Highlights of DB2 12 for z/OS with the Fast Forward Button Activate

Bart Steegmans
DB2 for z/OS L2 Performance
2th June 2016 – GSE Belgium
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.

- Many thanks to John Campbell for allowing me to use his material.
Agenda

• Introduction
• Performance focus – traditional workloads
• Performance focus – enabling modern applications
• Application enablement
• Reliability, availability, scalability, security
• Migration
**Goals**

**Application Enablement**
- IDAA improvements to expand to new use cases
- SQL/SQL PL improvements for next wave of applications
- Relieve table scalability limits
- Simplify large table management
- 5-10% CPU reduction with use of in-memory features
- 2x increase in throughput for non-clustered insert
- 20-30% CPU reduction for query workloads
- Improve efficiency by reducing other resource consumption

**DBA Function**
- Address key customer requirements to expand use of existing features
- Mobile, hybrid cloud, and DevOps enablement
- Remove biggest 24x7 inhibitors
- Security and compliance improvements
- Remove system scaling bottlenecks for high n-way systems
- Serviceability, availability

**OLTP Performance**
- 80% UNION ALL performance improvement
- Simplify access path management
Quick Hits

• **Scale and speed for the next era of mobile applications**
  – Over 1 Million Inserts per second measured, will scale higher
  – 256 trillion rows in a single table, with agile partition technology

• **In Memory database**
  – Up to 23% CPU reduction for index lookup with advanced in-memory techniques

• **Next Generation application support**
  – 360 million transactions per hour through RESTful web API

• **Deliver analytical insights faster**
  – Up to 2x speedup for query workloads, 100x for targeted queries
Performance Focus – Traditional Workloads
Performance Enhancements

• **In-memory contiguous buffer pools**
  - Direct row access in-memory, greatly reduced GetPage overhead
  - Up to 8% CPU reduction measured for OLTP
  - PGSTEAL(NONE) – improved in DB2 12 to avoid LRU and hash chain management overheads
  - Overflow area is used in case objects do not fit
    * Automatically managed by DB2
    * Allocated when buffer pool is allocated, but only backed when used

• **In-memory index for fast traversal**

• **More granular Global Commit LSN and Read LSN**
  - Potential huge improvement in lock avoidance (data sharing)
  - Help space reuse for insert
In-Memory Index Optimization

• A new Index Fast Traverse Block (FTB) is introduced
  — Memory optimized structure for fast index lookups
    • Contains index non-leaf page information
  — Resides in memory areas outside of the buffer pool
    • New zparm INDEX_MEMORY_CONTROL
      • Default=AUTO (min of 500 MB or 20% of allocated buffer pool storage)
  — UNIQUE indexes only, key size 64 bytes or less

• DB2 automatically determines which indexes would benefit from FTB

• DISPLAY STATS command shows which indexes are using FTBs

• New SYSINDEXCONTROL catalog table
  — Specify time windows to control use of FTBs for an index

• New IFCIDs 389 and 477 to track FTB usage
Simple Index Look-up: Faster & Cheaper

- Up to 23% CPU reduction for index look up using DB2 12 In-memory index tree
Performance Enhancements ...

• **Piecemeal list prefetch (still under test)**
  - Improved query performance for unclustered data, ISO(UR)
  - REORG avoidance feature

• **Avoid scheduling unnecessary prefetch**
  - Problem: when all pages are in memory, dynamic prefetch needlessly schedules prefetch
  - Wastes CPU, can cause “out of prefetch engine” condition
  - Attempts to solve this in the past failed - solved in DB2 12
  - Up to 6.8% CPU reduction for OLTP, 4.5% for query
Performance Enhancements …

• INSERT performance improvements

• RLF control for static packages

• SYSPACKAGE.LASTUSED
  — No longer ‘reset’ during BIND REPLACE
  — Becomes useful for determining when packages can be freed

• DB2 / DASD synergy enhancements – retrofit to V10, V11
  — Exploit z/OS Hyperwrite (PPRC log write accelerator)
    • Up to 30% log write latency reduction
  — Improved Integration with DS8870 Easy Tier multi-temperature management
    • Avoid I/O degradation after REORG
INSERT Performance

• Insert workloads are amongst the most prevalent and performance critical

• DB2 12 delivers significant improvements for non-clustered insert: journal table pattern
  — UTS with MEMBER CLUSTER (both with/without APPEND)

• Advanced new insert algorithm to streamline space search
  — Default is to use the new fast algorithm for qualifying table spaces
    • INSERT ALGORITHM zparm can change the default
    • INSERT ALGORITHM table space attribute can override zparm
Faster Ingestion - Scale and Speed – Special Case

- 3x DB2 response time improvement with 20% CPU Reduction using new DB2 12

PBR MC/APPEND in 2 way data sharing

> 1 Million inserts/sec
Faster Ingestion - Scale and Speed – Journal Scenario

![Diagram showing CPU and Throughput from PBR/RLL/MC/SEQ]
Performance Enhancements ...

