DB2 12 — The ultimate enterprise database for business-critical transactions and analytics

Optimizing Data Transformation with Db2 for z/OS and Db2 Analytics Accelerator

Maryela Weihrauch
IBM Distinguished Engineer z Systems Analytics
weihrau@us.ibm.com

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

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 is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including 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 results similar to those stated here.

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Agenda

- Challenges of typical ETL processing today
- ETL Modernization
  - History generation using Db2 system temporal transparent archiving
  - Integrate more data sources using IDAA Loader V2
  - In-database transformation with IDAA using AoT
- Real-time data transformation for data consumability in SQL via VIEWs
  - (optional) Optimization by separating subqueries accessing categorial data that can be pre-calculated

Challenges of Typical ETL Processing Today

- **Processing pattern**
  - Move data from original data source(s) through ETL tools or custom transformation programs to target DW/DM
  - Typically, data is stored several times in intermittent staging areas
- **Myth: main purpose for ETL**
  - To make data consumable for end users
  - To optimize for performance (star schema)
  - Merging and cleansing (making consistent)
- **Reality: majority of the ETL processing is generating history data**
  - SLA of OLTP “data generation” workloads
  - Little communication between OLTP and DW teams
  - …

Challenges of Typical ETL Processing Today…

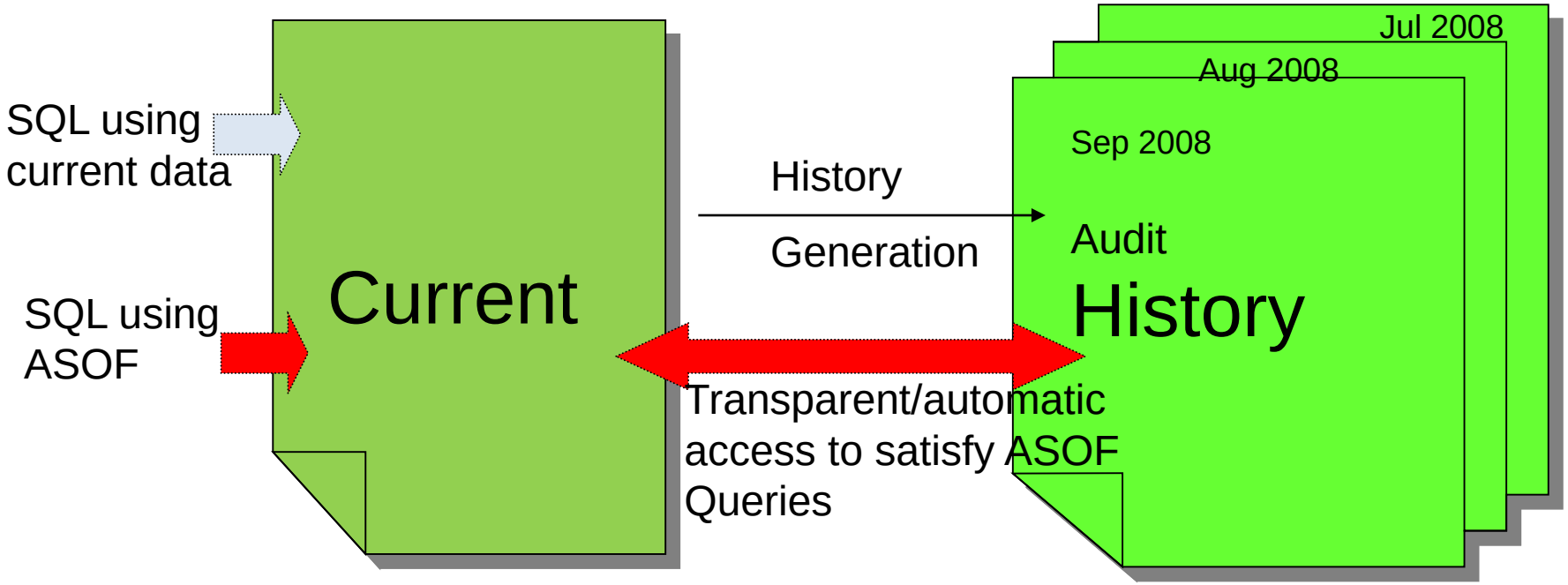
- **Problems with current ETL architecture**
  - Latency of data typically >1 day, not acceptable any longer
  - Amount of data ever increasing -> prolonging ETL window even more
  - New business requests typically declined if data is not readily available in DW or it takes months to implement ETL process for new data elements

