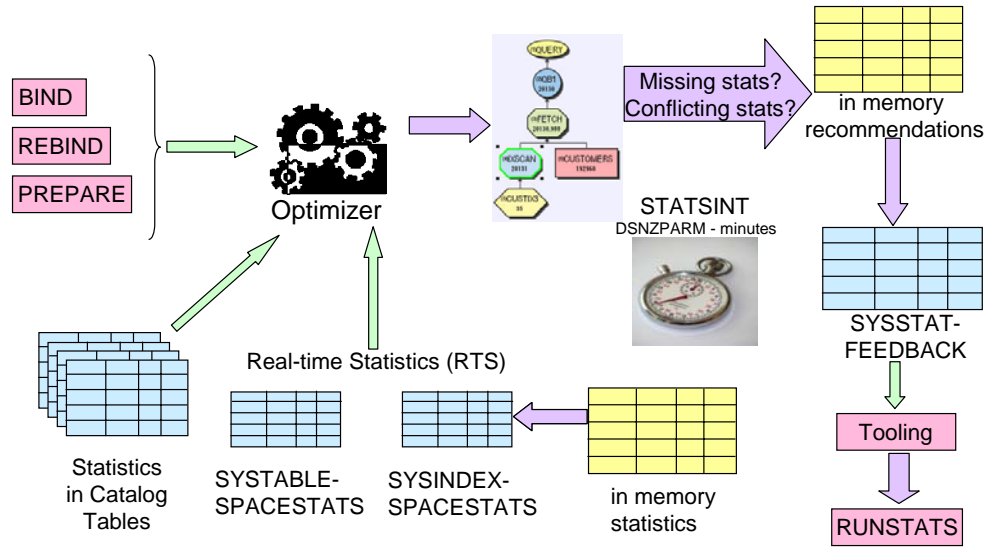


## DB2 Optimizer and Statistics - Challenge

- DB2 cost-based optimizer relies on statistics about tables and indexes
- Customers often gather only standard or default statistics
  - e.g. RUNSTATS TABLE(ALL) INDEX(ALL) KEYCARD
- Queries would often perform better if DB2 optimizer could evaluate more complete statistics
  - DB2 Lab estimates that >50% of query performance PMRs are solved by collecting the right stats
  - DB2 Lab estimates that most customers would see >10% CPU savings for queries if better stats were available
- Customers have difficulty knowing which statistics should be gathered
- This DB2 11 enhancement allows smarter decisions about which stats to collect
  - Allows users and/or tooling to provide interpret the output for improved RUNSTATS

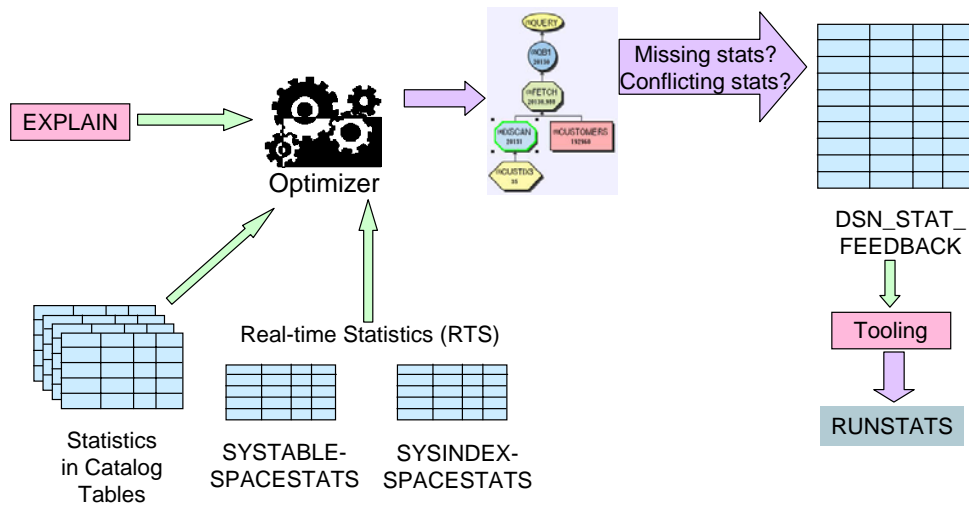


# Optimizer Externalization of Missing Statistics





## Using EXPLAIN





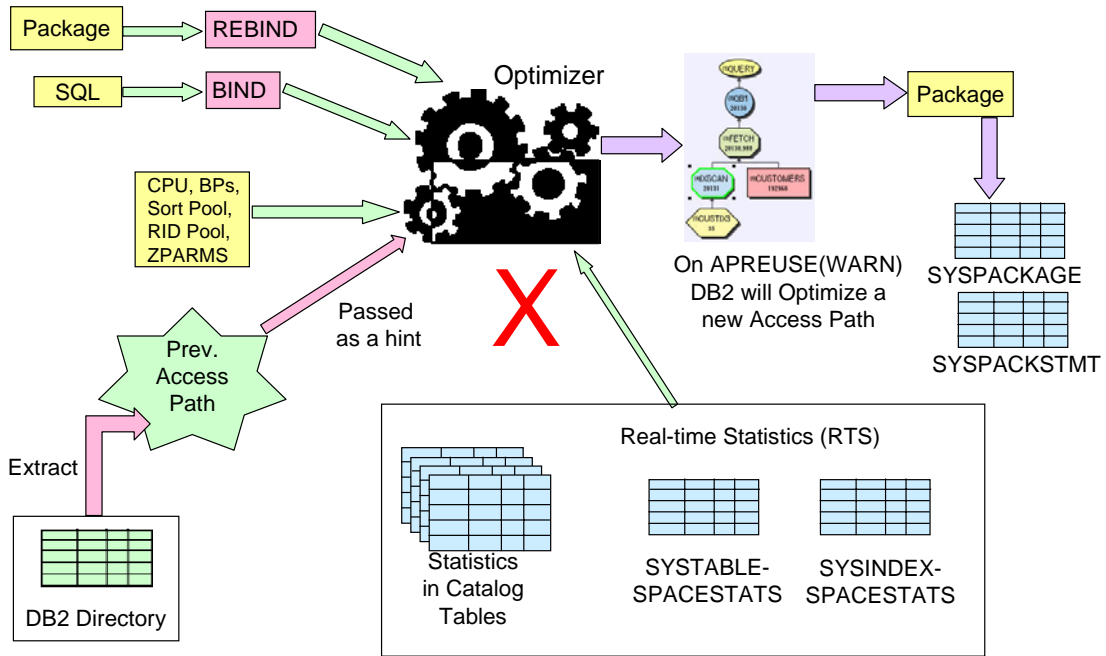


## Query Indexability Improvements

- Improve Query Performance by increasing the use of indexes
  - Improve indexability of commonly used stage 2 expressions
    - DATE, SUBSTR, YEAR
  - Remove “always true/false” literal expressions
    - For example – “always true” (WHERE 1=1)
    - Simplify queries containing Complex OR predicates with always false (OR 1=2)
    - NOTE: OR 0=1 is NOT removed
  - Reduce overhead with expressions in SELECT list  
Convert correlated subqueries to non-correlated subqueries when covered by local predicate
    - Further extending the DB2 10 enhancement to non-boolean term and UNION
  - Rewrite OR COL IS NULL predicates to single indexable
  - Indexability for CASE predicates
  - Predicate pushdown into materialized views/table expressions
    - DB2 11 adds support for stage 2 predicate, non-boolean term predicates, ON clause predicates



## BIND/REBIND PACKAGE – APREUSE(WARN) Failure





## Query Performance and Management Improvements ...

- In-memory techniques, e.g. open up sparse index, expand use of in-memory workfile
- RID overflow to workfile handled for Data Manager set functions
  - Adds RID overflow to workfile for set functions (COUNT, MAX, MIN etc) which was excluded in DB2 10
- Stage 2 predicate pushdown for RID LPF access
  - Adds stage 2 pushdown for list prefetch - whereby index manager or data manager (stage 1) could call stage 2 to evaluate an expression earlier
- Sort performance improvements
- zparm to control max storage allocation for sort
  - Range 1-128M, default=1M (same as V10)
- Reusable workfile support
- DPSI performance improvements
- EXPLAIN and virtual index improvements

27

N4505 – expanded use of in-memory workfiles

N2333 – exploiting sparse indexes whenever possible

S10817 – RID overflow to workfile improvements. Avoid TS scan for some cases, 80-99% improvement. More confidence in list prefetch, customers avoided list prefetch to prevent R-scan disasters (RID pool shortage). RID list improvements: Stage 2 predicate push down into Index Manager / Data Manager for list prefetch (single index only, no index ANDing ORing). RID overflow fallback to work file for Data Manager-evaluated aggregate function access type. Restrict Hybrid Join consumption of RID pool to 80%

Story 2351: predicate pushdown enhancements. The more filtering, the more cpu savings.