- Implement new PL/X compiler

- **Buffer Pool advisory mode to simulate larger buffer pools – rollback to v11**
  - Low CPU and real memory overhead
  - Statistics provided to indicate I/O savings
  - Retrofit to V11 (PI22091)

- **[Increase use of 1M size pageable large page frames for DB2 working storage]**
  - 4.7% CPU decrease measured (no paging)
  - zparm controlled – if DB2 is paging, then there is overhead for 1M paging
  - Flash Express improves 1M paging
  - **Still under evaluation**

- **Streamlined Claim/Declaim processing**
  - Avoid re-claim overhead across multiple commit scopes serially reusing persistent thread running RELEASE(DEALLOCATE)
  - Online REORG and other drainers can break in
Performance Enhancements ...

• Improvements in pool management to simplify and remove the scalability inhibitors
  – EDM pools
  – LOB/XML storage

• Other Improvements
  – Reducing DGTT DECLARE overhead
  – Remove log force write from identity column and sequence caching in data sharing
System Scaling Enhancements

• **Large n-way scaling**
  – Improved efficiency on LPARs with high number of CPs
  – Log latch contention reduction: testing shows up to 41% CPU reduction and 6% throughput improvement for high contention cases
  – Buffer Pool scaling improvements:
    • LC23 reduction, PLO avoidance
    • 5-30% CPU improvement when accessing hot pages
  – IRLM latch contention reduction
  – EDM DBD and skeleton pool scalability improvements

• **Optimizations for new hardware**
  – Exploitation of z13 decompression enhancement
  – Internal structure changes for cache efficiency, more processor prefetch

• **Raise total buffer pool size limit to 16 TB**

• **>4G size active log datasets**
High level performance expectation

• System and OLTP performance
  — 2-3% CPU reduction without Index In-Memory feature
  — 5-10% CPU reduction by exploiting Index In-Memory feature
  — Further reduction is possible with contiguous buffer pools, and / or persistent thread with RELEASE(DEALLOCATE)

• Query performance
  — Wide range of improvement
    • Typically 0-20% without new access path
    • Typically 10-40% with new access path
    • Up to 90% reduction is observed in our evaluations

• Concurrent insert against UTS and MEMBER CLUSTER
  — 5-10% CPU reduction
  — Throughput improvement if current bottleneck is space search or page contentions
Instrumentation Enhancements

• More granular wait times for IFCIDs 316 (dynamic) and 401 (static)
  — Accumulated wait time due to global contention for locks (broken out by type)

• Enhance IFCIDs 53/58 statement level section for PREPARE
  — Similar to INSERT/UPDATE/DELETE

• Enhance SQL performance tracing adding RDI Section Number in IFCIDs 53/58

• Add batch job STEP name in correlation header

• Add REFRESH TABLE to counts in DSNDQXST

• Add workfile, tempfile usage information to Accounting trace

• Enhanced precision for IFCID 199 dataset I/O stats - microseconds
Access path (plan) stability

- Dynamic SQL plan stability
- Static plan stability usability
- Preserve dynamic statement cache at rollback
- Integrated RUNSTATS with optimizer
  - Automated update of statistics profiles by optimizer
- Statistics profile support
  - Automated update for Index DDL
  - Inline stats usage of profiles
- Simplify creation of all tables required by Explain
  - New ADMIN_EXPLAIN_MAINT stored procedure
Dynamic SQL Plan Stability

• **Problem:**
  - Unstable performance of repeating dynamic SQL statements
  - Environmental changes can result in change in access path or performance regression, and this can be tough to manage
    - RUNSTATS
    - Applying software maintenance
    - DB2 release migration
    - zparm changes
    - Schema changes

• **Static SQL has several advantages**
  - Access path established at BIND time
  - Static plan management gives advanced management functions

• **Objective: extend static SQL advantages to dynamic SQL**
Dynamic SQL Plan Stability ...

• **Base infrastructure**
  - Opaque parameter CACHEDYN_STABILIZATION
  - Capture
    * Command with / without monitoring
    * Global variable
  - FREE
  - EXPLAIN (current, invalid)
  - Invalidation
  - LASTUSED (identify stale statements)
  - Instrumentation (query hash, explain, cache + catalog hit ratio)
  - APPLCOMPAT is part of matching criteria
Dynamic SQL Plan Stability ...