- **Motivation to look into an alternative architecture**
  - Reduce/Eliminate the latency associated with data transformation and movement
  - Improve trust in transformed data if used in external analytical service offerings
  - More agile - respond quickly to new business requirements including new data elements

- Functionality in Db2 and IDAA can help to implement an alternative ETL architecture that delivers data with agility, significantly less latency, user consumable and with great performance

ETL Modernization - History Generation
Db2 System Temporal

History table contains version of every update on a single row



ETL Modernization - History Generation
Db2 System Temporal

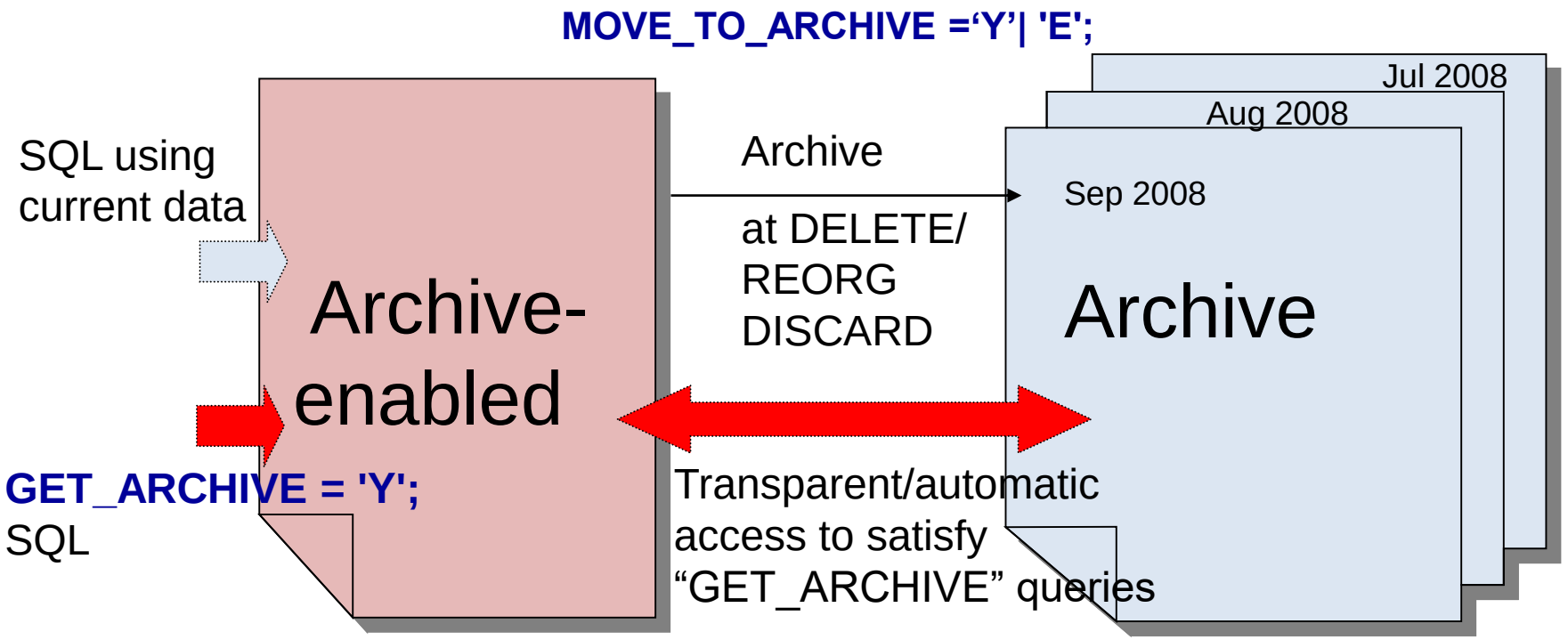
- **Concept of period** (SYSTEM_TIME and BUSINESS_TIME periods)
  - A period is represented by a pair of datetime columns in Db2 relations, one column stores start time, the other one stores end time
  - **SYSTEM_TIME period** captures DB2's creation and deletion of records. Db2 SYSTEM_TIME versioning automatically keeps historical versions of records
  - **BUSINESS_TIME period** allows users to create their own valid period for a given record. Users maintain the valid times for a record.
- **Temporal tables**: System-period Temporal Table (STT), Application-period Temporal Table (ATT), bitemporal table (BTT)
- **DML syntax** allow query/update/delete data for periods of time
  - **Period specification with base table reference:**

