THE FILLMORE GROUP
Relational Database Solutions

IDAA v4.1 PTF 5 - Update

The Fillmore Group – June 2015

A Premier IBM Business Partner

Premier Business Partner IBM

History

The Fillmore Group, Inc.

- Founded in the US in Maryland, 1987
- IBM Business Partner since 1989
- Delivering IBM Education since 1994
- DB2 Gold Consultant since 1998
- IBM Champions since 2009

Premier Business Partner IBM

THE FILLMORE GROUP
Relational Database Solutions

The Fillmore Group, Inc.

- DB2 Technical Support and Consulting
- IBM Training Partner with Global Training Partner Arrow ECS
- IBM Information Management Software Reseller

IBM CHAMPION

THE FILLMORE GROUP
Relational Database Solutions

IBM DB2 Analytics Accelerator

What is it?

The IBM DB2 Analytics Accelerator is a workload optimized, appliance add-on to DB2 for z/OS, that enables the integration of business insights into operational processes to drive winning strategies. It automatically accelerates select queries, with unprecedented response times and negligible MIPS impact.

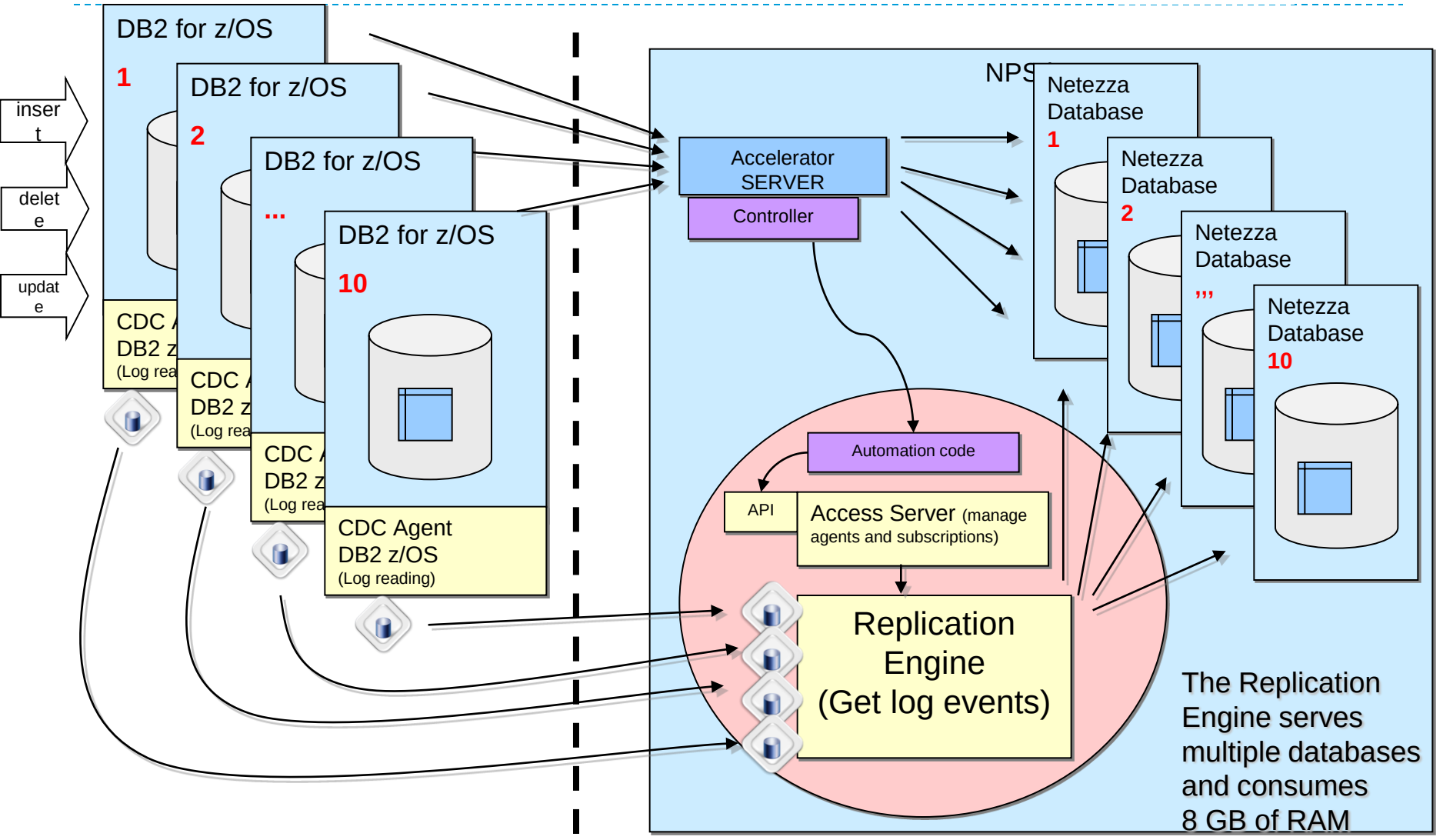
How is it different?

