DB2 12 – The ultimate Database for Cloud, Analytics & Mobile

Jørn Thyssen, IBM Denmark – jthyssen@dk.ibm.com
Agenda

- DB2 12
  - High level overview of features
  - Utilities improvements
  - Tools
- Mobile
  - DB2 native REST interface
Announcing DB2 12

The #1 Enterprise Data Server

Improved business insight
Highly concurrent queries run up to 100x faster

Faster mobile support
6 million transactions per minute via RESTful API

Enterprise scalability, reliability and availability for IoT apps
11.7 million inserts per second, 256 trillion rows per table

Reduced cost
23% lower CPU cost through advanced in-memory techniques
DB2 for z/OS Timeline

<table>
<thead>
<tr>
<th>Version</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>V7</td>
<td>3/2001</td>
</tr>
<tr>
<td>V8</td>
<td>3/2004</td>
</tr>
<tr>
<td>V9</td>
<td>3/2007</td>
</tr>
<tr>
<td>V10</td>
<td>10/2010</td>
</tr>
<tr>
<td>V11</td>
<td>10/2013</td>
</tr>
<tr>
<td>V12</td>
<td>10/2016</td>
</tr>
</tbody>
</table>

DB2 12 was years in the making

DB2 12 ESP (“beta”) started in March, 2016
In-Memory Index Optimization

- **A new Index Fast Traverse Block (FTB) is introduced**
  - Memory optimized structure for fast index lookups
  - Resides in memory areas outside of the buffer pool
    - New zparm INDEX_MEMORY_CONTROL
      - Default=AUTO (min. of 500 MB or 20% of allocated BP 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 Look-up : Faster & Cheaper

Up to 23% CPU reduction for index look up using DB2 12 In-memory index tree
Insert workloads are amongst the most prevalent and performance critical.

- DB2 12 delivers significant improvements for Non-clustered insert: journal table pattern
- UTS, MEMBER CLUSTER
  - 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
Significant CPU Reduction in DB2 Query Workloads

Cpu Reduction %

- Cryst.Rpt-Long: 90%
- SAP CDS Fiori: 85%
- WAS Portal: 80%
- Cust1-Uncl: 66%
- SAP CDS FIN: 45%
- BIDAY-Short: 23.77%
- Cust2: 24.00%
- TPCH-seq: 23.00%
- TPCH-parallel: 20.00%
- Cust1-Clus: 17.00%
- Cryst.Rpt-Short: 15.70%
- Cust3: 13.18%
- TPCH-SQLPL: 11.00%
- TPCD: 8.66%
- SAP BW: 4.52%
- BIDAY-Long: 3.02%

- UNION ALL w/View
- Complex Outer Join, UDF
- Complex reporting, large sort
- Simple query or large data scan
DB2 12: Simplicity and RAS

- Dynamic SQL Plan Stability
  - Stabilize performance of repeating dynamic SQL statements

- RUNSTATS automation
  - Optimizer automatically update profile with RUNSTATS recommendations

- RLF control for static packages

- LOB compression
  - Using zEDC hardware

- DRDA Fast Load
  - Callable command for fast load of data into DB2 directly from files on distributed client
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 sw 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 Plan Stability

DB2 12 plan – 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

Key DB2 12 limitations

- Temporal stabilization not currently included
- REBIND support not included
- No PLANMGMT/SWITCH/APREUSE
**Problem:**
- DB2 provides the DSNUTILU stored procedure to load data from a client
  - But this is difficult to use, app must xfer data to z/OS file

**Solution:**
- DB2 Client API (CLI and CLP) for remote load into DB2
  - Easy/fast loading of data from file that resides on client
  - Internal format (SAP), as well as delimited and spanned (LOB data)
  - Overlap network operations with data ingest on the DB2 server
  - Measured results show as fast or faster than DB2 LOAD utility
  - zIIP eligible
DB2 12: application enablement

Several SQLPL Improvements

SQLPL in triggers, including versioning and debug support
- SQLPL obfuscation
- Support for constants
- Dynamic SQL in SQLPL UDFs and stored procedures

- ARRAY and LOB global variables

- JSON function improvements for easier retrieval of JSON data
DB2 12: application enablement...