**Key limitations**

- Literal concentration and temporal stabilization not currently included
- REBIND support not included
  - No PLANMGMT/SWITCH/APREUSE
Static Plan Stability: Usability

• BIND REPLACE of same version to keep LASTUSED

• FREE PACKAGE improvements
  — Selectively FREE either original or previous
  — Choose to FREE if invalid only
  — Allow FREE of inactive package copies while application is running

• REBIND PACKAGE improvements
  — SWITCH to include APREUSESOURCE option so as to choose ORIGINAL or PREVIOUS copy as source for APREUSE
  — Addresses issue of switch accidentally causing invalid copy to become current
  — APREUSESOURCE option avoids the two step process of
    • REBIND SWITCH followed by
    • REBIND APREUSE
RUNSTATS Enhancements for SQL Performance

• **RUNSTATS** *(for access path selection benefit)*
  
  — CLUSTERRATIO formula improvements
  
  — Statement cache invalidation
    
    • Optional for RUNSTATS (new default)
    
    • For other utilities ONLY if objects in pending state before utility executed
  
  — Profile support for inline statistics
  
  — Automated COUNT for FREQVAL
    
    • Allow DB2 to collect the skewed values
      
      — Up to top 100 or until no skew for remaining values
  
  — Optimizer to automatically update statistics PROFILE with RUNSTATS recommendations
    
    • DSNACCOX to recommend RUNSTATS after profile update
      
      — Specify USE PROFILE on RUNSTATS to collect current statistics recommendations
Enhanced statistics profile management

Inline statistics & RUNSTATS

Statistics in Catalog

Table structure in Catalog

Profiles in Catalog

Create/Drop Index

Optimizer

Query

Access Path

All new with V12
DB2 for z/OS – Ultimate Database for Cloud, Analytics and Mobile
Industry-leading performance, security, scale and reliability

Performance Focus – Enabling new applications
Query Performance Enhancements

• Up to 25% CPU improvement for traditional query workloads

• Up to 2x improvement for modern SQL applications
  – Performance improvements for next generation SAP applications
  – Real-time analytics, complex OLTP workloads

• 100% zIIP eligibility for parallel query child tasks

• Modern applications contain more complex SQL patterns (targeted in DB2 12), and more sorting, joins, stage 2 predicates etc.
  – These complex patterns are less common in traditional OLTP/batch
Query Workload CPU and Elapsed time Improvement from V11 (%)

- WAS Portal
- SAP SFIORI
- SAP SFIN
- Customer 2
- BIDAY-short
- TPCH 30GB NPI
- Crystal Reports
- TPCH-SQLPL
- Customer 1
- Customer 3
- TPCD
- SAP/Bw
- BIDAY-long

**Diff (%)**

- UNION ALL w/View
- Complex Outer Join, UDF
- Simple query or large data scan
- Complex reporting, large sort

CPU

Elapsed
High-level Performance Focus

• **Query (RDS) focus based upon new(er) workloads**
  - Complex views or table UDFs
    - UNION ALL
    - Outer joins
    - Join predicates with (Stage 2) expressions
  - CASE expressions, CAST functions, scalar functions

• **Query - General Bottlenecks**
  - Sort/workfile reductions
  - Reducing prepare cost and frequency
  - I/O performance
    - Reduce unnecessary prefetch scheduling
Query Performance Focus

• Improve performance of
  ─ UNION ALL and outer join performance enhancements
    • Performance issues are similar with both types of query pattern
    • Reduce materializations
      ─ Bypass workfile usage when materialization required
      ─ Trim unnecessary columns from materializations
  • Push predicates inside UNION ALL legs or OUTER JOIN query blocks
  • Push ORDER BY and FETCH FIRST into UNION ALL legs
  • Reorder OUTER JOIN tables to avoid materializations

── Table UDFs
  • Improve merge similar to views
  • Indexability of join predicates
Query Performance Focus ...

**Improve performance of ...**

- Join predicates with Stage 2 expressions
- Stage 2 to indexable
  - VARBINARY, COLLATION_KEY IOE
- Expression evaluation
  - CASE, SUBSTR, etc
  - Expression sharing (SELECT list only)
  - Caching deterministic UDF results
- Enabling parallelism
  - 100% zIIP offload for parallel child tasks
  - Reduce cost and resource consumption
- Sort
  - Reduce workfile usage for GROUP BY/DISTINCT
  - Reduce key length for GROUP BY/DISTINCT and sparse index
  - Continued progress towards in-memory for smaller sorts (begun in V9)
Query Performance Focus ...

- **Adaptive index**
  - Allow RID based plans (single index list PF or multi-index) to quickly determine filtering from index
    - Without requiring REOPT(ALWAYS)
    - For list prefetch or multi-index ORing
      - Earlier opportunity to fallback to tablespace scan if large % of table to be read
    - For multi-index ANDing
      - Reorder index legs from most to least filtering
      - Early-out for non-filtering legs, and fallback to rscan if no filtering
  - Optimizer to use uncertainty to determine risk of a single index plan
    - Quick evaluation done based upon literals used
    - Any further evaluation of filtering deferred until after 1 RID block retrieved
      - Ensuring that very short running queries do not incur overhead
Query Performance Focus ...