- Performance: unprecedented response times to enable 'train of thought' analyses frequently blocked by poor query performance.
- Integration: deep integration with DB2 for z/OS 10 and 11 provides transparency to all applications.
- Self-managed workloads: queries are automatically executed in the most efficient location
- Transparency: applications connected to DB2 are entirely unaware of the Accelerator
- Simplified administration: appliance hands-free operations, eliminating most database tuning tasks

Breakthrough Technology Enabling New Opportunities

N3001 Model	001	002	005	010	020	040	080
Cabinets	n/a	¼	½	1	2	4	8
S-Blades	n/a	2	4	7	14	28	56
SPU CPU Cores	40	40	80	140	280	560	1120
SPU FPGA Cores	n/a	32	64	112	224	448	896
Capacity in TB (user data space)	4	8	24	48	96	192	384
Effective Capacity in TB (assuming 4x compression)	16	32	96	192	384	768	1536

N2002 Model	002	005	010	020	040
Cabinets	¼	½	1	2	4
S-Blades	2	4	7	14	28
SPU CPU Cores	32	64	112	224	448
SPU FPGA Cores	32	64	112	224	448
Capacity in TB (user data space)	8	24	48	96	192
Effective Capacity in TB (assuming 4x compression)	32	96	192	384	768


DB2 for z/OS
1
DB2 for z/OS
2
DB2 for z/OS
...
DB2 for z/OS
10
CDC Agent
DB2 z/OS
(Log reading)
insert
delete
update
Accelerator
SERVER
Controller
Netezza
Database
1
Netezza
Database
2
Netezza
Database
...
Netezza
Database
10
Automation code
API
Access Server (manage
agents and subscriptions)
Replication
Engine
(Get log events)
The Replication
Engine serves
multiple databases
and consumes
8 GB of RAM

PTF 5 enhancements at a glance

Improved transparency

- **Suspending faulty tables from Incremental Update (IU) replication - Isolation of tables in error**
  - Single tables are suspended from replication if an error on the table(s) occurs on either Capture or Apply side
  - *Ensures that replication continues automatically for the remaining tables of the subscription*

PTF 5 enhancements at a glance

Enhanced capabilities

- **Improved Incremental Update performance**
  - Changes to tables can now be **applied in parallel** to up to 4 tables at a time
  - DB2 Analytic Accelerator now **sets detected unique constraints of the DB2 table automatically as Distribution Keys and Organizing Keys** for tables that are enabled for Incremental Update
  - Increased Netezza priority for all replication-related processes

PTF 5 enhancements at a glance

Enhanced capabilities

- **In-database Transformation**
  - Introduction of **Accelerator-only Tables to store intermediate or final results** of data transformation or reporting processes
    - *Accelerate in-database data transformations and data movement processes*
    - *Enables multi-step reporting on the Accelerator*
    - *Saves disk space and CPU cost on System z currently used for transformations and reporting steps*

PTF 5 enhancements at a glance

Enhanced capabilities

- **DB2 now caches accelerated queries in the Dynamic Statement Cache (DSC)**
  - DB2 10: Available with APAR PI22531
  - DB2 11: in development
  - *Significantly improves performance of PREPARE statements for accelerated dynamic queries*