- Enhanced MERGE support
- New SQL Pagination syntax
- Piece-wise modification of data (DELETE)
- XMLModify multiple update support
- Bi-temporal improvements
  - Inclusive/inclusive support
  - Temporal RI
  - Logical transaction for system time
Enhanced MERGE

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

- **Numeric-based pagination**
  ```sql```
  SELECT * FROM tab OFFSET 10 ROWS FETCH FIRST 10 ROWS ONLY
  ```

- **Data-dependent pagination**
  - **Existing syntax**
    ```sql```
    WHERE (LASTNAME = 'SMITH' AND FIRSTNAME >= 'JOHN') OR (LASTNAME > 'SMITH')
    ```
  - **New equivalent syntax**
    ```sql```
    WHERE (LASTNAME, FIRSTNAME) > (SMITH, JOHN)
    ```
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**

- Allow the fetch clause to be specified on a searched delete statement

```sql
DELETE FROM T1 WHERE C1 > 7
FETCH FIRST 5000 ROWS ONLY;
COMMIT;
```
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
Partition By Range Current Limitations

- Maximum table size limited to 16Tb (4k pages) or 128Tb (32k pages)

- Maximum number of partitions is also dependent on DSSIZE and page size
  - E.g. if DSSIZE = 256 GB and page size = 4K then Max Parts is 64

- DSSIZE is at Table Space Level not Part Level
  - All Parts inherit the same DSSIZE set at Table Space
  - No ability to have differing Partition sizes
  - Altering DSSIZE requires REORG of entire tablespace
New PBR tablespace structure called ‘PBR RPN’
- Relative page numbers (RPN) instead of absolute
- Remove dependency between #partitions & partition size
- New RID is Relative RID
  - Part Number stored in Partition Header Page
  - Page number stored in Data Page, relative to start of the partition
- Up to 1TB Partition Size, or 4 Petabytes (PB) per table space
- Maximum number of rows with 4K pages increased from 1.1 to 280 Trillion
  - @1,000 rows inserted per second, more than 8800 years to fill!
- **Increasing DSSIZE is supported at partition-level – and is immediate!**
- New DSSIZE support for indexes
- These infrastructure changes position DB2 for future enhancements
  - Increase in partition limits, increase number of rows per page
  - Attribute variance by partition, schema changes via REORG PART
DB2 12 Online Schema Improvements

- Insert partition

- Online deferred ALTER INDEX COMPRESS YES
  - Previously placed indexes in RBDP

- Option to defer column-level ALTERs
  - Materialize through online REORG
  - Avoid availability constraints & conflict with other deferred alters

- TRANSFER OWNERSHIP
Migration & Catalog

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

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

- BRF is deprecated
  - BRF pagesets still supported, but zparm & REORG options are removed

- Temporal RTS tables
  - Defined in catalog, enablement is optional
DB2 12 for z/OS Accelerated Value

Deliver desirable, consumable capabilities to the marketplace with speed and quality

- DB2 for z/OS is moving to a Continuous Delivery model based upon DB2 12
- Why?
  - Faster delivery of easily consumable new features
  - Integrates perfectly with new DevOps
  - Methodologies being adopted by our users
  - Eased deployment burden enables faster
  - Adoption of new technology
  - Available on Replay
- http://ibm.biz/DB2zContinuousDelivery
IBM DB2 Utilities – key to enabling DB2 function

- Continuing evolution of REORG utility
  - Diminishing importance of data re-clustering for application performance
    - Optimizer improvements, I/O performance improvements, caching improvements, contiguous buffer pools
  - Increasing use of IBM REORG for schema evolution
    - Insert partition
    - PBR RPN conversion
    - Deferred column-level alter
    - LOB compression
  - Improved PBG partition management
    - Overflow to new PBG partition to ensure successful partition-level REORG of PBGs
Maximizing Efficiency & Eliminating Application Impact