N4504/11836 Reusable workfiles. Reuse workfile for repeated access for correlated subqueries. S15913 to externalize in Explain (may not be resourced in rally7)

NL017 Sort performance: Reduced WF usage for small sorts, SPROC concept extended to fetch from sort: 1,5,7% cpu time improvement.

New ZPARM MAXSORT\_IN\_MEMORY. V10 used a fixed 1M value. Values between 1000K and 128,000K (default 1M). DB2 will use only the storage required by the query.

N1794: Explain improvements



## DPSI performance improvements

- Goal is to improve query performance for DPSIs such that customers can safely replace NPIs in more situations
- Targeted enhancements to increase DPSI “sweet spot”
  - CPU parallelism for joins to DPSIs
    - Parallelism cut on partition boundaries
    - Straw-model parallelism support for DPSI
  - I/O parallelism for single table DPSI access
  - Page Range Screening on join predicates
    - Access only qualified partitions
  - Improved join performance
    - Partition-level join with sequential access to join inner table
  - Enhancements to sort avoidance for DPSIs
- DPSI performance improvements are workload agnostic

28

### N231 – DPSI query performance.

DPSI Part-level (Nested Loop) Join Operation. Parallelism cut on inner table (at part/DPSI boundaries) such that each child task processes 1 DPSI part. Outer table is repeated for each child task. If <32K rows, outer table uses in-memory WF. Controlled by zparm PARAMDEG\_DPSI.

Straw-model parallelism support for DPSI. Straw-model (delivered in V10) implies that DB2 creates more work elements than there are degrees on parallelism.

DPSI parallelism zparm: PARAMDEG\_DPSI

When a table is partitioned on the join columns, then join performance will be enhanced if DB2 is able to use page range screening on the join predicates so that each inner table probe only accesses the necessary partitions. When a table is partitioned by non-join columns, and the index supporting the join is a DPSI, then join performance will be improved if each DPSI partition is able to be processed sequentially.

Enhancements to sort avoidance for DPSIs (Also known as DPSI return in order). Use of Index On Expression (IOE), ability to avoid sorting with DPSI IOE (already available for DPSI non-IOE). Index lookaside for DPSIs when order is provided



## Explain Enhancements

- DSN\_PREDICAT\_TABLE
  - Contains info about all the predicates in a query
    - More values in column ADDED\_PRED to indicate why DB2 added the predicate
    - 'T' Transitive closure, 'B' Bubbleup, 'P' Pushdown, 'L' Localisation, 'C' Correlation, 'J' Join, 'K' Like for IOE, 'S' Simplification, ' ' Predicate is not added by DB2
- DSN\_QUERY\_TABLE
  - Support extended for XML, before and after SQL shown for XML queries
- DSN\_VIRTUAL\_INDEXES
  - Columns added to complete index modelling capabilities
    - UNIQUE\_COUNT to support INCLUDE index columns
    - SPARSE to support NULL Supressed indexes
    - DATAREPEATFACTORF to support enhanced statistics gathering
    - KEY\_TARGET\_COUNT & IX\_EXTENSION to support Index on Expression and XML Index
- DSN\_VIRTUAL\_KEYTARGETS
  - New EXPLAIN table used for modelling IOE and XML indexes

29

The predicate table, DSN\_PREDICAT\_TABLE, contains information about all of the predicates in a query.

DSN\_QUERY\_TABLE – contains query text for before and after query transformation. XML introduced in DB2 9 now corresponding EXPLAIN process added. This table will now show XML query rewrite in the same way that it does with other index types.

Index Advisor tool has been enhanced to exploit the expanded virtual index capabilities.

UNIQUE\_COUNT similar to same column in SYSINDEXES. The number of columns or key targets that make up the unique constraint of an index.

SPARSE similar to same column in SYSINDEXES Null suppressed indexes allow an index to NOT have an index entry when all key columns that comprise the index contain the NULL value.

DATAREPEATFACTORF similar to same column defined in catalog – the anticipated number of data pages that will be touched following an index key order. This statistic is only collected when the STATCLUS subsystem parameter is set to ENHANCED.

KEY\_TARGET\_COUNT & IX\_EXTENSION similar to same column in SYSINDEXES. These columns describe the number and type of columns in IX\_EXTENSION Table. For each row in DSN\_VIRTUAL\_INDEXES that represents the creation of a virtual IOE. Corresponding rows need to be inserted in KEYTARGETS.



## Availability Improvements

- Extended RBA/LRSN expand to 10 bytes
- Increase 2G limit for a single internal DB2 storage pool
- BIND / DDL / Online REORG concurrency with persistent threads
  - Use of persistent threads has increases with DB2 10 with DBM1 ASID 31-bit VSCR
  - Examples: IMS PWF1, CICS protected entry
  - Avoid having to shut down these applications to get a REBIND through
- DEFER DEFINE improved concurrency
- More online schema changes
  - Alter partitioning limit keys
  - DROP column
  - Alter Drop Pending Changes: AREOR status is now removed
  - Point in time recovery support for deferred schema changes

30

N194: BIND and DDL break into persistent threads.

DM1616/S9135. DEFER DEFINE improved concurrency. Today, DBD lock used for serialization at insert/LOAD time when datasets allocated. In V11, it will be done via unrel EU switch.

N67: DROP COLUMN

N1211/6090 – PIT recovery support prior to materializing REORG

N1211/6088/6089 – PIT recovery for LOBs and XML prior to materializing REORG



## Extended 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 do not care about larger RBAs/LRSNs then you do not 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



## Bind / DDL / Online REORG break in to persistent threads

- DB2 11 delivers a break-in mechanism for persistent RELEASE(DEALLOCATE) threads
  - Persistent thread automatically during commit processing detects operations that would like to break in
  - If detected, then RELEASE(DEALLOCATE) will behave like RELEASE(COMMIT)
  - zPARM PKGREL\_COMMIT=YES must be set (online changeable)
    - Default is YES
  - Enhancement planned to handle idle threads which are not handled automatically
- Packages resume normal RELEASE(DEALLOCATE) behavior after the break-in operation completes

32

NFM is required. Zparm default is YES. COMMIT is still required to break-in.

The break-in mechanism only applies if  
no CURSOR WITH HOLD is open.

package is in addition not bound with KEEP DYNAMIC(YES).

the COMMIT does not occur in a stored procedure.

Note: These 3 restrictions apply also for packages originally bound with RELEASE(COMMIT). Refer also to next slides.





## Online ALTER Partition Limit Keys

- Currently: Affected partitions are set to REORP
  - These partitions cannot be accessed
  - REORG is run to redistribute the data and remove the status
- In DB2 11, Alter limit key is treated as a pending alter (NFM)
  - The affected partitions are set to AREOR
  - Online REORG must be run to materialize the pending changes
  - Supported table spaces types are:
    - UTS – partitioned by range (PBR)
    - Classic partitioned table spaces (table controlled partitioning)
  - The new limit keys are materialized in SYSTABLEPART in the SWITCH phase
  - Major restrictions
    - MQT, field-procedure, RI, index on expression, trigger

33

The online alter limitkey only applies to table partitioning. If the table is index-partitioning, then they will continue to incur an outage when they issue the alter.

Proposed DCR to add 2 new zparms: 1) A new zparm that would prevent alter limitkey against index-partitioned tables. This would allow customers to convert to table-partitioning before performing the alter. 2) A new zparm that would prevent the creation of new index-partitioned tables.

These two items would position us better for removing support for index-partitioning in future, and is also consistent with allowing customers to enforce best practices, exploit new function and avoid unnecessary outages.

ALTER TABLE ... ALTER PARTITION integer ENDING AT ...

Table spaces in which the partitioning is index controlled, must be first converted to table controlled. Conversion from index controlled partitioning to table controlled partitioning by e.g.

ALTER the CLUSTER attribute of the PI index (i.e. change to CLUSTER NO and change back to CLUSTER YES in the same commit).

New message DSNU2916I when the new limit keys are materialized in SYSTABLEPART.



