Shadow Tables en DB2 LUW
Transacciones y analítica en la misma base de datos

Raquel Cadierno Torre
IBM Analytics
@IBMAnalytics
rcadierno@es.ibm.com
16 de Diciembre de 2016
**DB2 Business Differentiators**

**Enterprise Warehouse, Analytic and OLTP Workloads**
- Only database in industry with both shared-disk and shared-nothing architecture - with fully integrated high availability

**Speed of Thought Analytics and Reporting**
- Leveraging columnar and dynamic memory exploitation technologies with BLU acceleration

**24x7x365 Continuous Availability for OLTP**
- Leveraging best of breed z/OS sysplex technology with pureScale

**Next Generation of Applications for NoSQL and Big Data**
- Native support for XML, JSON and Triple Graph data
- Integrated with Mobile devices and emerging Big Data platforms

**Ready for All Deployment Models**
- Time to value and simplicity with PureData Systems
- Virtual, Cloud and Multi-tenancy deployments all available
DB2 Business Differentiators

• Different Workloads are supported by DB2 LUW
  – OLTP workloads
  – OLAP workloads
  – Hybrid OLTP/OLAP (HTAP) workloads

• Underlying IBM leading technology
  – BLU Acceleration
  – Shadow Tables
  – IBM InfoSphere Data Replication (CDC Replication)
OLTP vs OLAP (definition)

- **OLTP** – Online Transaction Processing
  - Use INSERT/UPDATE/DELETE/SELECT to access most of the columns of a few records (even only 1)
  - Quickly process a massive number of transactions
  - Maximize the Transaction Throughput (number of transactions per second)

- **OLAP** – Online Analytical Processing
  - Access only a subset of columns from many records, often as large as the entire table
  - Complex queries may include several JOINs or aggregates (SUM, AVG)
  - Minimize response time

![OLTP and OLAP applications](image-url)
**Scenario: OLTP + OLAP**

Los datos de los sistemas OLTP se replican a un sistema OLAP (escenario habitual)

- Sistemas que operan y se gestionan de forma independiente.
  - Las transacciones por segundo no se ven afectadas por las consultas complejas.
- Requiere de procesos ETL para el movimiento de datos
  - pueden ser complejos e introducir latencia de datos.
Scenario: OLTP + OLAP queries

Sistema OLTP en el que se lanzan consultas analíticas.

- Se trata de un sistema OLTP sobre el que se lanzan consultas analíticas.
  - Requiere tareas adicionales como la creación de índices, MQTs, etc.
- Las transacciones pueden verse afectadas
  - por la ejecución de consultas pesadas, índices, etc.
**Scenario : HTAP**

**Hybrid Transaction / Analytical Processing**

Sistema único que permite cargas de trabajo mixtas.

- Las aplicaciones transaccionales y analíticas acceden al mismo sistema de forma trasparente
- Se basa en la tecnología de Shadow Tables de DB2
DB2 BLU Acceleration (Recap)

Dynamic In-Memory
In-memory columnar processing with dynamic movement of data from storage

Actionable Compression
- Analyze and move data while compressed
- Patented techniques that preserves order so data can be used without decompressing

Parallel Vector Processing
Multi-core and SIMD parallelism (Single Instruction Multiple Data)

Data Skipping
Skips unnecessary processing of irrelevant data

Super Fast, Super Easy — Create, Load and Go!
No Indexes, No Aggregates, No Tuning, No Application/SQL changes, No schema changes
What are Shadow Tables?

A shadow table is a replication-maintained, column-organized materialized query table (MQT).

- Column-organized supported by BLU Acceleration
- A materialized query table (MQT) is a table whose definition is based upon the result of a query.

Maintained by CDC Replication

Use shadow tables to get the performance benefits of BLU Acceleration for analytic queries in OLTP environments.
Shadow Table Features

- Column Organized Table – BLU Technology (no indexes)
- Base table must be row organized
- Only one Shadow Table per Row Organized base table is allowed
- Only one base table reference (no joins allowed)
- Columns cannot be renamed through the column list or AS clause
- Primary key or unique constraint must exist on the base table

- For a query to be eligible for shadow table routing, all referenced row organized tables must have a Shadow Tables defined

- A Shadow Table CREATE DDL must contain the following:
  - DATA INITIALLY DEFERRED
  - REFRESH DEFERRED
  - ENABLE QUERY OPTIMIZATION
  - MAINTAINED BY REPLICATION
  - ORGANIZE BY COLUMN
Shadow Table
Restrictions