• Adaptive index ...
  
  — Simple example of targeted use case
  
  ```sql
  SELECT * FROM TAB1 WHERE COL1 < ? AND COL2 < ? AND COL3 < ?;
  
  INDEXES: IX1 (col1), IX2 (col2), IX3 (col3)
  ```

  ```sql
  SELECT * FROM CUSTOMER
  
  WHERE LASTNAME LIKE ?
  
  AND FIRSTNAME LIKE ?
  
  AND ADDRESS LIKE ?
  
  AND CITY LIKE ?
  
  AND ZIPCODE BETWEEN ? AND ?
  ```

  — Filtering of the above query is dependent on literals at execution

  • Common pattern for search screens (with BETWEENs or LIKEs)
  
  • Common that 1 index is filtering unless a highly skewed value is searched
DB2 for z/OS – Ultimate Database for Cloud, Analytics and Mobile
Industry-leading performance, security, scale and reliability

Application Enablement
Many modern application developers work with REST services and JSON data formats.

DB2 Adaptor for z/OS Connect provides the means to do this.

Beta in progress, planned for DB2 10 and 11 as well as DB2 12.


```
{"customer":
{"ID": "32537",
"Name": "Bob Rady Enterprises"}
}
```
Getting REST and JSON into your mainframe environment in a way that enables you to best take advantage of the assets that exist there:

- Liberty Profile Server
  - Provides a common and consistent entry point for mobile access to one or many backend systems
  - Simplifies front-end functions by allowing them to pass RESTful and JSON rather than be aware of or involved in data transformation

IBM BlueMix

Cloud/Mobile integration with DB2 and z/OS Connect

Cloud APIs

Mobile-Optimized APIs

Enterprise Systems Integration

Batch, WAS

Enterprise Apps

Enterprise Data

Enterprise Systems Integration

On-Premise Enterprise APIs

Cloud-based Services

IBM BlueMix

CICS, IMS

Mobile-Optimized APIs

DB2

On-Premise Enterprise APIs

Services

App Development

App Execution

IMS

CICS, IMS

Enterprise Transaction Processing

Systems of Record
Application Related Enhancements

• DRDA Fast Load – callable command for fast streaming of data into DB2
• System profiling enhancements
  – Auto start of system profiles
• Set global variables based on profiles (e.g. transparent archiving)
• New MODIFY DDF PKGREL(BNDPOOL) option
  – Pool high performance DBATs at connection termination
• DSNULI support for IMS Attach
• Long list of SQL and XML improvements
DRDA Fast Load

• Problem:
  — DB2 provides the DSNUTILU stored procedure to load data from a client
  — But this is difficult to use, application must transfer data to z/OS file

• Solution:
  — DB2 Client API (CLI and CLP) for remote load into DB2
  — Easy and fast loading of data from file that resides on client
  — Internal format (SAP), as well as delimited and spanned (LOB data)
  — Applicable for some “cloud” use cases
  — Overlap network operations with data ingest on the DB2 server
  — Measured results show as fast or faster than DB2 LOAD utility

• Plans to use this feature for fast write from Spark
SQL Enhancements

• Simple VALUES using dynamic SQL – V10, V11
• JDBC/ODBC Type 2 performance Enhancements
• ODBC driver improvements:
  — TIMESTAMP with TIMEZONE support
  — Multi context support using ASSOCIATE/DISSOCIATE THREAD
  — Ability to preserve dynamic statement cache after rollback
• SQL Pagination syntax support – OFFSET n and LIMIT
• Support prepareAttribute literal replacement as BIND option
SQL Enhancements ...

- Keep views, UDFs intact on DDL for underlying tables
- Increase maximum number of tables referenced in view, UDF, statement
- Enhanced MERGE support
- SQL pagination improvements
- New built in functions
  - HEX2BIN (V11)
  - MEDIAN, PERCENTILE_CONT, PERCENTILE_DISC
  - [ COUNT(DISTINCT x) - under consideration ]
  - GENERATE_UNIQUE – new optional length parm
  - HASH functions (CRC32, MD5, SHA1, SHA256)
Enhanced Merge Support

• DB2 z/OS initial support for MERGE statement with limited functionality was delivered with Version 9:
  — Limited to UPDATE and INSERT and only one of each
  — Focused on use of host variable column arrays to provide multiple rows of input data

• In DB2 12, DB2 z/OS MERGE statement will be aligned with behavior defined in SQL Standard and DB2 family
  — Source data as a table-reference
  — Multiple MATCHED clauses
  — Additional Predicates with [NOT]MATCHED
  — Support DELETE operation
  — Allow IGNORE and SIGNAL
SQL Pagination