## Drop Column: Materialization of the pending change

- Pending alteration
  - AREOR is set for the tablespace
  - Partitioned TS: Materialisation only if all partitions are addressed
- Invalidation of all packages
- PIT recovery is not allowed
  - SYSCOPY record with
    - ICTYPE=A (=alter)
    - STYPE=C (=column)
    - TTYPE=D (=drop)



## RAS Improvements ...

- Cancel DDF Threads – new FORCE option
  - Prior command without FORCE must be issued first
  - Only DDF threads
  - z/OS 1.13 APAR OA39392 required
- DRDA SQLCancel() improvements
  - Interrupt even when waiting on locks, executing SPs, or statement forwarded to another DB2
- Open data set limit raised to 200K
- Workfile space shortage warning new system parameters, instrumentation and messages
- Restrict hybrid Join to 80% of the total RID pool
- Query parallelism dynamic adjustment to available system resources
- Virtual storage scalability improvements
  - Shared memory object increased from 128G to 1T
  - Internal maximum for single storage pool lifted from 2G to 4G
  - More robust virtual storage allocation with query parallelism to avoid overruns

35

N2238/10630: Cancel DDF thread FORCE option. The FORCE keyword is only allowed after a non force cancel event has occurred, such as via the Cancel Thread command. The FORCE keyword is useful in rare situations, such as executing in relatively tight loop logic where no cancel detection points exist, where the DDF DBAT does not terminate after a standard cancel event. A new z/OS CALLRTM TYPE=SRBTERM service is being provided in z/OS 1.13 to allow DB2 CANCEL THREAD command processing to proactively cancel the thread when it's executing under an SRB. z/OS 1.13 APAR OA39392. New message DSNV519I.

N2238/10629: SQLCancel. *Interrupt any currently executing statement even when running in a stored procedure, executing a static statement, waiting for an IRLM lock or when the statement is forwarded to another DB2 server.* InterruptProcessingMode IBM Data Server Driver configuration keyword has values 0, 1 (default), 2:

0 = disable interrupt processing, SQL Cancel will not interrupt processing regardless of the server's capabilities

1 = interrupt is supported. If the server supports an interrupt then it will be sent.

2 = interrupt drops the connection regardless of the server's interrupt capability.

IBM has been recommended setting InterruptProcessingMode=2. A setting of 1 should now work ok with DB2 11.

S10817 Restrict hybrid Join to 80% of the total RID pool. Background: HYB join with SORTN = NO does frequent incremental processing of its accumulated RIDs, making it a "well-behaved" user of DB2's shared RID pool. HYB join with SORTN = YES ONLY does incremental processing when forced to by an out of RID resource type of condition, making it a



## RAS Improvements...

- Autonomics improvements
  - Automatic index pseudo delete cleanup
  - Overflow row reduction
  - Optimizer externalizes missing stats to enable automated RUNSTATS
- DDF enhanced client info fields for improved granularity
- New command option to externalize RTS stats (ACCESS DB)
- Performance monitoring improvements
  - zIIP time added to CPU trace header
  - Package detail for rollup accounting
  - Reduction in 'not accounted for' time for query parallelism
  - Accumulated transaction summary data by connection type (new IFCID 369)
  - More granular stored procedure and UDF monitoring

36

DN1627 – Specialty Engine time in CPU header

DN1664 – Rollup package detail. Benefit: package level detail even if rollup accounting enabled

PM62797 – retrofit of accounting data via statistics

PM53243 introduces new IFCID380 and IFCID381 to provide SP and UDF execution detail respectively. In addition, IFCID 497, 498, and 499 are created to identify the statements that are executed for non-nested, UDF, and SP work. The statements are identified by statement ID's that are comparable to those in IFCID401 or IFCID316.

N2846/S1502 IFC improvements. Reduction in 'not accounted for' time for query parallelism by recording time that parent waits for children.



## Auto Cleanup of Pseudo-deleted Index Entries

- Pseudo deleted index entries
  - Increases getpages, lock requests, CPU cost
  - Applications may encounter deadlocks and timeouts during update processing
- Prior to DB2 11, REORG INDEX required in most cases to remove pseudo-deleted index entries
- DB2 11 automatically cleans up pseudo deleted entries
  - zIIP eligible processing runs in the background
  - Designed to have minimal or no disruption to applications
  - New zparm to control number of concurrent cleanup tasks, default=10
  - New SYSIBM.SYSINDEXCLEANUP catalog table to control auto cleanup at index level
    - Day of week/month, start/end time.
    - By default cleanup is enabled for all indexes
- Benefits of automatic psuedo delete cleanup
  - Reduce size of some indexes, fewer getpages
  - Improve SQL performance, Lower CPU, less elapsed time
  - Reduce need to run REORG INDEX

37

Insert application timeout or deadlock when inserting into unique index with large number of pseudo-deletes. Extra locks have to be acquired on the pseudo-deleted entries when checking for uniqueness violation during insert. Often the RIDs are reused by insert, so can cause a deadlock situation.

Cleanup of pseudo empty pages introduced in DB2 10, but performed under the application thread, therefore cannot be aggressive (cost to thread) plus issues with span of commit

SYSIBM.SYSINDEXCLEANUP catalog table: Control of cleanup to avoid disruption. Object level and timing of cleanup. Recommended to use for exceptions only. Use SYSIBM.SYSINDEXCLEANUP to specify Name of databases and indexes, Cleanup enabled or disabled, Day of week or day of month, Start time and end time. By default index cleanup is enabled for all indexes if INDEX\_CLEANUP\_THREADS > 0. Sample to disable cleanup for all indexes

```
INSERT INTO SYSIBM.SYSINDEXCLEANUP(DBNAME, INDEXSPACE, ENABLE_DISABLE, MONTH_WEEK, MONTH, DAY, START_TIME, END_TIME)
```

```
values (NULL,NULL,'D', 'W', NULL, NULL, NULL , NULL ); Delay of up to 10 minutes before DB2 acts upon newly inserted row
```



## Overflow Rows and Indirect References – Review

- Row updates to variable length and/or compressed rows can increase the length of the row
  - If not enough space on the data page DB2 moves the row to another data page and replaces the original row with a pointer record
  - Index entries continue to refer to the original row (RID)
  - RTS indicators REORGNEARINDREF and REORGFARINDREF
- Reasons to avoid indirect references
  - Often causes additional I/O to read the extra data page into a buffer pool
  - REORG TABLESPACE to remove indirect references
- DB2 11 solution to reduce indirect references
  - New PCTFREE FOR UPDATE attribute to reserve free space for updates
    - Default is zero, or current behaviour
    - Value of -1 is the autonomic option – DB2 figures out optimal setting using RTS

38

N4504/S13763.

Example: SAP BW TS scan took >20 min. with lots of indirect references, after REORG it took 38 sec.

REORGNEARINDREF - The number of overflow records that are created and relocated near the pointer record

REORGFARINDREF - The number of overflow records that are created and relocated far from the pointer record since the last time the REORG or LOAD REPLACE utilities were run, or since the object was created. For non-segmented table spaces, a page is far from the present page if the two page numbers differ by more than 16. For segmented table spaces, a page is far from the present page if the two page numbers differ by at least  $(SEGSIZE*2)+1$ .

PCTFREE FOR UPDATE: all TS types except simple.

MAXROWS can also be effective, provided that most rows will eventually reach a maximum row size. Otherwise MAXROWS wastes space. But very few customers use it.



## Enhanced Client Information Fields

- Longer client information fields
  - Tolerated in CM
  - Exploited in NFM
- Supported
  - In DDF traces, messages and displays
  - Resource Limit Facility (RLF) can utilize
  - System Profile Monitoring can utilize
  - WLM classification rules can utilize
- New CLIENT\_CORR\_TOKEN Client Info
  - Customers can correlate application or monitor business processes by setting this special register
- New SYSIBM.CLIENT\_IPADDR built-in session global variable that represents the requester

39

### Clients can set new fields in DB2 11 CM, but cannot retrieve until NFM + APPLCOMPATE V11R1

NFM only. New long length support for clientUser(128), clientApplication(255), clientHostname/clientWorkstation(255), clientAccountingString(255).

Those values are pass in to DB2 during DB2 SignOn call and post DB2 SignOn via Set Client ID call.

DB2 for z/OS lengths for client information fields are incompatible with the corresponding fields on other DB2 family products. Some customer applications and monitoring tools must manipulate or truncate the client information values to fit in the fields supported by DB2 for z/OS. When this occurs, DBAs cannot use the full content of the client information fields, reducing their ability to monitor client performance.

