Lecture #19: System Analysis (Snowflake)

15-721 Advanced Database Systems (Spring 2024)
https://15721.courses.cs.cmu.edu/spring2024/
Carnegie Mellon University
Prof. Andy Pavlo

1 Introduction

Snowflake [1] represents the pioneering cloud-native OLAP database in the commercial sphere. It introduces a pay-as-you-go pricing model and exhibits seamless scalability in response to varying workloads.

Topics covered throughout this lecture:

- Data Storage: Snowflake's proprietary data format and utilization of object storage and its caching layer.
- Push-based Vectorized Query Processing: Snowflake's execution engine and work-stealing mechanism.
- Unified Query Optimizer and Adaptive Optimizations: Snowflake's optimizer and its support for complex expressions.

1.1 Architecture

This section explores Snowflake's architectural approach, emphasizing its utilization of cloud object storage, cache layer for latency management, and virtual warehouses for scalable computing.

1.1.1 Data Storage

Rather than developing its proprietary storage layer, Snowflake opts to utilize existing object storage solutions provided by cloud vendors, complementing this with its cache layer to address potential latency issues associated with object storage access.

1.1.2 Virtual Warehouses

Virtual warehouses comprise an arbitrary number of worker nodes (VM instances) running Snowflake software, each equipped with locally attached disks for caching purposes. As of 2022, Snowflake supports serverless deployment, enabling nodes to automatically deactivate during periods of query inactivity.

1.1.3 Cloud Services

Every component is instantiated as a cloud service and is shared among multiple users. These cloud services rely on a transactional key-value store (FoundationDB).

2 Execution Engine

Snowflake employs a push-based vectorized engine that utilizes precompiled primitives for operator kernels. Tuple serialization/deserialization between workers is generated via LLVM for enhanced performance.

Worker processes push data to each other, eliminating the need for a shuffle step between different execution stages.

In the event of failure, the entire query will be rerun, with no provision for partial query retries.

2.1 Work Stealing

The optimizer determines which files workers will retrieve for processing a query prior to execution. If a worker process completes its task ahead of schedule, it attempts to claim work from stragglers. To minimize network traffic on the stragglers, the requester always downloads from object storage.

2.2 Flexible Compute

When a query plan fragment is anticipated to process a large volume of data, the DBMS can temporarily enlist additional worker nodes (potentially sourced from other customers) to enhance its performance. This concept closely resembles the spot instance model offered by AWS. The ephemeral workers consistently write the intermediate results back to the object store to avoid contaminating their cache.

3 Data Storage

Snowflake opts to fully leverage the existing object storage solutions provided by cloud vendors (e.g., AWS, GCP, and Azure) and makes substantial investments in constructing a caching layer to mitigate latencies.

3.1 Data Format

Snowflake primarily stores all tables in its internal columnar format by partitioning them into micropartition files. The system automatically organizes and restructures micropartitions in the background based on query access patterns. Additionally, the DBMS endeavors to infer the format of semi-structured data, segregating them into binary columns upon their initial introduction to the system.

3.2 File Mapping

The DBMS utilizes consistent hashing to assign micropartition files to worker nodes. This ensures that query fragments (tasks) accessing the same micropartition are directed to the same worker nodes. An additional advantage is Snowflake's capability to incorporate new compute nodes seamlessly without necessitating adjustments to micropartition assignments.

3.3 Hybrid Tables

Similar to Dremel and Databricks, Snowflake faces the challenge of lacking statistics for data files created outside of the DBMS. To address this, Snowflake extends its architecture to support alternative methods for data ingestion. The introduction of Hybrid Tables (Unistore) in 2022 enables users to execute OLTP workloads directly within the Snowflake ecosystem. Writes are performed in row-based storage with robust transactional guarantees, while background processes merge them into micropartition files for analytical workloads.

4 Optimzer

Snowflake employs a Unified Cascades-style top-down optimization approach. During query compilation, the optimizer references the catalog to identify micropartitions that can be pruned before query execution.

4.1 Statistics Collection

The DBMS maintains statistics for data stored in Snowflake's proprietary table format. Snowflake exclusively supports simple zone maps, and these statistics remain synchronized with the data when using the internal file format (micropartition).

4.2 Pruning

The optimizer utilizes the zone map to determine which micropartitions to omit. A local cache is retained to expedite evaluation during optimization. Notably, Snowflake supports the evaluation of complex expressions during the pruning process.

4.3 Adaptive Optimizatoin

Upon establishing join ordering, Snowflake's optimizer identifies aggregation operators to be pushed down into the plan beneath joins. Although the optimizer incorporates downstream aggregations, the DBMS only activates them at runtime based on statistics observed during execution.

References

[1] B. Dageville, T. Cruanes, M. Zukowski, V. Antonov, A. Avanes, J. Bock, J. Claybaugh, D. Engovatov, M. Hentschel, J. Huang, A. W. Lee, A. Motivala, A. Q. Munir, S. Pelley, P. Povinec, G. Rahn, S. Triantafyllis, and P. Unterbrunner. The snowflake elastic data warehouse. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, page 215–226, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450335317. doi: 10.1145/2882903.2903741. URL https://doi.org/10.1145/2882903.2903741.