PTF 5 enhancements at a glance

Enhanced capabilities

- **New set of statement level monitoring counters**
  - Report **SQL statement's eligibility** to be routed to the Accelerator
  - Report **SQL statement's execution statistics** for queries that are routed to the Accelerator
  - DB2 10: Available with APAR PI23083
  - DB2 11: in development
  - *Simplifies planning and management of workloads on the Accelerator*

PTF 5 enhancements at a glance

More query acceleration

- **DB2 now supports application-period Temporal Tables for loading to the Accelerator and query acceleration**

PTF 5 enhancements at a glance

More query acceleration

- **DB2 now supports accelerating queries using the following scalar functions:**
  - DECODE
  - LTRIM and RTRIM: in v10 DB2 allows a 2nd argument for the trim expression
  - TRIM
  - NVL

Isolation of tables in error

- **Single tables are suspended from replication if an error on the table(s) occurs on either Capture or Apply side**
  - E.g. Decompression errors, datastore errors, out-of-synch errors
- Process: replication stops for all tables, faulty table is suspended, replication restarts for the remaining tables
  - Before:
    - Replication stopped for all tables
    - DB2 Analytics Accelerator restarted replication because the stop was not initiated by the user
    - Error occured again
    - Replication stopped/started again -> loop
- **Suspending faulty tables ensures that replication continues automatically for the remaining tables of the subscription**
  - Suspending the table and restarting replication may take up to 2 minutes

Improved Incremental Update performance

- **Changes to tables can now be applied in parallel to 4 tables at a time**
  - Before:
    - Changes were applied serially, one-by-one
- **DB2 Analytics Accelerator now sets detected unique constraints of the DB2 table automatically as Distribution Keys and Organizing Keys for tables that are enabled for incremental update**
  - Before:
    - User had to do it manually

Distribution and Clustering in the Accelerator

- In general, use defaults
  - No Distribution Keys - random/balanced distribution
  - No Organizing Keys – no sequencing of data in a node

- For larger tables, joined frequently, orders of magnitude improvement can be seen by specifying Organizing and/or Distribution Keys
  - Distribution Key:
    - Determines how data is partitioned across worker nodes
  - Organizing Key:
    - Determines the row sequence on disk within each node

Premier Business Partner IBM

Parallel Apply – Setup and usage

- Optional feature, default: Off
- Needs to be enabled per (replicated) DB2 subsystem using the configuration console
- Possible impacts:
  - The parallel Apply threads might lead to a high disk utilization on the Accelerator. A high disk utilization might impact query performance.
  - The Referential Integrity between replicated tables cannot be guaranteed in all cases because the changes that are effected by the different threads must be committed and that cannot be done at the same time for all threads.

Automatic definition of keys

- Background:
  - An UPDATE statement on the DB2 source table results in a corresponding DELETE and INSERT statement on the (IU enabled) Accelerator target table
  - Performance of DELETE statements can heavily be improved by setting the right Distribution and Organization Keys
    - Avoid full table scans to find the records to delete
- DB2 Analytics Accelerator V4 PTF 5 defines these keys automatically when a table is enabled for IU
  - *Improves out-of-the-box replication performance*
  - **Note:** already enabled IU tables are not affected
- DB2 Analytics Accelerator uses detected unique constraints such as a primary key for the automatic Distribution and Organizing Key setting
  - In most cases beneficial for query execution, too

Rules for automatic key selection

- Unique constraints that qualify for automatic key selection
  - DB2 side primary index PRIMARY KEY
  - DB2 side unique constraint on non-nullable columns (CREATE UNIQUE INDEX)
  - DB2 side unique constraint on nullable columns (CREATE UNIQUE INDEX)
  - DB2 informal constraint (UNIQUE keyword in CREATE TABLE statement)
- Constraints that do NOT qualify for automatic key selection
  - DB2 side constraint on nullable columns with WHERE NOT NULL clause (CREATE UNIQUE WHERE NOT NULL INDEX)
- Note: execute RUNSTATS on a table before you enable it for replication. This ensures that DB2 Analytics Accelerator can determine the best unique constraint to use for the key selection
  - Cardinality value is taken into account for key selection