CLIENT\_CORR\_TOKEN: Customers can correlate business processes by setting this special register.

SYSIBM.CLIENT\_IPADDR: DB2 obtains TCP/IP IPv6 address value from network if client does not provide it or set it.

A new WLM classify work request macro service, IWM4CLSY, introduced in z/OS V2R1, supports the new longer lengths client information fields to classify DDF server threads.

**If drivers at lower levels, then DB2 continues to use truncated client information fields even in NFM.** Clients can set new length client information fields when DB2 11 is running in CM. Clients cannot retrieve new length client information fields until DB2 is running in NFM and BIND option APPLCOMPAT (V11R1) set for static SQL or CURRENT APPLICATION COMPATIBILITY special register set to 'V11R1' for dynamic SQL.

Note: some client related lengths will **NOT** be enhanced:

START TRACE command filtering parameters [USERID, XUSERID, APPNAME, XAPPNAME, WRKSTN, XWRKSTN]. Users can continue to use partial filtering with asterisk (\*) to filter command output.

Qualification areas in DSNWQAL mapping macros (for READS through IFI).

Product section header QWHC and IFIC 0124 and 0148 **WILL** be enhanced to support low longer lengths; all other IFCIDs that describe client information data will continue to describe short (truncated) values

Message DSNT375I [deadlock] THREAD-INFO *thread-information1* and *thread-information2* – change would require change in IFCID 0172, which will not support the longer lengths. (see #3)



## Enhanced Client Information Fields ...

- Benefits
  - Full contents of client information fields available to DB2 for z/OS applications, tooling and DBAs
  - Improved granularity of monitoring, reporting and managing work
    - Accounting records
    - Messages
    - WLM workload classification
    - DB2-supplied stored procedures
    - RLF
    - System Profile Monitoring
    - Special registers
  - Compatibility with DB2 product family means more granular and meaningful data provided to DB2 for z/OS





## RAS Improvements...

- Data Sharing Enhancements
- Index Manager avoids RBLDP during group restart
- Fast log apply enabled during DB2 Restart
- SELECT from SPT01 & DBD01
- DESCSTAT BIND option
- New admin stored procedure to issue z/OS commands
- Compression dictionary handling for IFCID 306

N696 – enhanced client info fields

Externalize RTS stats: Challenge: running DSNACCOX before the latest RTS data has been externalized can lead to inferior recommendations.  
New command: -ACCESS DATABASE (DB2) MODE STATS.

N213 GBP write around. CFLEVEL=17 or above. z/OS apar OA37550, or z/OS 2.1.

N273/DN1639 – restart light option to remove all retained locks.

N4739 Index mgr reserve space (avoid rebuild pending indexes in unusual rollback situations)

N272 – Restart Performance

6098 – DESCSTAT BIND option, DN1596

DN1682 New stored procedure SYSPROC.ADMIN\_COMMAND\_MVS.



## Data Sharing Improvements

- Group buffer pool write-around to avoid CF cache structure flooding issues
- Castout enhancements:
  - Reduced wait time for I/O completion
  - Reduced notify message size sent to castout owner
  - More granular class castout threshold for large GBP size
- CF DELETE\_NAME utilizes a new CF request option to suppress XI signals when deleting directory entries
  - Improves efficiency of DELETE\_NAME especially for sysplex over extended distance
  - Retrofitted to DB2 10 and V9 with APAR PM67544
  - Adds a safety net to detect unexpected errors
- New LIGHT(CASTOUT) option on Restart Light
  - Causes all retained locks to be removed
    - Except in-doubt or postponed abort URs
  - Accomplished by initiating castout at end of Restart Light
    - After castout, pagesets become non GBP-dependent and retained page set P-locks can be safely released
    - Utilities can now be run after Restart Light completes

42

Avoid child 'U' lock propagation: v11 NFM sets the 'invert' bit for parent P-lock. IRLM child prop matrix is updated to suppress propagation when parent held S and no other interest. Applies for all TS types.



## Data Sharing Improvements ...

- Index split performance and other indexing improvements
  - Avoids placing indexes in RBDP during group restart in rare cases
    - Reduces DB2 outage time
    - Increases index availability
  - Improves index split performance
    - Reduces multiple log force write I/Os in data sharing for index split operation
    - Reduces multiple log force write I/Os for pseudo-delete operation
  - Improves index split rollback performance
    - Reduces backout time by reducing several log force write I/Os on rollback of deleted pages
- Auto LPL recovery improvements
- Full LRSN spin avoidance
- Avoid child 'U' lock propagation for single-member read-only

43

Avoid child 'U' lock propagation: v11 NFM sets the 'invert' bit for parent P-lock. IRLM child prop matrix is updated to suppress propagation when parent held S and no other interest. Applies for all TS types.



## Security Enhancements

- DB2/RACF authorization control enhancements
  - Invalidate cached authorization info and static SQL packages when RACF changes are made
    - New AUTHEXT\_CACHEREFRESH Zparm to activate the new behavior
  - AUTOBIND, BIND, REBIND present PKG-owner ACEE to RACF
  - Dynamic SQL authorisation checking
    - When DYNAMICRULES not equal to RUN, DB2 presents AUTHID to RACF
    - DYNAMICRULES defines whether AUTHID is
      - Package (PKG) owner
      - ID that defined the routine
      - ID that invokes the routine
  - New AUTHEXIT\_CHECK zparm to activate the new behavior
- Bind plan option to ensure the program is authorized to use the plan
- Remove column masking restrictions for GROUP BY and DISTINCT

44

N290 – RACF exit auth control enhancements: autobind under BIND owner's authid, auto auth cache refresh when RACF permissions are changed. Also invalidate static packages: If access control authorization exit is used, this function adds support for invalidating packages when a privilege required by the package is revoked and the zparm AUTHEXIT\_CACHEREFRESH is set. The package invalidation is supported when profiles with discrete names are revoked in RACF and the profiles are not in DSNADM class

N126/8552 – Remove GROUP BY restriction for tables with access control

N1398 - Remove pkg security vulnerabilities: plan owner has the ability to sign the program executable. DB2 will check the program signature when the program is loaded. (ITERGO). Problem: v10 requires packages. Once a user is granted execute privilege for an application plan, the user can execute any package identified in the package list. Application programs can be changed without the knowledge of the owner. The user can change the program to execute packages the owner never attended to be called without knowledge of the owner. This epic provides two stories to ensure an application program can not execute packages the owner never attended and to ensure a consistent approach to package authorization to reduce security vulnerabilities related to package execution. 10639: This story provides a new security mechanism to guarantee the program created by the plan owner is unchanged when the program is executed by DB2. 15699:

Story 10639 Support a bind plan option to ensure the program is authorized to use the plan.



## Column Masking Enhancements

- Remove column access control restrictions for GROUP BY
  - Remove SQLCODE -20478 reason code 5
    - If column mask contains a scalar fullselect or an aggregate function, allow to apply the column mask under an aggregate function
  - Remove SQLCODE -20478 reason codes 3,4,6
    - If column mask contains non-grouping columns, allow to apply the column mask when there is a GROUP BY clause
- Correct implementation of aggregate function with DISTINCT
  - V10: DISTINCT operation on masked value and aggregation on masked value
  - V11: DISTINCT operation on unmasked value and aggregation on masked value

45

Column masking is a function that was introduced in DB2 10. DB2 11 removes restrictions associated with SQLCODE -20478 reason code 3, 4, 5, 6.

THE STATEMENT CANNOT BE PROCESSED BECAUSE COLUMN MASK *<mask-name>* (DEFINED FOR COLUMN *<column-name>*) EXISTS AND THE COLUMN MASK CANNOT BE APPLIED OR THE DEFINITION OF THE MASK CONFLICTS WITH THE REQUESTED STATEMENT. REASON CODE *<reason-code>*.

3 – The column is also referenced in a grouping expression of the GROUP BY clause.

4 – Each column in the same table as *<column-name>* that is referenced in *<mask-name>* must also be referenced as a simple column reference in the GROUP BY clause. Such columns must not be referenced in a grouping expression in the GROUP BY clause.

5 – The select list also includes an aggregate function and the column is specified as an input argument to the function. In this case *<mask-name>* must not reference a scalar fullselect or an aggregate function.

6 – A column in the select list maps directly or indirectly to the column *<column-name>* of a table expression, view, or common table expression. Such a column must only be referenced as an argument to an aggregate function in the select list.