• With the growth of web and mobile applications, application developers are looking for more efficient ways to develop good performing applications

• Numeric-based pagination

  ─ SELECT * FROM tab OFFSET 10 ROWS FETCH FIRST 10 ROWS ONLY

• Data-dependent pagination

  ─ Existing syntax:

    WHERE (LASTNAME = ‘SMITH’ AND FIRSTNAME >= ‘JOHN’) OR (LASTNAME > ‘SMITH’)

  ─ New equivalent syntax

    WHERE (LASTNAME, FIRSTNAME) > (SMITH, JOHN)
SQL Enhancements ...

- Unicode columns in EBCDIC tables
- Piecemeal DELETE – allow for interim commits
  - e.g. Delete from T1 where C1 > 7 “FETCH FIRST 5000 ROWS ONLY”
  - AUTONOMOUS SQL PL procedure is an existing tactical solution
- MQ UDF enhancements: allow MQ message header to be passed
- BiTemporal enhancements
  - Auditing enhancements – V11
  - Inclusive/inclusive for business time
  - Logical transaction for system time
  - Temporal RI
Piece-wise Modification of Data

- Mitigate the effects of locking and logging when potentially millions of rows could be affected by a simple statement like: "DELETE FROM T1 WHERE C1 > 7"

- Solution
  - Support fullselect as the target of a DELETE statements where the fullselect allows for the FETCH FIRST N ROWS ONLY to be specified.
  
  - DELETE FROM (SELECT * FROM T1 WHERE C1 > 7 FETCH FIRST 5000 ROWS ONLY)
SQL Enhancements ...

• APPLCOMPAT support for DB2 12
• Global variable enhancements
  – Array Global Variables
  – LOB Global Variable
  – FETCH statement and global variables as a target
  – EXECUTE statement and global variables
  – OPEN statement and Global Variables
• SQL PL
  – SQL PL in triggers, including trigger versioning and debugging support
  – Support for constants
  – SQL PL obfuscation
  – Dynamic SQL in SQL PL UDFs and stored procedures
  – DBMS_OUTPUT for UDF tracing – V11
DB2 12 SQL PL Triggers

• Rich capability in trigger body
  — Allow SQL PL control statements
    • IF-THEN-ELSE, LOOP, REPEAT, ...
  — Allow more SQL statements, dynamic SQL, variables, handlers
  — Easier porting of triggers from other DBMSes
  — DEBUGGER support
  — VERSIONs support
    • Provides a better way to change a trigger without DROP, CREATE
    • Can change trigger online and maintain trigger activation order

• Richer capability means some performance overhead compared to an equivalent traditional trigger
**XML Improvements**

- Improve the performance of the XML queries by choosing optimal access:
  
  *76% / 77% reduction in the Class 1 / Class 2 time respectively.*

- Improve the performance of XMLTable that conducts the pivot-like operation to XML data with name-value pair pattern:
  
  *90% CL2 CPU improvement using customer's xml data/queries*

- Improve performance and developer productivity by allowing multiple update inside XMLModify function:
  
  *Up to 90% improvement in ET; up to 97% improvement in CPU time (when comparing to semantics equivalent single update)*

- XSLTRANSFORM function for XML extender users

  Allow for easy migration to pureXML
DB2 for z/OS existing support for JSON

• **Store data from web/mobile applications in its native form**
  – Many web applications use JSON for storing and exchanging information
  – JSON is often the preferred data format for mobile application back-ends

• **Move from development to production in no time**
  – Ability to create and deploy flexible JSON schemas
  – Gives power to application developers by reducing dependency on IT
    • No need to pre-determine schemas and create/modify tables
    – Ideal for agile, rapid development and continuous integration

• **DB2 provides two ways for working with JSON**
  – Java driver for JSON API
  – SQL extensions – DB2 11, with enhancements in DB2 12
DB2 for z/OS JSON Document Store


• Java Driver for JSON API
  – Java Driver supporting JSON API
  – Transactions
  – Parametric SQL statements (Delete, Select)

• JSON Wire Listener
  – Leverage community drivers

• CLP-Like Command Shell
  – Ad-hoc updates / queries
  – Administration commands

• DB2 Data Server
  – JSON documents stored as BSON (Binary JSON) in BLOBs within DB2
  – Scalar function and UDF extensions
  – Enhanced indexing on expression capabilities allows indexing of JSON members
JSON Enhancement

To extract and retrieve JSON data into SQL data types from BSON

Example (before):

```sql
JSON_VAL(column1, 'PO.customer.@cid', 'i:na')
```

In DB2 12, we remove the requirement that 1st parameter has to be a BLOB column (already retrofit to V11 in PI39003)