    SELECT … FROM ATT/BTT ***FOR BUSINESS_TIME AS OF exp/FROM exp1 TO exp2/BETWEEN exp1 AND exp2*** *...;*
    SELECT … FROM STT/BTT ***FOR SYSTEM_TIME AS OF exp/FROM exp1 TO exp2/BETWEEN exp1 AND exp2*** *...;*
  - **Period clause with base table reference:**

    UPDATE/DELETE FROM ATT/BTT ***FOR PORTION OF BUSINESS_TIME FROM exp1 TO exp2*** *...;*
- **Business value**:
  - It helps meet compliance requirements
  - It performs better
  - It is easier to manage compared to home-grown solutions

ETL Modernization - History Generation
Db2 Archive Transparency

History table contains version of every update on a single row



Adding Archive Timestamp for Transparent Archiving

- Adding row change timestamp is not required but recommended
- Row change timestamp column must be added to both base and archive table.
  - MODIFY_TS in the base table POLICY_INFO
    contains the timestamp when the row was inserted or last updated.
  - MODIFY_TS in the archive table ARCHINVE_POLICY_INFO
    contains the insert timestamp, which is the **archive timestamp**.

Use the clause GENERATED ALWAYS FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP:

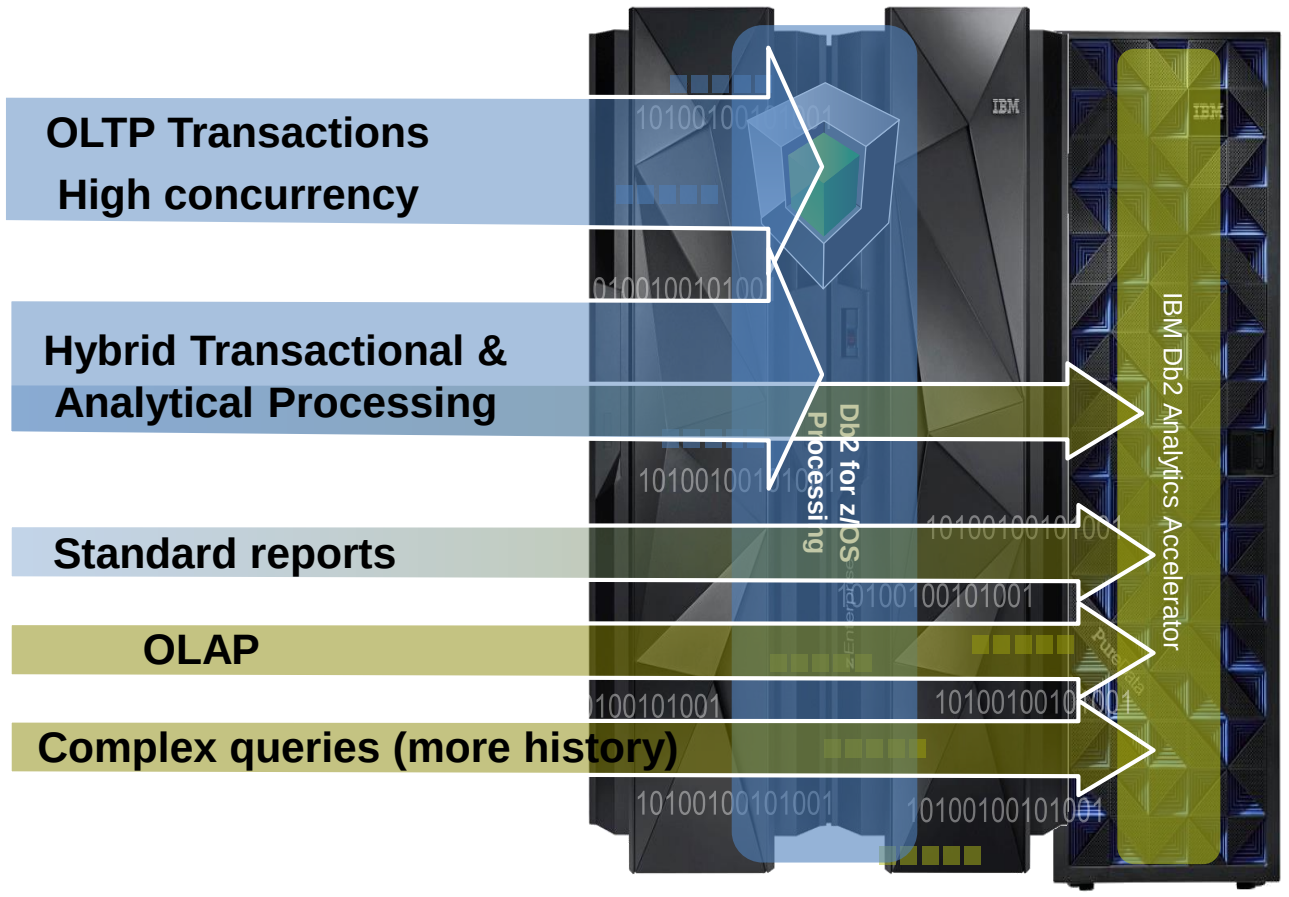
```
CREATE TABLE POLICY_INFO
  (POLICY_ID CHAR(4) NOT NULL ,
   COVERAGE INT NOT NULL ,
   MODIFY_TS TIMESTAMP(6) NOT NULL GENERATED ALWAYS
   FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP);

CREATE TABLE ARCHIVE_POLICY_INFO
  (POLICY_ID CHAR(4) NOT NULL ,
   COVERAGE INT NOT NULL ,
   MODIFY_TS TIMESTAMP(6) NOT NULL GENERATED ALWAYS
   FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP);

ALTER TABLE POLICY_INFO ENABLE ARCHIVE USE
    ARCHIVE_POLICY_INFO;
```