## Utilities Enhancements

- Better availability
  - Outage reduction for online REORG through improved drain processing & SWITCH phase elapsed time reduction
  - Better control of SWITCH phase timing
  - Online rebalance of data across partitions
  - Faster LOAD processing
- Better performance & reduced resource consumption
  - Reduce/eliminate need for RUNSTATS through inline stats
  - Faster LOAD processing
  - Faster recovery from part-level inline image copies
  - Faster REORG processing
  - More utility parallelism & greater parallelism control
  - More zIIP exploitation
  - Optimizer input to RUNSTATS



## Utilities Enhancements ...

- Simplified data management & improved usability
  - Automated REORG mapping table management
  - Intelligent default settings for REORG
  - Better PBG management through deletion of unused partitions
  - Improved dataset management with part-level inline image copies
  - Improved XML handling with Crossloader support
  - System cloning improvements
  - Lift some point in time recovery restrictions
  - More information & greater transparency through `-DISPLAY UTILITY` improvements
  - Enhanced statistics profile support



## Utilities Enhancements ...

- Online REORG
  - SWITCH phase outage window reduction and drain break-in enhancements
  - LOGRANGES YES|NO support for SYSLGRNX usage
  - SHRLEVEL CHANGE support for SORTDATA NO
  - RECLUSTER YES|NO option for SORTDATA NO
  - Automatic mapping table creation
  - Removal of mapping index size limitation
  - Deletion of trailing empty PBG partitions on a table space level REORG
  - Change to intelligent defaults: DRAIN ALL and NOPAD
  - Partition level sequential inline image copies
  - REBALANCE SHRLEVEL CHANGE support
  - REBALANCE redesign to better handle data skewing
  - SORTNPSI performance improvement on NPI processing (V9/V10 via APAR PM55051)
  - LISTPARTS n option to determine # of partitions to be processed together in LIST execution
  - Support for new online schema changes: drop column, online alter limit key
  - Enabling SHRLEVEL CHANGE REORG and COPY concurrency
  - Extended page format conversion via REORG
  - Inline DSTATS/HISTOGRAM collection with REORG





## New Analytics Features

- Query performance improvements
- Temporal data enhancements
  - Support for views
  - Special register support
  - Integrated auditing support (planned)
- Transparent archive query

N4309/8309 – temporal special register + archive transparency

N109/8960 IDAA heuristics improvements

N4629/8972 – IDAA RTS to indicate update after last refresh

N4629/8962 – IDAA WLM integration

N4629/8768 – IDAA enlarge query offload scope

N4629/4632 – IDAA insert with select

N4629/8963 – IDAA Explain enhancements with virtual accelerator

N4629/175 – IDAA static SQL support

N4629/8769 – IDAA temp tables support, and INSERT with SELECT on temp tables

N4629/9151 – IDAA Rowset cursor and multi row fetch support

N4629/4631 – IDAA subquery level offload

Hadoop access: Java table UDF returning all string columns for access to Hadoop file systems. LUW supports, but we need Java table UDF support for that. And we are planning to support the next level of generic table UDF in Java for Hadoop access, compatible with LUW's.



## Enhanced Temporal Support for Views

- Introduce support for period specification when a View is referenced in a FROM clause by SELECT, UPDATE, DELETE

- Example 1:

```
CREATE VIEW v01 (col1, col2, col3) AS SELECT * FROM stt;

SELECT * FROM v01
FOR SYSTEM_TIME AS OF TIMESTAMP '2010-01-10 10:00:00';
```

- Example 2:

```
CREATE VIEW v8 (col1, col2, col3) AS SELECT * FROM att;

UPDATE v8
FOR PORTION OF BUSINESS_TIME FROM '2009-01-01' TO '2009-06-01'
SET c2 = c2 + 1.10;

DELETE FROM v8
FOR PORTION OF BUSINESS_TIME FROM '2009-01-01' TO '2009-06-01'
WHERE COL1 = 12345;
```

50

Global variable + before trigger can give you auditing capability for history tables.

Allows period specifications on SQL statements that refer to VIEWS.

*stt* refers to a system-period temporal table, *att* refers to an application-period temporal table.



## Temporal Registers

- Provides an option to retrieve data from temporal tables without modifying existing queries
  - If the new special register CURRENT TEMPORAL SYSTEM\_TIME is used, DB2 rewrites the query adding the clause  
`FOR SYSTEM_TIME AS OF CURRENT TEMPORAL SYSTEM_TIME`
  - If the new special register CURRENT TEMPORAL BUSINESS\_TIME is used, DB2 rewrites the query adding the clause  
`FOR BUSINESS_TIME AS OF CURRENT TEMPORAL BUSINESS_TIME`
- An existing non-temporal query can be executed repeatedly while successively setting the temporal special registers to different points in time which would then return rows based on the implied temporal semantics
- Use of the new temporal registers depends on the new temporal Bind Options SYSTIMESENSITIVE and BUSTIMESENSITIVE

51

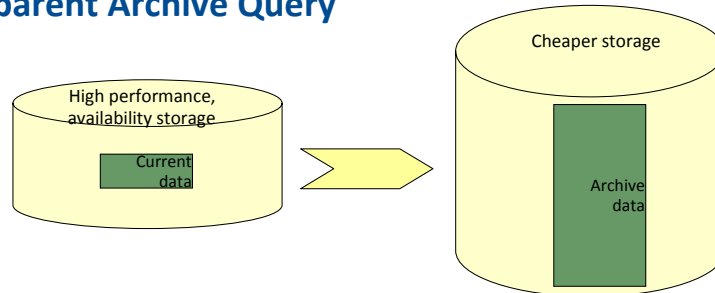
Exploitation of the temporal aspects of system time, business time, and bitemporal tables requires the specification of period clauses in all of the SQL statements that require access to historic portion of temporal data. This has been an inhibitor to some customers exploiting temporal tables for existing applications since modification of existing SQL statements is required. In some cases customers are not able to modify the application SQL statements because they are using 3<sup>rd</sup> party application code. Temporal registers provide an alternative to directly modifying SQL statements. DB2 can rewrite queries adding period clauses based on the contents of special registers.

For packages, the two new Bind options specify if query rewrite should occur for temporal tables when the related special register contains a valid date or timestamp.

For non-inline scalar functions and native SQL procedures, the two new routine options (SYSTEM\_TIME SENSITIVE and BUSINESS\_TIME SENSITIVE) specify if query rewrite should occur for temporal tables when the related special register contains a valid date or timestamp.



## Transparent Archive Query



- Applications can query current + archive with no SQL changes
  - By default, data is retrieved from base table only, as usual
  - Set a new global variable when archive data is desired
  - DB2 automatically converts SQL to UNION ALL via dynamic plan switching technique (high performance)
- Archiving process is user-controlled
- Move\_To\_Archive global variable allows DELETES to be automatically archived
- Leverages DB2 10 temporal constructs for archiving use cases
- Future potential for more IDAA synergy

52

Gives user the control to migrate "aged data" from the "current" table to "archive" table, ;  
 New DDL to relate the current table to the archive  
 Archive data is not used for uniqueness checking.  
 Add column is allowed and automatically propagated from base to archive, like bi-tempo  
 REORG DISCARD will fail for archive enabled table, or for system period temporal table  
 Proposal for BdoB requirement: Set MOVE\_TO\_ARCHIVE='E', in addition to 'Y' or 'N'. To a



## New Analytics Features

- SQL Grouping Sets, including ROLLUP, CUBE
  - ROLLUP is helpful in providing subtotaling along a hierarchical dimension such as time or geography
  - CUBE is helpful in queries that aggregate based on columns from multiple dimensions
- IFI 306 performance improvement for CDC and IDAA V3
- DB2 support for IDAA V3 (rolled back to V10)
  - Propagating DB2 changes to the accelerator as they happen
  - Detect staleness of data via RTS
  - Reducing disk storage cost by archiving data in the accelerator and maintaining the excellent performance for analytical queries: High Performance Storage Saver
  - Workload Manager integration and better monitoring capabilities
  - Increasing the query off-load scope via new special register CURRENT QUERY ACCELERATION
- High performance SPSS in-database scoring via PACK/UNPACK (rolled back to V10)

53

Hadoop access: REST API. BigInsights Fixpacks scheduled for May and next release is Oct. of 2013.