In V12, we support more options as 1st parameter:
- view column
- CASE expression
- table expression with union all
- trigger transition variable
- SQL PL variable/parameter
Reliability, availability, scalability, security
DBA Productivity – DB2 12 Goals

• Relief for table scalability limits
• Simplify large table management
• Improve availability
• Agile schemas (more online schema changes)
• Security and compliance improvements
• Streamline migration process
• Utility performance, availability, usability
Lifting Partition Limits – Problem Statement

• Maximum number of partitions is dependent on DSSIZE and page size
  – e.g., If DSSIZE = 256 GB and page size = 4K then maximum number of parts is 64
• Running out of space in a partition is an application outage
• When altering DSSIZE, REORG must run on entire table space
• DSSIZE is at table space Level and not at part level
  – All parts inherit the same DSSIZE set at table space level
  – No ability to have differing partition sizes
  – Rebalance must run against multiple partitions
• Maximum table size limited to 16TB
Lifting Partition Limits – Solution

- New UTS PBR tablespace structure called ‘UTS PBR RPN’
  - Use **relative page numbers** (RPN) instead of absolute page numbers
  - Simplicity, usability, availability, scalability
  - Remove dependency between number of partitions and partition size
  - 7-byte RIDs (2 byte part number, 5 byte page number)
    - Up to 256 trillion rows in a single table
    - New REORG mapping table format, optional until new function enabled
  - Support up to 1TB partition size
  - Maximum table size increased from 16TB (4K page size) to 4PB
    - Architected to go even larger
  - Increasing DSSIZE is supported at partition-level
  - New DSSIZE support for indexes
Lifting Partition Limits – Considerations

**Conversion / Exploitation:**
- zparm to control whether creation of range partitioned uses relative page numbering
- PAGENUM RELATIVE/ABSOLUTE option on CREATE & ALTER TABLESPACE
- Conversion is pending alter - requires tablespace-level online REORG
- Online alter to increase DSSIZE – immediate, non-disruptive
- Online alter to decrease DSSIZE - pending alter requiring table space-level REORG

**Log record formats changed to support 7 byte RIDs**
- Improved serviceability, DSN1LOGP now formats partition number explicitly
- New log record format written in “CM”, requiring fallback toleration support in V11
- Not just for RPN page sets!
Online Schema

• Insert partition

• Online deferred ALTER INDEX COMPRESS YES
  — Previously placed indexes in RBDP

• Option to defer column-level ALTERs
  — Materialise through online REORG
  — Avoid availability constraints & conflict with other deferred alters
Insert Partition

• Problem statement
  – Large range-partitioned tables often have hot spots and rebalancing across entire set of partitions is onerous
  – Partitioning scheme chosen in the past may no longer be optimal

• Solution
  – Ability to insert a new partition with specified limitkey value
    • ALTER TABLE ADD PARTITION ENDING AT xxx
  – Split existing partition, distribute data between new and old
  – Online change through pending alter
  – Only affected partition(s) need to be REORGed
  – No PIT recovery prior to materializing REORG
  – Applies only to UTS PBR (not classic partitioned TS)
Pending Column Level Alter

• **Problem statement**
  - Some column-level alters result in application impact
    • e.g. indexes placed in RBDP
  - Immediate alters conflict with pending alters
    • Additional REORGs required to materialize pending alters to avoid DDL failures

• **Solution**
  - Allow existing immediate alters to become pending alters, UTS only
  - All pending alters accumulated and materialized through online REORG
  - New zparm
    • ALWAYS IMMEDIATE: Existing behavior for existing alters
    • ALWAYS PENDING: Current immediate alters are converted to pending alters
  - ALTER COLUMN type to avoid RBDP on indexes
Security & General Enhancements

• LOB compression
  — zEDC hardware requirement

• Improved LOB handling for ISO(UR) queries to avoid SQLCODE +100

• TRANSFER OWNERSHIP

• Permit installation, migration without Install SYSADM authority to limit data access

• New UNLOAD privilege

• Support long, camel-cased DBRM names

• IFI 306 log records returned in the correct version
**Utilities**

• **REORG**

  – Improved Flashcopy management
    • REORG with only a FLASHCOPY inline image copy (no traditional ICs) now causes REORG to fail (RC=8) if the flashcopy fails

  – Improved part-level UTS PBG REORGS
    • Supports creation of new PBG partition for overflow rows during a part level reorg

  – Prevent copy-pending on LOB tablespace during REORG of UTS PBGs
    • Avoid leaving copy pending on LOB table spaces when new PBG partitions grown during log phase of REORG

  – REORG-level management of delete of UTS PBG partitions
    • Add a new keyword, DROP_PART, for REORG empty UTS PBG partition pruning when zparm alteration is not feasible

  – Support new COMPRESSRATIO catalog column
    • Add LOAD/REORG/RUNSTATS ability to gather average compress ratio at the record level instead of the page saved level
Utilities ...