Table Layout Alternatives

- **Tables with system time period**
  - Version of every update on a single row
  - Partitioning approach
    - Active table partitioned by key, e.g. accounted
    - History table partitioned by system_time_end timestamp for sequential insert
    - Will allows for archiving to IDAA on partition level

- **Archive-enabled tables (transparent archiving)**
  - Row exists in active or archive table
  - Partitioning approach
    - Add modify_ts column as archive timestamp (generated row-change-timestamp)
    - Archive-enabled table partitioned by key, e.g. accounted
    - Archive table partitioned by modify_ts timestamp for sequential insert
      (apar PI63830 - generated row-change-timestamp as partitioning column)
    - Allows for archiving to IDAA on partition level

Db2 for z/OS with IDAA



Db2 12 CPU savings target

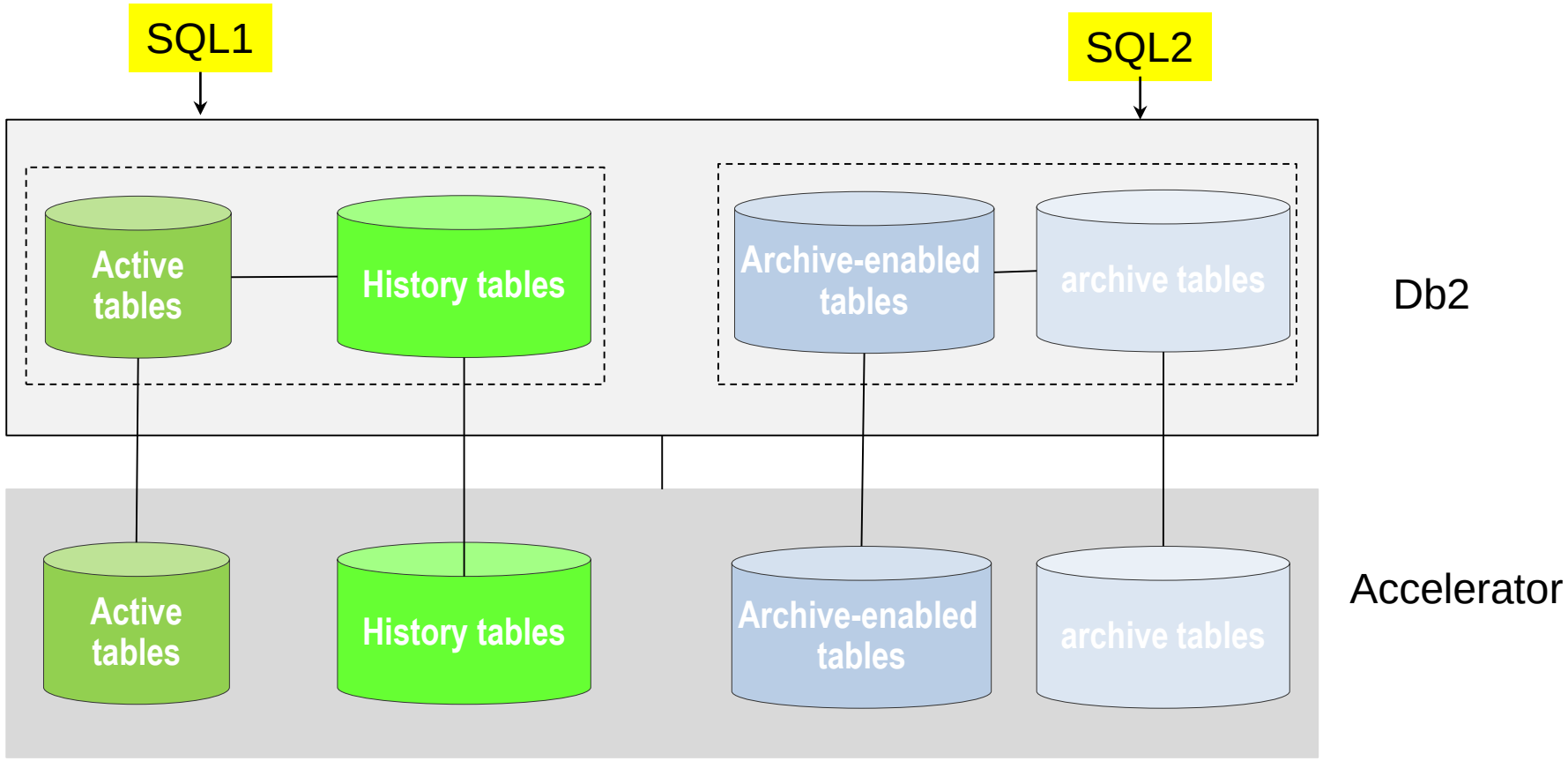
- Operational (in transaction) analytics
- (complex) OLTP

IDAA focus

- Ad-hoc queries
- Complex queries scanning large amount of data
- ETL acceleration/virtual transformation

Combining History in Db2 and on the Accelerator

Both active|archive-enabled and history|archive table need to be accelerated to route SQL to IDAA