N139 Grouping Sets. Look at last line of example. In it are 3 groups, one result set for each of the 3 with one pass of the data.. Has been supported in LUW for a while. For more sophisticated groupings. Multiple GROUP BY in a single SELECT, for example:

```
SELECT T.YEAR, C. RETAILER, P.LINE,
       SUM(F.UNITSSOLD) AS UNITSOLD
FROM FACTVARS F, TIMELEVEL T, CUSTLEVEL C,
       PRODLEVEL P, CHENLEVEL CH
WHERE F.TIME_LEVEL = T.MONTH AND
      F.CUSTOMER = C.STORE AND
      F.PRODUCT = P.CODE AND
      F.CHANNEL = CH.BASE AND
      F.CHANNEL in ('G9GUPNDE0714')
```

GROUP BY GROUPING SETS(T.YEAR, C.RETAILER, P.LINE):

**SQL Rollup, Cube** Avoids convoluted SQL for this capability

**Rollup** is very helpful for subtotaling along a hierarchical dimension such as time or geography

```
ROLLUP(Year, Month, Day) =>
GROUPING SETS((Year, Month, Day),
              (Year, Month),
              (Year),
              ())
```

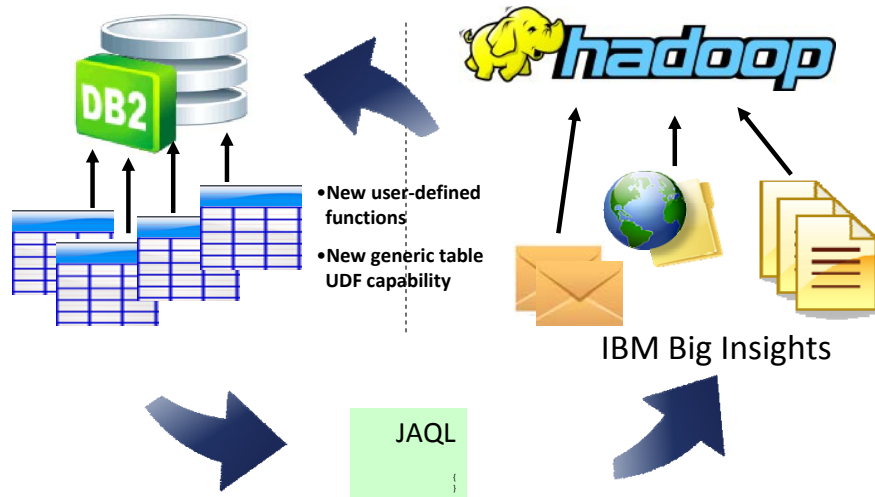
**CUBE** is helpful in queries that aggregate based on columns from multiple dimensions.

```
CUBE(YEAR, store, Line) =>
```



## Enhancing DB2 Analytics on “z” with Big Data

- DB2 is providing the connectors and the DB capability to allow DB2 applications to easily and efficiently access data in Hadoop



54

Big data and business analytics represent the new IT battleground. Here is some stats:

- IDC estimates the big data market will reach \$16.9 billion by 2015, and that enterprises will invest more than \$120 billion to capture the business impact of analytics, across hardware, software and services that same year.
- The "digital universe" will grow to 2.7ZB in 2012, up 48% from 2011 and rocketing toward nearly 8ZB by 2015 (IDC).
- 53% of business leaders don't have access to the information from across their organizations they need to do their jobs (IBM CMO Study).
- Organizations applying analytics to data for competitive advantage are 2.2x more likely to substantially outperform their industry peers (MIT/IBV Report)

The amount and types of data being captured for business analysis is growing. A classic example of this large superset of data is Web logs, which contain unstructured raw data.

In an increasing trend unstructured data is being stored on new frameworks. These infrastructures encompass hardware and software support such as new file systems, query languages, and appliances. A prime example being Hadoop.

### So what is Hadoop?

- A java-based framework that supports data intensive distributed



## Support for BigData

- Goal: integrate DB2 for z/OS with IBM's Hadoop based BigInsights Bigdata platform
  - Enabling traditional applications on DB2 z/OS to access Big Data analytics
- Analytics jobs can be specified using JSON Query Language (Jaql)
  - Submitted to BigInsights
  - Results stored in Hadoop Distributed File System (HDFS).
- A table UDF (HDFS\_READ) reads the Bigdata analytic result from HDFS, for subsequent use in an SQL query
- Must have a variable shape of HDFS\_READ output table
  - DB2 11 supports generic table UDF, enabling this function

55

The DB2 11 goal is to connect DB2 with IBM's Hadoop based BigInsights big data platform, and to provide customers a way to integrate their traditional applications on DB2 z/OS with Big Data analytics. Analytics jobs can be specified using JSON Query Language (Jaql) and submitted to IBM's Bigdata platform and the results will be stored in Hadoop Distributed File System (HDFS).

DB2 11 plans to integrate DB2 for z/OS with BigInsights from the database side and enable applications on DB2 z/OS to access big data analytics. It will include the ability to submit jobs specified in JSON Query Language (JAQL) to BigInsights and to access the Hadoop file system via user-defined functions.

(Remember that traditional table UDFs require that the output schema of the UDF is specified statically at function creation time. There would be a need to write a different external user-defined table function for reading each different Hadoop files which produce different output schema.

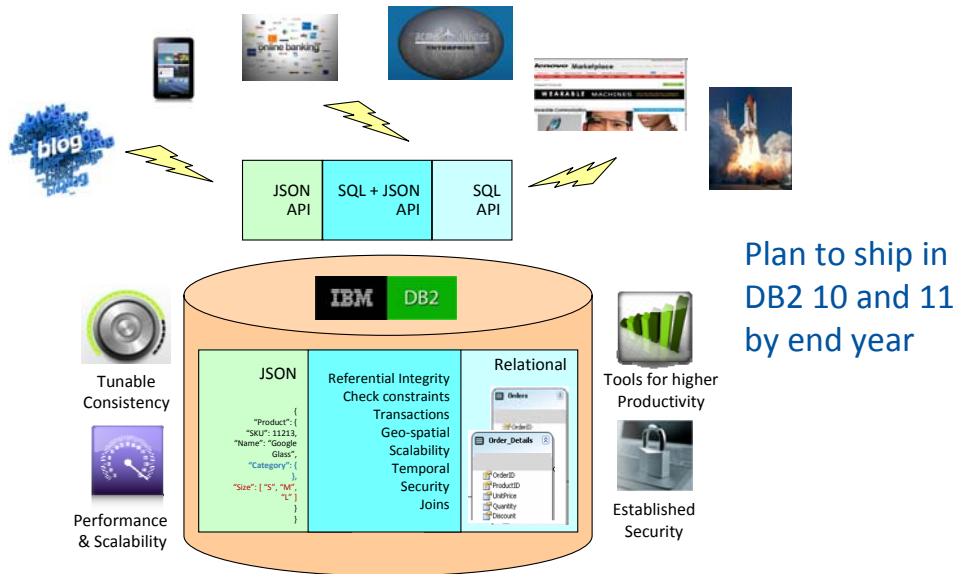
DB2 11 will provide a table UDF (HDFS\_READ) to read the Bigdata analytic result from HDFS so that it can be used in an SQL query. Since the shape of HDFS\_READ's output table varies, we will also support a generic table UDF which improves the usability of HDFS\_READ.

There would be a need to write a different external user-defined table function for reading each different Hadoop files which produce different output schema. DB2 11 will implement a new kind of user-defined table functions which are called generic table UDFs. Its output schema is determined at query compile-time. Therefore generic table UDFs are polymorphic, it increases reusability as the same table function can be used to read different Hadoop files and produce different output tables.

=====



## JSON Database Technology Preview Providing the best of both worlds



56

- customers everywhere are build systems of engagement using mobile technology
- these new systems are extremely agile, and typically use Javascript and JSON
- the app developers are pushing for JSON datastores that they see in the NoSQL world
- DB2 is evolving to be a world-class JSON datastore, so you don't need to leave DB2 to get
- DB2 LUW will start beta in June, and expects to be GA by Sept for this JSON datastore
- z is expecting to ship an APAR by 3Q13 that will make the JSON API a GA offering.

All this is about OLTP apps written in Javascript or Java.