• REORG ...
  — Up to 17% additional offload to zIIP
    • The reload phase can now be zIIP offloaded
  — New mapping table format support
    • The new format supports 7 bytes RIDs needed with UTS PBR RPN support
  — Permit REORG against RO page sets
    • REORG with any SHRLEVEL can be run against any RO table space and index
    • Retrofitted back to service level with APAR PI46774)
  — Display claimer information on each drain failure, not just last retry
Utilities ...

• **LOAD**
  
  — LOAD PART REPLACE with dummy input against empty PBR partition
    • NPI processing optimized to not scan the whole NPI to find keys to be deleted for the partition being REPLACEd if it was already empty
    • Reduces elapsed time and CPU time significantly if there were a lot of keys for the other logical parts of the NPI
    • Up to 99% CPU 98% Elapsed Time reduction
  
  — LOAD SHRLEVEL CHANGE PARALLEL support for UTS PBG
    • Modified to remove the single input parallelism restriction for PBG table spaces for LOAD SHRLEVEL CHANGE
    • Lab measurements saw up to 76% elapsed time savings with some CPU overhead
    • LOAD SHRLEVEL NONE PBG PARALLEL is still a restriction
  
  — Up to 90% additional zIIP offload
    • RELOAD phase of the LOAD utility can be zIIP offloaded
    • This includes the data conversion and loading of the record into the page set
Utilities ...

• LOAD ...

  — LOAD RESUME SHRLEVEL NONE BACKOUT YES to avoid RECP for page set on failure
    • New option to allow backout of rows already loaded upon encountering an error (such as conversion, LOB/XML, duplicate key, RI constraint) without leaving the page set in RECP
    • Also in DB2 10 / 11 - PI08421 + PI54658 (still open at time of writing)

  — PREFORMAT support for aux tables
    • Indicates the remaining pages are preformatted up to the high-allocated RBA in the table space and index spaces that are associated with the table that is specified in table-name
Utilities ...

• LOAD ...

  ‒ Maintain MAXASSIGNEDVAL for identity columns
    • This enhancement maintains the MAXASSIGNEDVAL for user provided input and also will reset the value if a LOAD REPLACE on the table space is run

  ‒ Eliminate data sharing overhead with UNLOAD ISO(UR) REGISTER NO option
    • New option to bypass page registration in the Coupling Facility and avoid any data sharing overhead

  ‒ LOAD REPLACE support for COMPRESSRATIO column
    • Gathers the average compress ratio at the record level instead of the page saved level
Utilities ...

• Backup & Recovery

  — PIT recovery support for UTS PBG table spaces
    • Allow recovery of PBG UTS to a PIT prior to a REORG that materialized these physical pending alters: page size (buffer pool), DSSIZE, SEGSIZE, MEMBER CLUSTER

  — Flashcopy_pprc parameter for RESTORE and RECOVER
    • Allows users to specify/override the preserve mirror option for PPRC (sync disk mirroring) on the utility statement
    • Previously: RESTORE SYSTEM used HSM default; RECOVER used zparm FLASHCOPY_PPRC

  — Option to skip PIT recovery for non-updated page sets
    • New *default* behavior for PIT RECOVER utility - objects which were not updated *after* the recovery point are skipped and not recovered
    • Reasoning: the data in these objects still reflect that PIT, so no need to recover
    • Saves CPU resource consumption and and elapsed time
    • This default behavior can be overridden with SCOPE(ALL)
Utilities ...

• Backup & Recovery ...

  – System Level Backup support for multiple copypools
    • BACKUP SYSTEM and RESTORE SYSTEM now support multiple copypools
    • Allows the user to keep daily “golden copy” or backup for critical events
  – DFSMSHsm messages included in utility job output for BACKUP/RESTORE SYSTEM
    • Improved messaging for BACKUP SYSTEM and RESTORE SYSTEM
    • HSM and DSS messages are included in the DB2 utility job output so that the user does not have to look in the (separate) HSM job logs for messages
    • Available with z/OS V2.2
  – COPY option to specify FASTREPLICATION(REQUIRED)
    • A new COPY utility zparm COPY_FASTREPLICATION REQ|PREF|NONE allows the user control of whether FlashCopy is required during creation of the FlashCopy image copy
    • With FASTREPLICATION REQUIRED, DSS will allocate target data set within same DASD box, ensuring that FlashCopy can be used
    • Previously COPY defaulted to FASTREPLICATION PREFERRED
Utilities ...