In-database transformation

- **Introduction of Accelerator-Only tables (AOT) to store intermediate or final results of data transformation or reporting processes**
  - *Accelerate in-database data transformations and data movement processes*
  - *Enables multi-step reporting on the accelerator*
  - *Saves disk space and CPU cost on System z currently used for transformations and reporting steps*

In-database transformation

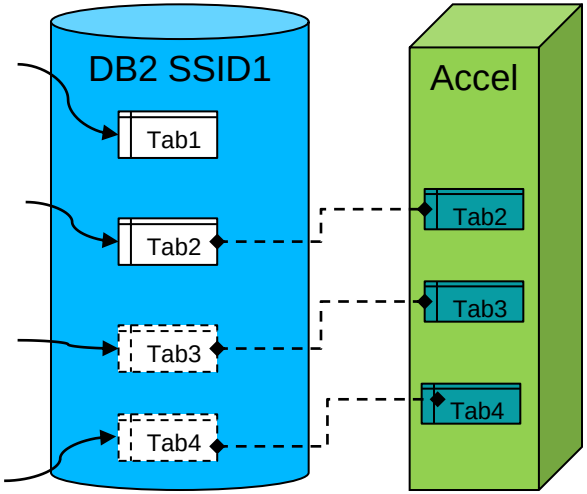
Table Types:

Non-accelerated DB2 table
- Data in DB2 only

Accelerated DB2 table
- Data in DB2 and the accelerator

Accelerator-Only table (AOT)
- with "Proxy table" in DB2
- Data is in accelerator only

Archive table
- with empty read-only table in DB2
- Data is in accelerator only



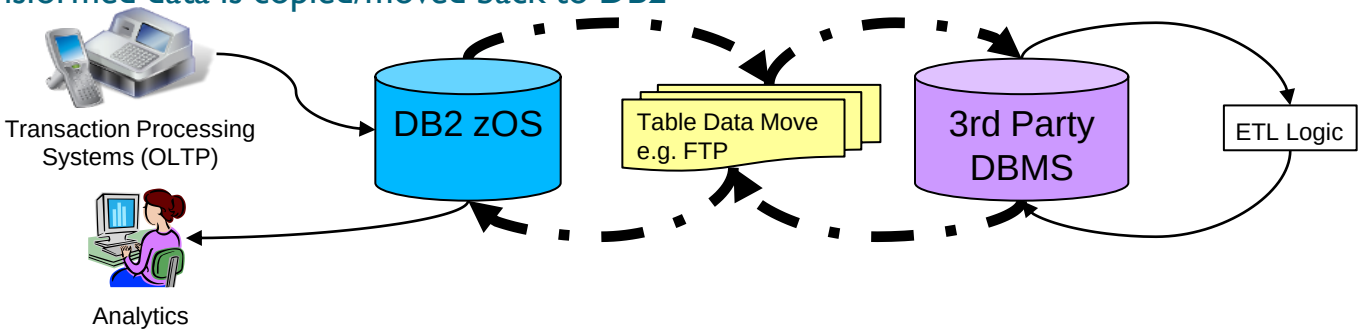
High Performance Storage Saver (HPSS)

- Historical and archival data need only reside on the Accelerator
- Saves DB2 for z/OS storage costs
- Provides cost-effective means to retain data "online" for search and analysis
- Supports auditing and compliance
- Special Register: GET_ACCEL_ARCHIVE