## New Application Features

- Global variables
  - Named memory variables that you can access and modify through SQL
  - Share relational data between SQL statements
    - Without the need for application logic to support the data transfer
- SQLPL improvements (performance, manageability, function)
  - Autonomous transactions
  - Array data type support
- Alias support for Sequence objects
  - Private alias, as currently supported for tables/views
  - Or new public alias support, enabled only for sequence objects
    - Implicit SYSPUBLIC qualifier
- Row/Column Access Control UNION/UNION ALL support
- Unicode column support for an EBCDIC table (planned)
- Support for mixed case package names (planned)
- Provide REST UDFs as DB2 samples

57

DCR DN1703 SYSIBMADM new schema for built-in global variable. add SYSIBMADM to the SQL PATH as support for the administrative built-in objects (like global variables) being introduced. This change will affect the initial value of the CURRENT PATH special register, and the implicit schemas in the SQL path, such as under the SET PATH statement, and the PATH bind option (for both the bind commands and SQL bind options for routines - i.e., CREATE/ALTER statement descriptions).

N100 – ARRAY datatype support

N102/8445 alias support for sequence objects. Private alias: The qualifier must be specified explicitly or implicitly via current schema. Public: implicit qualifier is SYSPUBLIC.

N102/8446 public alias support for sequences

N15701/15704 Provide REST UDFs as sample in DB2

Column level Unicode: allow a column in an EBCDIC table to be defined for string data encoded in UTF8 or UTF16. Extensions are introduced to

allow a column in an EBCDIC table to be defined as: UTF8:

VARCHAR(n) CCSID 1208, UTF16: VARGRAPHIC(n) CCSID 1200



## XML Enhancements

- New Features
  - Basic xQuery (retrofit to v10, PM47617, PM47618)
  - COBOL samples for XML (published on Developerworks website)
- Enhanced Features
  - Implicitly add doc node during insert/update
  - Crossloader support
  - Fix error reporting position predicate
  - Support xquery constructor as the source expression of insert and replace
- Performance Enhancements
  - Binary XML validation (retrofit to DB2 V10)
  - Partial validation after update
  - Date/Time Predicate Pushdown
  - XQuery(FLWOR) and XMLQUERY enhancement
  - Optimize Index Search Keys
  - XML Operator Improvements, use less storage and CPU
  - XQuery deferred construction
  - XMLTABLE pushdown cast
  - Avoid validation of validated binary XML data during LOAD

58

For xquery support, v11 added document node, comment node and PI node constructor. The main improvement in XML area in v11 is performance: xmltable, xquery and binary XML validation performance are improved.

XML Cobol samples are published in XML developer website ([https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en\\_US#/wiki/pureXML/page/DB2%20for%20zOS%20pureXML](https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en_US#/wiki/pureXML/page/DB2%20for%20zOS%20pureXML)).

Implicitly add doc node: Customers today have to manually add the XMLDOCUMENT function for insert/update. Customers can miss doing that and then are confused by the error message they get.

Fix error reporting: Today when cast fails we report the error right away and stop processing, but it is possible that the data in error will be eliminated by a predicate, so we want to record the error and only report it as an error, if needed, when we output the data, thus eliminate unnecessary error reporting and allow the processing to continue.

Binary XML validation: When a user sends over binary XML for schema validation, we need to serialize the binary XML into textual XML before executing schema validation. This process does not perform well.

Partial validation after update: only the updated part need to be validated, not the entire document.

4705: More efficient FLWOR expression in XQuery provides more flexibility than XPath expression. However, for semantic equivalent FLWOR expression and XPath expression, the FLWOR expression requires more document scans at runtime than the XPath expression. Thus, the performance of the FLWOR expression is worse than that of the equivalent XPath expression.

4791: *More efficient XML index with predicate pushdown.* The problem was: predicates become less restrictive after being pushed down from SQL to XML due to the lack of infrastructure for index key boundaries for data type DATE, TIMESTAMP and VARCHAR **improved unbounded index scan for '=' predicates against VARCHAR and TIMESTAMP indexes**

7281: *XPath evaluation use less storage and CPU.* In the current XPath evaluation algorithm, multiple local tokens with different life-spans co-exist. The storage allocation may not be optimal. Also, when multiple items qualify the search, each of them is created as a singleton sequence and then the sequence header is discarded during appending for all sequences except for the first one

7282: *XML operator use less storage and CPU.* Some XML operators don't need to use local storage token but were allocated with one

9078: *Improved performance in most case when I UNLOAD validated binary XML data from a XML type modifier and then LOAD the data back to a XML type modifier column with the same XML schema*

Story 9078: Avoid revalidation on validated binary XML (XDBX) data



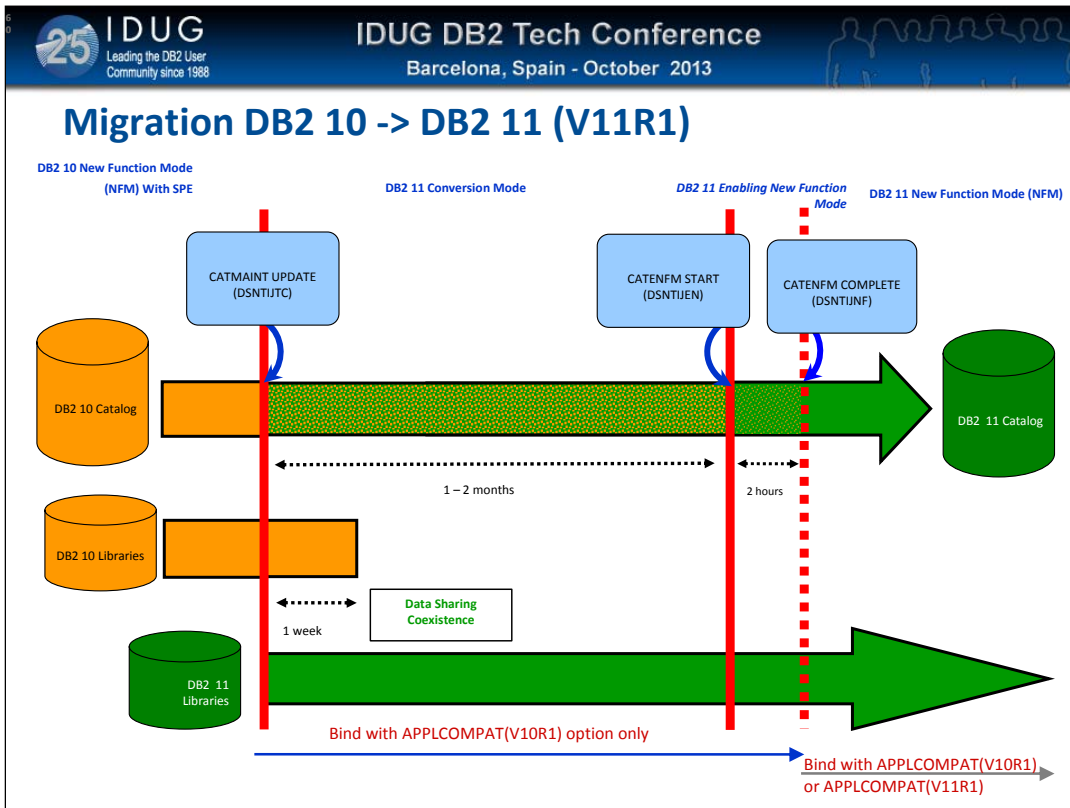
## Easier DB2 Version Upgrade – application SQL compatibility

- New DB2 releases can introduce SQL behavior changes which can break existing applications
  - For example, changes for SQL standards compliance
  - Example: DB2 10 CHAR function with decimal input no longer returns leading zeros when there is a decimal point
- Application SQL Compatibility – DB2 11 provides a new option for enforcement
  - Provide mechanism to identify applications affected by SQL changes
  - Provide seamless mechanism to make changes at an application (package) level or at a system level
    - This mechanism will enable support for up to two back level releases (N-2)
    - The release after DB2 10 will be the initial deployment of this capability
    - DB2 10 will be the lowest level of compatibility supported

59

V10: ZParm BIF\_COMPATIBILITY to revert to the way it worked in V9. Can also set SYSCOMPAT\_V9 to beginning of PATH Bind option or in CURRENT PATH. IFCID 366 Introduced to report when CHAR with decimal used

N8195 – SQL Compatibility. ZPARM for Default BIND Option. BIND/REBIND options for Packages. Special Register for Dynamic SQL (CURRENT APPLICATION COMPATIBILITY). Warnings provided when program uses incompatible SQL. IFCID 366 will report on Packages affected in both modes and Dynamic SQL. IFCID 376 is a roll up of 366, one record written for each unique static or dynamic statement. Migration job DSNTIJPM will also warn of static SQL packages affected before SQL used. APPLCOMPAT(VnnR1) – nn is the DB2 Version Number. V10R1 is the lowest release of DB2 catered for. V10R1 Only Allowed in CM. V11R1 or V10R1 Allowed in NFM. APPLCOMPAT(V10R1) assumed for all static SQL packages bound prior to V10