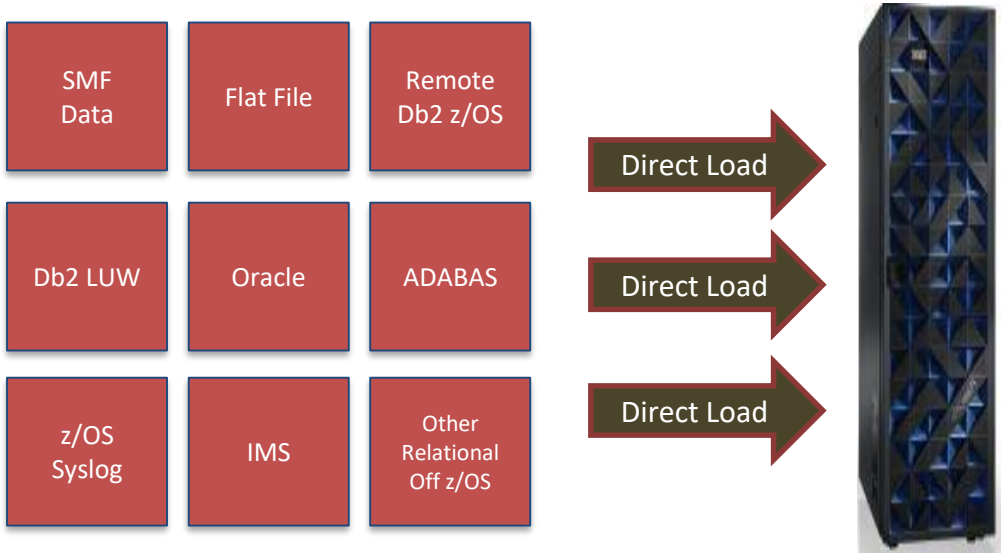


IBM Db2 Analytics Accelerator Loader V2.1

- Provides high performance and extended load capabilities
  - Allows loading of Db2 image/Log data with no table locking