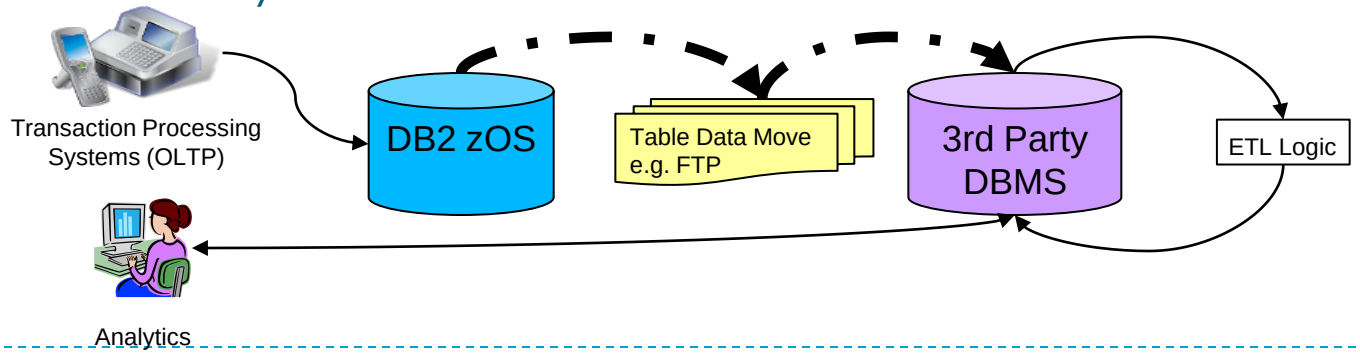
Premier Business Partner IBM

Use Case 1 - ETL

- ETL on a different DBMS – data copied outside of DB2 to transform it
  - Process-driven movement of large amounts of data
  - Aged data for analytics/reporting depending on performance of data movement and transformation process
  - Example 1: Transformed data is copied/moved back to DB2

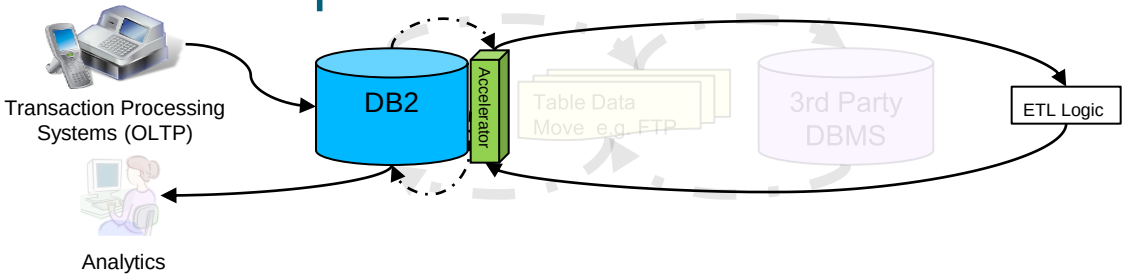


  - Example 2: Transformed data stays on the different DBMS

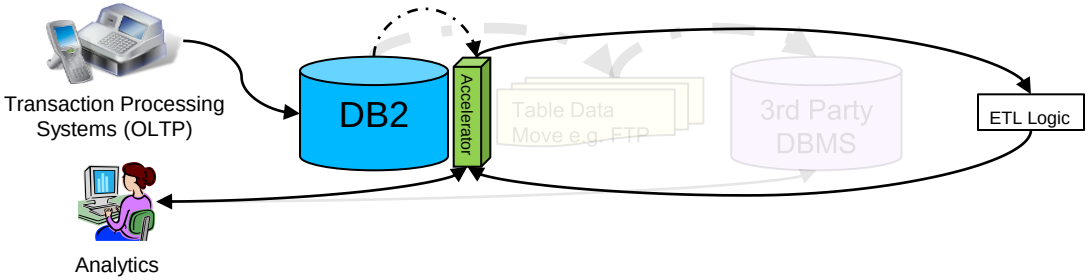


Use Case 1 - ETL (contd.)