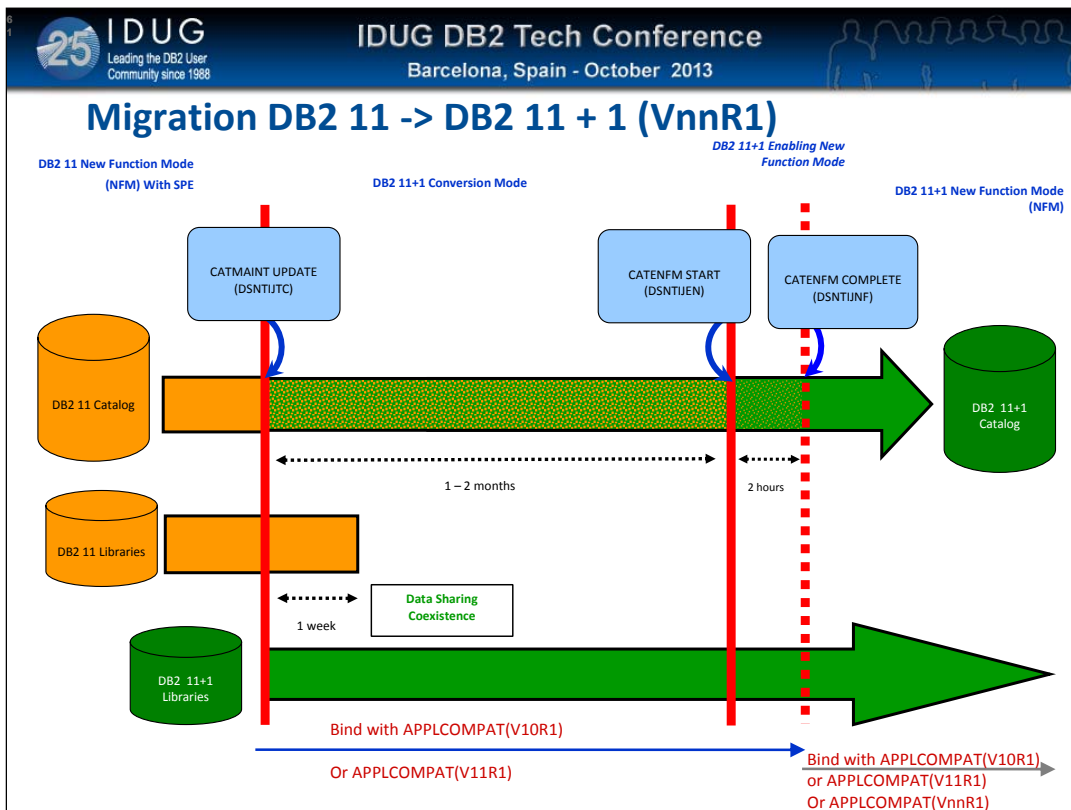
This diagram shows the migration process, what is highlighted at the bottom of the slide is that if any BINDS or REBINDS take place in Conversion mode then they can only use APPLCOMPAT(V10R1) as no new SQL functions for DB2 11 can be used in conversion mode.

APPLCOMPAT(V10R1) is assumed in conversion mode if not specified.

But when in new function APPLCOMPAT(V10R1) can be used to allow SQL that valid in DB2 10 but is not valid in DB2 11 or APPLCOMPAT(V11R1) so that new SQL functionality can be used.

APPLCOMPAT(V10R1) on BIND REPLACE in V11 NFM stops use of new SQL functionality introduced in V11 NFM.

Use of APPLCOMPAT(V11R1) is required in DB2 11 NFM on BIND ADD|REPLACE in order to use new SQL functionality as introduced in 11 NFM.



Vss is the version number for DB2 11 and VnnR1 is the version number for the next version of DB2 after DB2 11.

This diagram shows the migration process, what is highlighted at the bottom of the slide is that if any BINDS or REBINDS take place in Conversion mode then they can use APPLCOMPAT(V10) or APPLCOMPAT(V11R1) as no new SQL functions for DB2 11 can be used in conversion mode.

APPLCOMPAT(V10) will allow SQL that may not be valid in DB2 11 or in the version of DB2 after DB2 11 but was valid in DB2 10.

But when in new function APPLCOMPAT(V10R1), APPLCOMPAT(V11R1) or APPLCOMPAT(VnnR1) can be used

APPLCOMPAT(V10R1) - to allow SQL that valid in V10 but is not valid in DB2 11 or DB2 Sequioa + 1.

APPLCOMPAT(V11R1) - to allow SQL that was valid in DB2 11 but not in DB2 11 + 1.

APPLCOMPAT(VnnR1) - to allow SQL that is only valid in DB2 Sequioa + 1.



## 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
- Sysplex query parallelism support is removed

62

DB2 11 will work with (tolerate) any version of DB2 Connect that is in service - so 9.5 (with latest fixpack). We highly recommend 9.7 FP6+ or higher.

Our main testing focus is on 9.7 FP7 and 10.1 FP2 . Kepler (10.5 unannounced) will be the version of DB2 Connect that supports the following Sequoia features (FP1 GA):

- .Array data type available in NFM ,
- .Data sharing support for global variables available in NFM,
- .Performance improvements for procedures available in CM,
- .Client info enhancements available in CM

Kepler will not GA in time for the beta. If a beta customer wants to beta both Kepler and DB2 11, they can do that, but we are not pushing them to sign up for that.

All DB2 Connect releases in service at time of GA will support Db2 11. A certain FP level will be required for new functions.

V3.64 will be included in the ESP build (as a separate FMID).

IBM Data Server Driver for JDBC and SQLJ 3.64/4.15 will support some DB2 11 functions (this will release next month)

IBM Data Server Driver for JDBC and SQLJ 3.65/4.15 will support DB2 11 when DB2 11 GA's (if not earlier)



## Summary

- Performance Improvements
  - Improving efficiency, reducing costs, no application changes
  - 0-5% for OLTP, 5-15% for update intensive batch
  - 5-30% for query workloads
  - Less overhead for data de-compression
  - Exploitation of 2G real storage frames on zEC12 hardware
- Continuous Availability Features
  - Improved autonomies which reduces costs and improves availability
  - Making 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
  - Faster, more efficient performance for query workloads
  - Temporal and SQLPL enhancements
  - Transparent archiving
  - SQL improvements and IDAA enhancements
- Simpler, faster DB2 version upgrades
  - Divorce application changes for compatibility from from DB2 system upgrade
  - Access path stability improvements
  - Product stability: support pre GA customer production

63

DB2 11 will work with (tolerate) any version of DB2 Connect that is in service - so 9.5 (with latest fixpack). We highly recommend 9.7 FP6+ or higher.

Our main testing focus is on 9.7 FP7 and 10.1 FP2 . Kepler (10.5 unannounced) will be the version of DB2 Connect that supports the following Sequoia features (FP1 GA):

- .Array data type available in NFM ,
- .Data sharing support for global variables available in NFM,
- .Performance improvements for procedures available in CM,
- .Client info enhancements available in CM

Kepler will not GA in time for the beta. If a beta customer wants to beta both Kepler and DB2 11, they can do that, but we are not pushing them to sign up for that.

All DB2 Connect releases in service at time of GA will support Db2 11. A certain FP level will be required for new functions.

V3.64 will be included in the ESP build (as a separate FMID).

IBM Data Server Driver for JDBC and SQLJ 3.64/4.15 will support some DB2 11 functions (this will release next month)

IBM Data Server Driver for JDBC and SQLJ 3.65/4.15 will support DB2 11 when DB2 11 GA's (if not earlier)

**25** IDUG  
Leading the DB2 User  
Community since 1988

IDUG DB2 Tech Conference  
Barcelona, Spain - October 2013

# Questions



64





**THANK  
YOU**

**John J. Campbell**

Distinguished Engineer, DB2 for z/OS Development

[campbelj@uk.ibm.com](mailto:campbelj@uk.ibm.com)

**Joachim Limburg**

Director, DB2 for z/OS Development

[jlimb@us.ibm.com](mailto:jlimb@us.ibm.com)

Session A01/B01 and A02/B02

DB2 11 for z/OS Technical Overview