  - Provides fast load of Db2 format file data into accelerator only

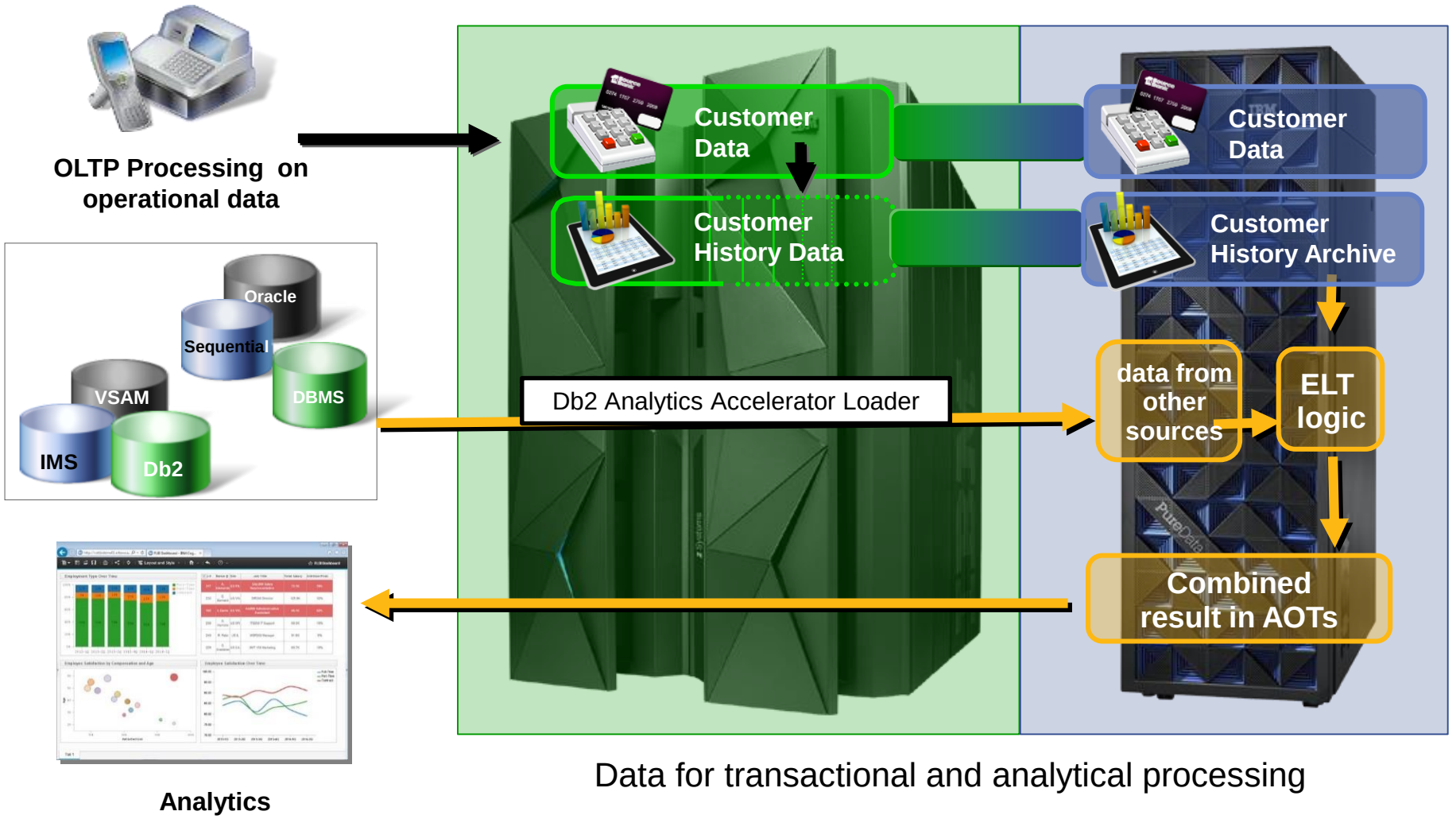
- Addresses challenges identified in loading non Db2 data
  - Manual – Labor intensive ETL
  - Slow – due to staging data to disk