- Using AOTs the ETL logic runs on the Accelerator instead of on a different DBMS
  - Simpler to manage (also security)
  - Better performance and reduced latency (no data movement)
  - Intermediate/final results can be kept on the Accelerator
    - In an acclerated table



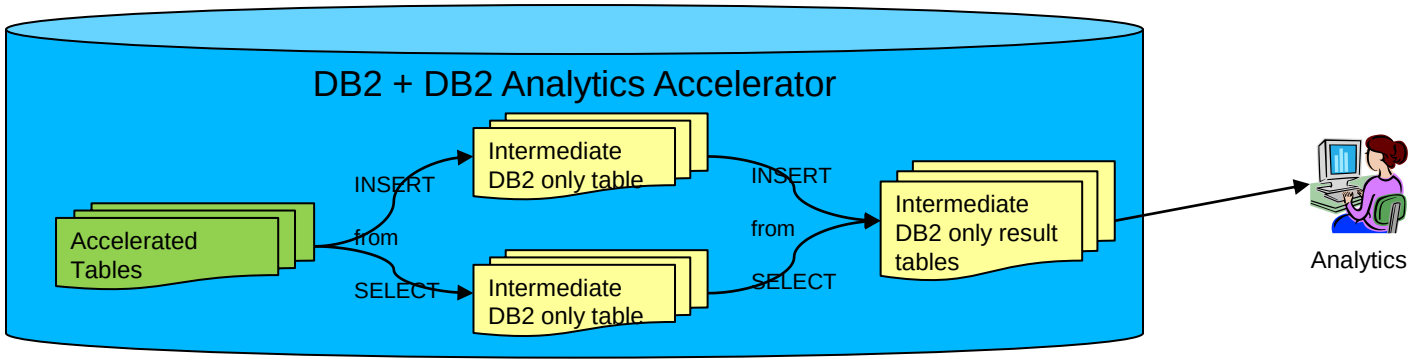
    - In an AOT



Use Case 2 - Intermediate table usage

For analytics or reporting purposes applications might

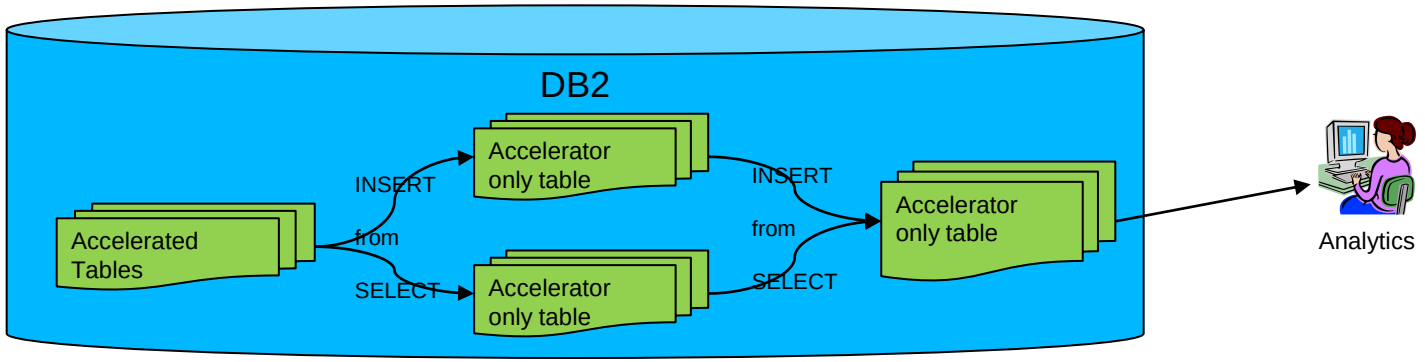
- Either create temporary tables to store results
- Or create regular tables and drop them after report closure (CREATE, DROP)
- Example: Microstrategy



- Without AOTs these intermediate tables are created on DB2
  - These tables must be loaded again for queries to be accelerated

Use Case 2 - Intermediate table usage (contd.)

- With AOTs all steps are processed in the accelerator
  - CREATE and DROP
  - Format 2 INSERT (with SELECT)
- Instead of creating intermediate (temporary or regular) tables in DB2, create AOTs in the accelerator
  - Prereq: Input tables must be accelerated DB2 tables. It is not possible to use non-accelerated DB2 tables as input tables for AOTs (INSERT from SELECT)



Accelerator-Only Tables – Technical basics

- Historical and archival data need only reside on the Accelerator: AOTs are created and dropped using DB2 DDL statements (CREATE; DROP)
  - Accelerator must be started
  - QUERY ACCELERATION behaviour can have any value during CREATE/DROP
  - Syntax:
    - `CREATE TABLE MYTABLE (...) IN ACCELERATOR <ACCEL1>;`
    - `DROP TABLE MYTABLE;`