• No LOBS, XML in base table
• The base table cannot use LBAC or RCAC
• Base table cannot be range partitioned, DPF, MDC, RCT or Temporal Table
• Queries having RR / RS will not be routed to the shadow tables
• Shadow Tables inherits the restrictions of MQTs
• Available in Advanced editions
Replication Technology CDC

IBM® InfoSphere® Data Replication - CDC Replication is a replication captures database changes as they happen and delivers them to target databases, message queues, or an ETL solution.
CDC Replication for Shadow Tables

InfoSphere CDC software is used for replication to shadow tables

Software and Components:

- CDC Engine for DB2 LUW provides capture and apply components for DB2 for replication purposes - InfoSphere CDC for DB2 for LUW Version 11.3.3 or later releases

- CDC Access Server is a gateway for communication between CDC engine and the management console also provides a command line utility `chcclp` to communicate to the CDC Engine - InfoSphere CDC Access Server Version 11.3.3 or later releases

- Optional: InfoSphere CDC Management Console Version 11.3.3 or later releases
CDC Replication Workflow
Steps to set CDC Replication ready

1. Installing Access Server
2. Installing CDC for DB2 LUW
3. Installing Management Console [optional]
4. Adding and configuring data stores
5. Adding and configuring subscriptions
6. Mapping and customizing tables
7. Starting and ending replication
Meetup DB2 LUW - Madrid

**CDC Replication Workflow (I)**

Steps to set CDC Replication ready means …

1. Installing Access Server

   *Runs at port 10102 & Creates CDC User = db2v11*

2. Installing CDC for DB2 LUW

   *Runs at port 10902 + Creates CDC instance + Config CDC instance Parameters*

<table>
<thead>
<tr>
<th>CDC Instance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 INSTACE</td>
<td>DB2V11</td>
</tr>
<tr>
<td>DB2 DBNAME</td>
<td>ROWDB</td>
</tr>
<tr>
<td>CDC METADATA SCHEMA</td>
<td>CDC</td>
</tr>
<tr>
<td>PATH REFRESH</td>
<td>/datossd/rowdb/load</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDC Instance Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAINTAIN_REPLICATION_MQT_LATENCY_TABLE=TRUE</td>
</tr>
<tr>
<td>FASTLOAD_REFRESH_COMMIT_AFTER_MAX_OPERATIONS=10000000</td>
</tr>
<tr>
<td>MIRROR_AUTO_RESTART_INTERVAL_MINUTES=2</td>
</tr>
<tr>
<td>GLOBAL_MAX_BATCH_SIZE=10000</td>
</tr>
</tbody>
</table>
CDC Replication Workflow (II)
Steps to set CDC Replication ready means …

3. Adding and configuring data stores
   \textit{Datastore} = \textit{cdc} + \textit{Datastore Connections}

\begin{verbatim}
        db2blu (db2v11) /opt/IBM/IDR/AccessServer/bin -> ./dmlistdatastores
Datastore  Hostname     Port      OS         DB        Version
-----------------  -----------  --------  ----------  --------  ---------------
cdc            9.172.170.22 10902 Java VM JDBC   V11R3M3T3BIIDR_Release_49
\end{verbatim}

4. Share Replication Latency between CDC and DB2

\begin{verbatim}
CREATE TABLE "SYSTOOLS"."REPL_MQT_LATENCY"
(
"COMMIT_POINT" BIGINT,
"DELAY_OFFSET" BIGINT
) ...
\end{verbatim}

\begin{itemize}
\item \textbf{COMMIT\_POINT} is a time stamp in seconds of last commit after apply changes
\item \textbf{DELAY\_OFFSET} is number of seconds between the source table data is read and the last time applied changes
\end{itemize}
### CDC Replication Workflow (III)

**Set up Subscription and Table Mapping for Shadow Tables**

5. Adding and configuring subscriptions

   A Subscription defines a container for the table mappings defined for each shadow table. Provides a single point of control for common replication operations (mirroring and refreshing).

   ```
   chcclp session set to cdc;
   connect server hostname <...>  
   add subscription name CDC persistency TRUE;  
   (1 subscription for all shadow tables in the same database)
   
   chcclp session set to cdc;
   ...
   add table mapping sourceschema DB2V11 sourcetable CUSTOMER targetschema SHADOW targettable CUSTOMER targetindexmode index targetindexname SHADOW.PK_CUSTOMER type standard method mirror;
   ```

6. Mapping and customizing tables

   Define the source table for replicating data to the shadow table.