- Additional Features
  - Load Resume
  - Mapping of non relational data
  - Views to load 100s of SMF records
  - Ability to load Syslog data



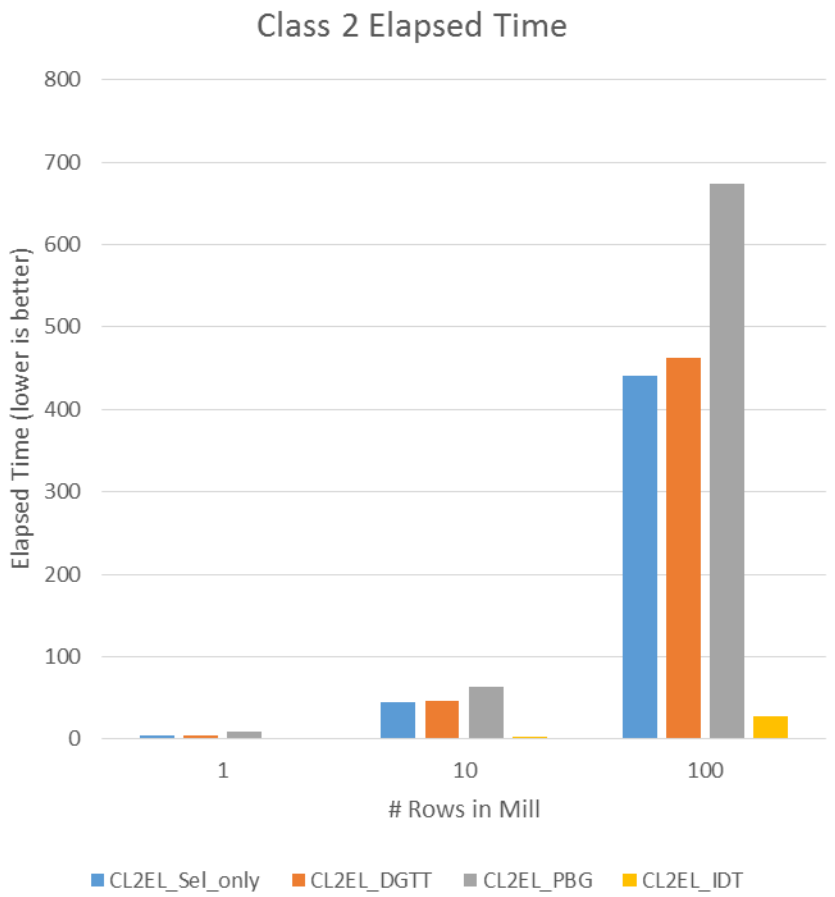
Integrate more data sources for analytics