• **RUNSTATS**
  - Use PROFILE support for inline stats in REORG & LOAD
  - COLGROUP inline stats support for LOAD PARALLEL
  - INVALIDATECACHE option to avoid dynamic statement cache invalidation
    - Default behavior has changed to not invalidate prepared statements
  - COLGROUP performance – 25% CPU, 15% Elapsed Time reduction
    - When COLGROUP column specifications are redundant with INDEX columns, the duplicate COLGROUP is ignored in lieu of the index processing
Utilities ...

- **RUNSTATS** ...
  - New DSNUTILV stored procedure support CLOB input to allow >32KB
    - The DSNUTILV UTSTMT utility statement parameter is now a CLOB instead of a VARCHAR, so DB2 now supports a 4 byte length (up to 2 GB)
    - For existing applications still passing a VARCHAR, DB2 will do the conversion from VARCHAR to CLOB
  - DSNACCOX changes for REORG avoidance
    - Number of recommendations reduced by changing the default for recommending REORG based on the number of inserts and pseudo deletes since the last REORG to off
    - Removes the criteria of recommending a REORG based on REORGLASTTIME, LOADRLASTTIME, or REBUILDLASTTIME being NULL
  - Improved utility concurrency for MODIFY, COPYTOCOPY, MERGECOPY
    - Allows MODIFY RECOVERY, COPYTOCOPY, MERGECOPY, and exclusive utilities like LOAD and REORG to run concurrently on the same target objects
Data Sharing Improvements

- Support for global transactions
- DDF shared session data across group
  - DDF transaction re-routing, session token for client fail-over
- Data sharing performance improvements:
  - Improved lock avoidance checking to reduce CF lock requests
  - In-memory indexes can reduce GetPages and CF GBP requests
  - Improved insert space search can avoid P-lock contention and streamline inserts
  - RUNSTATS and UNLOAD ISOLATION(UR) to avoid CF page registration
Data Sharing Improvements ...

• New data sharing peer recovery option
• Retry of automatic LPL and GRECP recovery
• Asynchronous CF Lock duplexing (not yet enabled)
  — Reduces overhead for system managed duplexing of CF LOCK1 and SCA structures
  — Secondary structure updates are performed *asynchronously* with respect to primary updates
  — DB2 will sync up with z/OS to ensure data integrity i.e., all modify locks have been “hardened” in the secondary lock structure before the corresponding undo/redo record for the update is written to DB2 the active log on DASD
  — Increases the practical distance for multi-site sysplex operations whilst duplexing of CF LOCK1 and SCA structures
  — Planned as 4Q 2016 deliverable with
    • z/OS 2.2 APAR
    • z13 GA2 CFCC 21
Data Sharing Improvements ...

- Asynchronous CF lock structure duplexing – how it will work

1. Request in
2. Request out
3. Response
4. Response out
5. Communication
6. Ordered execution

**Diagram:**
- CF1: Primary Lock
- CF2: Secondary Lock
- XES
- IRLM and DB2
- z/OS

**Flow:**
1. Request in
2. Request out
3. Response
4. Response out
5. Communication
6. Ordered execution
DB2 for z/OS – Ultimate Database for Cloud, Analytics and Mobile
Industry-leading performance, security, scale and reliability

Migration
Migration Prerequisites – Hardware & Operating System

• Processor requirements:
  – z196, or higher, processors running z/OS V2.1, or later
  – DB2 12 will probably require increased real storage for a workload compared to DB2 11 for z/OS

• Software Requirements:
  – z/OS V2.1 Base Services, (5650-ZOS), or later
  – DFSMS V2.1, or later
  – Language Environment Base Services
  – z/OS V2.1 Security Server (RACF), or later
  – IRLM Version 2 Release 3 (Delivered with DB2 12)

• Additional details:
Migration & Catalog

• Single phase migration process
  – No ENFM phase
  – New function activated through new command
    • -ACTIVATE NEW FUNCTION
  – APPLCOMPAT rules, fallback rules continue to apply

• BSDS conversion to support 10 byte log RBA is pre-requisite

• No pre-V10 bound packages
  – Get rid of 31-bit runtime, some performance improvements

• BRF is deprecated
  – BRF page sets still supported, but zparm and REORG options are removed

• Temporal RTS tables
  – Defined in catalog, enablement is optional
Online Migration Improvements

- Pause statistics externalization during migration to reduce contention (part of fallback SPE)
  - Real time statistics
  - Stats feedback
  - SYSPACKAGE.LASTUSED

- Catalog/directory lock duration reduction APARs (impacts online migration, catalog REORGs)
  - PI43662 – Accelerator resources released in a timely manner
  - PI43916 – DB2 plan allocation locks released in timely manner
  - PI39053 – Avoid locks from SET statement
  - PI40755 – Dynamic SQL release prepare locks earlier