Accelerator-Only Tables – Technical basics

- Queries using AOTs can only run on the accelerator
  - QUERY ACCELERATION behavior must be set to ENABLE/ELIGIBLE/ALL
- AOTs can be subject to INSERT/UPDATE/DELETE operations on other accelerated tables or AOTs
- Column type and SQL query restrictions of accelerated tables apply to AOTs, too
- Dynamic and static SQL can be used on AOTs
- Currently supported on DB2 10 for z/OS only
- *"Online" for search and analysis*
- *Supports auditing and compliance*

Dynamic Statement Cache (DSC) support

- New query preparation process:
  - For a new arriving query DB2 searches the DSC to determine whether the query has been prepared before
  - The ACCELERATED column value in the DSC defines what happened to the query when it was prepared before
    - Values: NO, YES, NEVER
  - The query acceleration behavior together with the ACCELERATED column in the DSC defines whether the query can be accelerated
    - Before the query is accelerated DB2 checks whether the accelerator is active and whether all involved tables are enabled
  - The DSC can have multiple entries for the same query depending on the query acceleration behavior on when the statement was prepared previously
  - If there is no matching record, the query is treated like a new one, goes through full preparation and a new record is added to DSC

Dynamic Statement Cache (DSC) support

- DB2 APARs:
  - DB2 10: PI22531 (PTF UI24359) - available
  - DB2 11: PI28660 – not available yet
  - This feature is provided by DB2 APARs only - PTF 5 is not required

Statement level monitoring support

- Report SQL statement's eligibility to be routed to the accelerator
  - zparm or Special Register for Accelerator Modeling must be enabled
- Report SQL statement's execution statistics for queries that are routed to the Accelerator
  - Execution statistics include:
    - Elapsed, execution, and wait times
    - CPU time
    - Number of rows and bytes returned
- *Simplifies planning and management of workloads on the Accelerator*

Statement level monitoring support

- Exposed via **IFCID 316, IFCID 401** and in the **DSN_STATEMENT_CACHE_TABLE**
- DB2 APARs:
  - DB2 10: PI23083 (available)
  - DB2 11: PI30005 (not yet available)
  - This feature is provided by DB2 APARs only - PTF 5 is not required

Application-period Temporal Table support

- **Application-period Temporal Tables are now supported for loading and query routing**
- An application-period temporal table …
  - is a table that stores the "in effect" aspect of application data. Use an application-period Temporal Table to manage data based on time criteria by defining the time periods when data is valid.
  - includes a BUSINESS_TIME period with columns that indicate the time period when the data in that row is valid or in effect. You provide the begin time and end time for the BUSINESS_TIME period associated with each row.

Application-period Temporal Table support

- **Implementation in DB2:** create or alter a base table with the BUSINESS_TIME attribute and include a pair of timestamp or date columns, stored by the application:
  - Begin column: represents the time at which row data begins to be valid
  - End column: represents the time at which row data is no longer valid

Application-period Temporal Table support

- **Example:**

```
CREATE TABLE policy_info
(policy_id CHAR(4) NOT NULL,
coverage INT NOT NULL,
bus_begin DATE NOT NULL,
bus_end DATE NOT NULL,
PERIOD BUSINESS_TIME(bus_begin, bus_end));
```

  - Limitation: TIMESTAMP(12) columns as BUSINESS_TIME columns are not supported by DB2 Analytics Accelerator

Acknowledgements

- Patric Becker, IBM Boeblingen Lab

Contacts

- Kim May
  - kim.may@thefillmoregroup.com
  - twitter.com/KimMayTFG
  - www.linkedin.com/pub/kim-may/4/462/84
- Frank Fillmore
  - frank.fillmore@thefillmoregroup.com
  - twitter.com/ffillmorejr
  - www.linkedin.com/pub/frank-fillmore/6/597/9a6/
  - tinyurl.com/ChannelDB2
  - Flipboard for iPad, iPhone, Android: "BigData"