EFFICIENT TRANSACTION PROCESSING IN SAP HANA DATABASE

THE END OF A COLUMN STORE MYTH

Subhasree Neela (2657462)
Sruthi Kadaveru (2657464)
Problem

- Data Management Scenarios
- Need of specialized systems for specific problems – OLTP and OLAP
- Complexity to link different systems
- High maintenance and error prone
- Observations of motivations for the situation
  - Usage Perspective
  - Cost Awareness
  - Performance
Why SAP HANA Database?

• Distinguishing features of the SAP HANA database for the scope of typical business applications
  o The HANA database comprises a multi-engine query processing environment that offers different data abstractions supporting data of different degrees of structure— from well-structured relational data to irregularly structured data graphs to unstructured text data
  o It supports the representation of application-specific business objects (like OLAP cubes) and logic (domain specific function libraries) directly inside the database engine
  o HANA database is optimized to efficiently communicate between the data management and the application layer
  o Last not least, the SAP HANA database supports the efficient processing of both transactional and analytical workloads on the same physical database leveraging a highly-optimized column-oriented data representation
Architecture

- The SAP HANA database is the core component of the SAP HANA product and takes care of the efficient and flexible data storage and data querying scenarios.

- COLUMN-ORIENTED MAIN-MEMORY CENTRIC

- Follows a strict layered architecture

- Similar to classical systems, the SAP HANA database distinguishes between compile time and run time of a database request.

- SAP HANA database is an extensible platform for different query languages.
Calculation Graph Model

- Types of operators
  - Relational Operators
  - OLAP Operators
  - L Runtimes
  - Text Operators
  - Graph Operators
Life Cycle Management of Database Records

- SAP HANA database system provides life cycle management for an individual database record with a unified table structure.
- The technique of the unified table is not only the key to provide excellent performance for both scan-based aggregation queries but also for highly selective point queries.
- This technique provides a key differentiator to classical (row-based) database architectures.
- While a record conceptually remains at the same location throughout its lifetime in update-in-place-style database systems, the SAP HANA conceptually propagates records through different stages of a physical representation.
• Three stages of record lifecycle
  □ L1 delta
  □ L2 delta and
  □ Main Store

• Since the SAP HANA database was originally designed for OLAP-heavy use-cases with complex and high-volume loading scenarios, the system provides a special treatment for efficient bulk insertions, which may directly go into the L2-delta, bypassing the L1-delta
Merges in Unified Table

- The current SAP HANA database system provides two transformations, called “merge steps“:
  - L1-to-L2-delta Merge
  - L2-delta-to-main Merge

Persistent Storage

- Although the SAP HANA database is a main-memory centric database system, its full ACID support guarantees durability as well as atomicity and recovery in case of a system restart after regular shutdown or system failure.

- In general, no fine-grained UNDO mechanisms with respect to persistent storage are necessary, because only bulk changes like a new version of a main structure are propagated to persistent storage and have to be rolled back in case of a system failure.
From the above figure, the persistency is based on a combination of temporary REDO logs and save pointing for short-term recovery or long-term backup.
Merge Optimization

- **Classic Merge**

  Step 1: Dictionaries of L2-delta are merged with the dictionary of the main to yield the sorted new main dictionary for the specific column.

  Step 2: The new main index is constructed with the positions referring to the new dictionary for existing and newly added entries.

*Figure 7: Details of the L2-delta-to-main merge*
### Partial Merge

- For large tables or partitions, computing a new dictionary and re-generating the main index does have a negative impact on available CPU and disk resources.

- The core idea of the partial merge is to split the main into two (or even more) independent main structures:
  - **Passive main**: The passive main reflects a stable part of the main store which is in general not part of the merge process.
  - **Active main**: The active main is the part of the column which grows/shrinks dynamically and takes part of the merge process with the L2-delta.
Characteristics of SAP HANA Database Record Lifecycle

(a) Workload optimization

- Read optimized
- Write optimized

(b) Memory consumption per record

- High
- Low

(c) Type of record propagation

- Full
- Incremental

(d) Frequency of record propagation

- High
- Low
Conclusion

- Based on the classical column store architecture of the SAP HANA database, we outline the query processing environment showing query transformation, plan generation, and the interaction model of different specialized engines.

- Common unified table data structure consisting of different states on the one hand but providing a common interface to the consuming query engines on the other hand.
THANK YOU 😊