- Improved efficiency
  - Further reduction in CPU cost & more offload to zIIP
    - REORG up to 57% zIIP offload
    - LOAD up to 90%
  - REGISTER NO option to eliminate data sharing overhead for RUNSTATS, UNLOAD
  - COLGROUP statistics CPU cost reduced by up to 25%, elapsed time up to 15%
  - More efficient handling of compressed data to reduce CPU and elapsed time across range of utilities
  - REORG avoidance: Immediate increase of partition DSSIZE with PBR RPN
  - Improved FlashCopy support
    - Multiple DFSMS COPYPOOL support for SLBs & better messaging
    - Improved FlashCopy handling in REORG & template support for MGMTCLAS, STORCLAS

- Eliminating application impact
  - Improved LOAD utility support for sequences with automatic handling of MAXASSIGNEDVAL
  - Online LOAD REPLACE – non-disruptive refresh of reference tables
  - Skip invalidation of cached statements by RUNSTATS
  - Removed recoverability restrictions for PBG table spaces
IBM DB2 Management with IBM Tools

- Comprehensive and up to the minute support for DB2 12 and beyond
- Automatic exploitation of latest capabilities
- Industry-leading analytics support from current tooling investment
- Simplified, yet powerful management of DB2 systems
- Consolidating information with integrated visual experience via Data Server Manager (DSM)

Database
- V12 schema
- DB2 Analytics Accelerator
- Cloud provisioning

Utilities
- V12 utilities
- Automation
- Self-management

Performance
- V12 instrumentation
- V12 SQL tuning

https://ibm.biz/Bds478
Native REST API
Based on WebSphere Liberty Profile for z/OS - WLP is a faster lightweight version of WebSphere

z/OS Connect V1 – z/OS Connect is a special Java Servlet in WLP to handle connectivity to z/OS services, HTTPS request and reply, REST/JSON message formatting and transformation, forward message to registered service providers, configure services, discovery service, security, auditing for enterprise data

- REST – Representational State Transfer ... Stateless client-server protocol, typically using HTTP URLs that map to a ‘service’, such as ‘query account’ or ‘update data’
- JSON – JavaScript Object Notation … a standard of representing data as a set of name/value pairs. This is passed back and forth along with REST request/responses

- DB2 Adapter maps a service to a single SQL statement (eg. INSERT, SELECT, CALL) execution
- DB2 REST Service SQL statement is executed dynamically, using the DB2 JDBC/SQLJ (JCC) driver
Provides a DB2 integrated solution for enabling RESTful access DB2 Data

- Leverages existing distributed data facility (DDF) capability including thread pooling, profiling and WLM classification of services
- A new HTTP listener is provided to process HTTP requests, accepts JSON payloads and returns results as a JSON response
- Does not require installation of any additional products or require configuring a new network interface
- Description of service is in the catalog tables and a prepared package used to execute the service is saved in the directory
- Provides a set of system defined services that can be used create (DB2ServiceManager) and discover (DB2ServiceDiscover) DB2 services
- Security, accounting, statistics, and auditing uses existing DB2 instrumentation infrastructure
- DB2 maps a service to a single SQL statement (eg. INSERT, SELECT, CALL) in a DB2 REST service package
- DB2 REST Service SQL statement is executed statically, following existing DB2 static package security and authorization semantics
DB2 DDF Native REST services are invoked from z/OS Connect EE by using the z/OS Connect EE REST Client.

- Enables z/OS Connect EE users to include DB2 DDF Native REST Services in z/OS Connect API packages.
Current Status

- **DB2 V11 APAR PI66828**
  - Initial DB2 DDF Native REST Services delivery. Provides basic infrastructure and functionality
  - Will have some limitations (see APAR II14827), such as:
    - Max 1MB request & reply
    - No LOB/XML support
    - CALL is supported, but not with result set
    - And more...
  - **PTF is available now!**

- **DB2 V12 APAR PI70652**
  - Initial DB2 V12 availability. Adds zCEE support and will address limitations in the initial V11 delivery
  - Will also update the DB2 V11 code at this time with another DB2 V11 APAR (PI70477)
Demo !