   ```
   chcclp session set to cdc;
   ...
   add table mapping sourceschema DB2V11 sourcetable CUSTOMER targetschema SHADOW targettable CUSTOMER targetindexmode index targetindexname SHADOW.PK_CUSTOMER type standard method mirror;
   ```

7. Starting and ending replication

   **Mirroring vs Refreshing**

   Mirroring is the process that replicates changed data from the source table to the shadow table.
   Refreshing is the process that synchronizes the shadow table with the current contents of source table.
CDC Commands

• Start Access Server (install path = /opt/IBM/IDR/AccessServer/bin)

  $ nohup ./dmaccessserver &

  $./dmcreateuser db2v11 db2v11 db2v11 password SYSADMIN TRUE FALSE TRUE

  $./dmlistusers

• Configure CDC Instance (install path = /opt/IBM/IDR/AccessServer/bin)

  $./dmconfigurets

  $./dmset -I CDCDEMO global_max_batch_size=10000

• Start the instance

  $nohup ./dmts64 -I CDCDEMO &

• Adding DataStore and datastore connections (install path = /opt/IBM/IDR/AccessServer/bin)

  $./dmcreatedatastore cdc “My CDC Datastore for Shadow Tables” <myserverIP> 10092

  $./dmaddconnection db2v11 cdc
Database Environment (I)

DB2 environment is OLTP ready.

- Variable DB2_WORKLOAD=ANALYTICS is NOT set
- Parameters SORTHEAP & SHEAPTHRES_SHR have to be raised to use column organized tables
- Override SORTHEAP values for OLTP workloads:
  DB2_EXTENDED_OPTIMIZATION=OPT_SORTHEAP_EXCEPT_COL nnn

```
db2set DB2_EXTENDED_OPTIMIZATION=OPT_SORTHEAP_EXCEPT_COL 31569
```

Recommended initial values

<table>
<thead>
<tr>
<th>SHEAPTHRES_SHR</th>
<th>40-45% of DATABASE_MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTHEAP</td>
<td>5 -20% of SHEAPTHRES_SHR</td>
</tr>
</tbody>
</table>
Database Environment (II)

Tables are ROW organized by default

- Instance: db2v11
- Database: ROWDB

```
db2blu (db2v11) /home/db2v11 -> cat rowdb.cfg

Default table organization (DFT_TABLE_ORG) = ROW
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 5000000
Sort list heap (4KB) (SORTHEAP) = 500000
Size of database shared memory (4KB) (DATABASE_MEMORY) = AUTOMATIC(24127020)
Catalog cache size (4KB) (CATALOGCACHE_SZ) = 330
Utilities heap size (4KB) (UTIL_HEAP_SZ) = AUTOMATIC(4000000)
First log archive method (LOGARCHMETH1) = DISK:/staging/rowdb/db2arch/
Database memory threshold (DB_MEM_THRESH) = 100
```

Recommended initial values

```
UTIL_HEAP_SZ = 1000000 (AUTOMATIC)
```
Database Environment (III)
Creating Shadow Tables

- Data Model Subset for shadow tables

Sample query q1:
```
select * from (select i_manufact_id, sum(ss_sales_price) sum_sales,
            avg(sum(ss_sales_price)) over (partition by i_manufact_id) avg_quarterly_sales
            from item, store_sales, date_dim, store
            where ss_item_sk = i_item_sk and ...)
```
Database Environment (III)
Creating Shadow Tables

- DDL Sample Shadow Table

```sql
CREATE TABLE SHADOW.ITEM AS
(SELECT * FROM DB2V11.ITEM)
DATA INITIALLY DEFERRED
REFRESH DEFERRED
ENABLE QUERY OPTIMIZATION
MAINTAINED BY REPLICATION
ORGANIZE BY COLUMN
IN USERSPACE1;

SET INTEGRITY FOR SHADOW.ITEM ALL IMMEDIATE UNCHECKED;

ALTER TABLE SHADOW.ITEM ADD CONSTRAINT PK_ITEM PRIMARY KEY ("I_ITEM_SK");
```