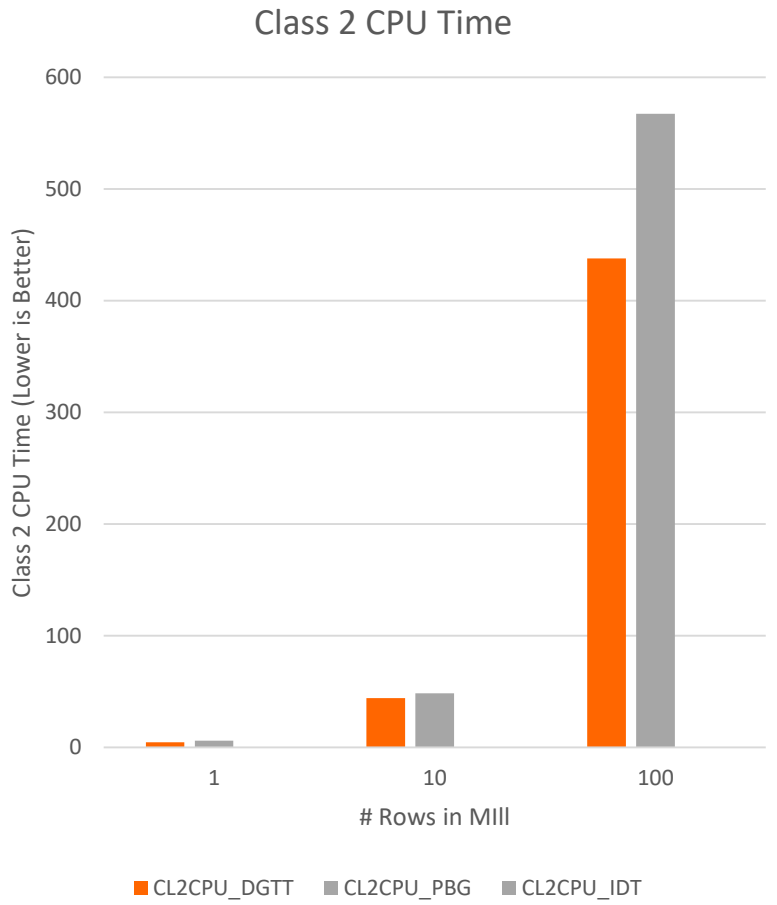
Load external data and combine it with operational or historical archived data for analytics. Save combined results in accelerator-only tables (AOTs)



Performance INSERT FROM SELECT – Many Rows

Up to 95% better elapsed time and negligible CPU time in Db2 for INSERT FROM SELECT into accel-only tables for large amount of data





Real-time Data Transformation for Data Consumability in SQL via VIEWs

- Transformation logic is often expressed in SQL
  - CASE Statements often attach columns just like a join
  - Outer Joins attach columns for categorical, key and fact data
  - UNIONs append data from multiple applications and/or time periods
  - Embedded "Select sum(..) group by“ often used to order and categorize
  - Embedded "Select max(...) group by“ often used to order and categorize
  - Max(Effective date) is used to group period columns within a category
  - Multiple uses of sub-string transform columns into categorical data
  - …
- These typical transformations imply opportunities for the data model to meet reporting requirements
- Why not standardize these transformations and simplify consumability?

Real-time Data Transformation for Data Consumability in SQL via VIEWs

- VIEWs can hide SQL complexity from user and contain the intelligence to retrofit data and simplify access
- Can reflect existing DW/DM schema and keep existing workloads running
- Views can include the transformations necessary to simplify data for end user consumption
  - Rewrite complex SQL within views or..
  - Leverage existing database objects (dimensional structures) to transform and standardize data within the views
- Repetitive transformations from “operational data” to “information data” could be standardized by leveraging data mart modeling techniques and objects and by staging prepared data objects prior to their joins to fact data
- Removing the complex processes and prestaging the data could significantly improve performance and simplify data access to operational data

Real-time Data Transformation for Data Consumability in SQL via VIEWs

- Performance implications due to transformations executed multiple times on data which is in essence categorical (data that only changes periodically i.e. semi-static data)
- (Optional) optimization opportunity by separating subqueries generating categorical, “semi-static data”.
  - If it rarely changes why derive the value every time?
  - Pre-calculate and materialized to make user queries more efficient and avoid recalculating the same result set multiple times.
- Db2 Analytics Accelerator, database performance objects, materialized query tables and Accelerator-Only Tables are optional potential opportunities to enhance performance
- This approach can be combined with archiving in order to optimize operational processing and information retrieval

Benefits of “query-able archive” on Accelerator

- **Performance – avoided transformation to recreate history in DW and queries, even scanning years of data can complete in second**

- **Insight: you can leverage large amounts of historical data for decision support purposes, versus having the data just sit there**
  - If historical data is archived offline, it provides no business intelligence value
  - Even if historical data is kept online, if queries targeting the data perform poorly then analytical usage may be minimal

- **Cost-effective: large majority of table’s data physically stored only on Accelerator (via High Performance Storage Saver)**
  - Cost of data storage on the Accelerator is significantly less than cost of high-end disk systems typically used for Db2 for z/OS data