After the CREATE TABLE statement execution the table state is set in integrity pending.
CREATE TABLE "DB2V11"."ITEM" (  "I_ITEM_SK" INTEGER NOT NULL,  "I_ITEM_ID" CHAR(16 OCTETS) NOT NULL,  "I_REC_START_DATE" DATE,  "I_REC_END_DATE" DATE,  "I_ITEM_DESC" VARCHAR(200 OCTETS),  "I_CURRENT_PRICE" DECIMAL(7 , 2),  "I_WHOLESALE_COST" DECIMAL(7 , 2),  "I_BRAND_ID" INTEGER,  "I_BRAND" CHAR(50 OCTETS),  "I_CLASS_ID" INTEGER,  "I_CLASS" CHAR(50 OCTETS),  "I_CATEGORY_ID" INTEGER,  "I_CATEGORY" CHAR(50 OCTETS),  "I_MANUFACT_ID" INTEGER,  "I_MANUFACT" CHAR(50 OCTETS),  "I_SIZE" CHAR(20 OCTETS),  "I_FORMULATION" CHAR(20 OCTETS),  "I_COLOR" CHAR(20 OCTETS),  "I_UNITS" CHAR(10 OCTETS),  "I_CONTAINER" CHAR(10 OCTETS),  "I_MANAGER_ID" INTEGER,  "I_PRODUCT_NAME" CHAR(50 OCTETS) )
ORGANIZE BY ROW
DATA CAPTURE CHANGES
IN "USERSPACE1"
COMPRESS NO;

ALTER TABLE "DB2V11"."ITEM" ADD CONSTRAINT "SQL161212120255520" PRIMARY KEY ("I_ITEM_SK");
CDC Replication Configuration (definitions)
Set up Subscription and Table Mapping for Shadow Tables

- Create one Subscription for all shadow tables in the same database

- Create Table Mappings

Define the source table for Replicating data to the shadow table.

```sql
add table mapping sourceschema DB2V11 sourcetable ITEM targetschema SHADOW targettable ITEM targetindexmode index targetindexname SHADOW.PK_ITEM type standard method mirror;
```

- Start Mirroring

```sql
start mirroring method continuous;
```

Monitor directory (path refresh) to see the data files used by CDC to load into shadow tables

- After LOAD source and shadow tables have same data

<table>
<thead>
<tr>
<th>Table Name</th>
<th>DB2V11</th>
<th>SHADOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMER</td>
<td>500000</td>
<td>500000</td>
</tr>
<tr>
<td>DATE_DIM</td>
<td>73049</td>
<td>73049</td>
</tr>
<tr>
<td>ITEM</td>
<td>102000</td>
<td>102000</td>
</tr>
<tr>
<td>STORE_SALES</td>
<td>28800991</td>
<td>28800991</td>
</tr>
<tr>
<td>STORE</td>
<td>102</td>
<td>102</td>
</tr>
<tr>
<td>CUSTOMER_ADDRESS</td>
<td>250000</td>
<td>250000</td>
</tr>
</tbody>
</table>

6 record(s) selected.
High Level Architecture for Shadow Tables

Hostname: db2blue
IP: 9.172.170.22

Instance: db2v11
Database: ROWDB

OLTP Transactions

Optimzer

Log

Data changes applied asynchronously

Single CDC Instance

Capture Engine
Apply Agent

ROW Table
SHADOW Table

OLAP Queries
Query Routing to Shadow Tables

- **DB2 Engine** chooses between row organized and shadow tables automatically based upon certain conditions:
  - All tables referenced in the query should have a shadow table
  - `INTRAPARTITION` parallelism is enabled for the sesión
    
    ```
    db2 get dbm cfg | grep -i INTRA_PARALLEL
    ```
  - **Special register** `CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION` is set to be considered by the optimizer while processing dynamic SQL queries
  - `CURRENT REFRESH AGES` is set to a value other than 0

```
  db2blu (db2v11) /home/db2v11/demo/shadow ->db2 -tvf ..//queries/qlatency.sql
  select latency, refresh_age, delay_offset from latency

<table>
<thead>
<tr>
<th>LATENCY</th>
<th>REFRESH_AGE</th>
<th>DELAY_OFFSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.06058</td>
<td>500.000000</td>
<td>0</td>
</tr>
</tbody>
</table>

  1 record(s) selected.
```
Sample 1

db2 connect to rowdb
db2 set current explain mode explain
db2 -tvf 1.sql
db2 set current explain mode no
db2exfmt -d ROWDB -1 -o 1.sql.exfmt
cat 1.sql.exfmt

The following MQT was not used in the final access plan, because the plan cost with this MQT was more expensive or a better candidate was available.
Sample 2

db2 connect to rowdb
db2 set current explain mode explain
db2 -tvf 2.sql
db2 set current explain mode no
db2exfmt -d ROWDB -1 -o 2.explain
cat 2.explain

The following MQT or statistical view was considered in query matching: “SHADOW” …
Links of Interest

IBM Knowledge Center

Shadow Tables

IBM BLU Acceleration

IBM BigData Hub
Gracias