CelerData

StarRocks

The new kid on the real-time analytical databases block

By Sanjeev Mohan

Principal, Sanjmo

September 2022

Introduction

The rise of analytical databases continues unabated, reflecting the growing need to meet ever more demanding analytical use cases. As the demand to support new use cases increase, organizations are looking to modernize their analytical solutions. One cause for the modernization is the shift away from batch reports to decision-making on data as soon as it is produced.

The focus on low-latency and high-concurrency analytical solutions has led to a cacophony of new offerings that tout capabilities, like columnar formats, vectorized processing, OLAP schema, and denormalized tables. Each of these options provides significant performance gains but also introduces tradeoffs in other areas, such as the complexity of schema modifications, or the reduced speed of ingestion of new data.

StarRocks is a new massively parallel processing (MPP) columnar analytical offering that has overcome many of the common challenges by supporting multiple open and proprietary approaches. It originally started, in May 2020, leveraging the framework of Apache Doris. StarRocks has since added its own native kernel comprising the cost-based optimizer and the query execution engine.

Key Takeaways

- StarRocks' native **cost-based optimizer** and **vectorized execution engine** automatically query **denormalized tables**, **star schemas**, and **external tables**. This is unique amongst analytical data stores that use one or the other approach to speed up queries. This enables StarRocks to deliver sub-second responses to thousands of concurrent queries on petabyte-scale data scans.

- StarRocks support of **change data capture** (CDC) allows it to ingest real-time data and perform analytics on it with very low latency and high concurrency. The ingestion approach updates StarRocks data instead of appending changed values, which optimizes read performance.

- StarRocks' **wire-compatibility with MySQL** allows existing MySQL applications to use its analytical solution with no modifications to the code.

- **Fully managed StarRocks Cloud** offering, called **CelerData Cloud**, is currently in Beta and is expected to be Generally Available in Q4 2022.

This research takes a deep dive into the technical aspects of StarRocks.

Introducing StarRocks

When the existing interactive SQL solutions failed to scale to meet the needs of many Chinese service providers, recent developments, such as Apache Kylin in eBay and Project Palo at Baidu were developed. Baidu donated Palo to Apache Software Foundation and renamed it as Doris.

StarRocks originally set out to create an enterprise version of Doris, only to find out that there was significant room for improvement. Since going on its own path, it has developed its own code base which is now 80% different from Apache Doris. Its native source code is open-source and is available on GitHub under Elastic License v2 (ELv2). ELv2 allows users to freely use the product, but they can't provide others with a managed service offering.

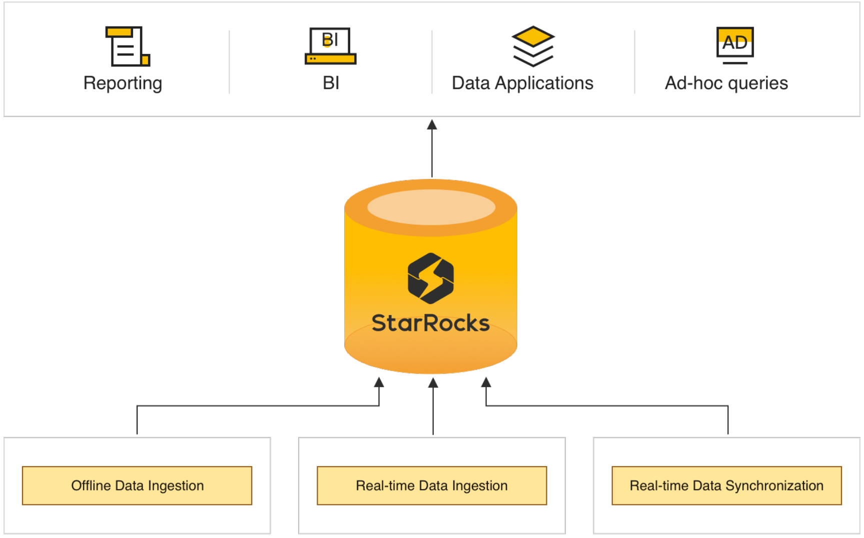

Figure 1: StarRocks Architecture

StarRocks Key Characteristics

On the surface, most analytical tools look alike. However, savvy users need to understand the internals of the solutions to assess their capabilities and align them to their use cases. This section is a deep dive analysis of evaluation criteria to select your next analytical solution and explains how StarRocks meets the criteria requirements.

Data Model / Workloads

Organizations have a myriad types of data intensive workloads - streaming, ETL, BI, ML. When selecting a modern solution, it is important to remember that one size doesn't fit all. Most solutions have grown by specializing in a small range of use cases. This leads to organizations deploying different solutions to meet their disparate needs.

StarRocks, however, has multiple data models that are optimized for different workloads. This is one of its major advantages. For example, traditional data warehouses are only optimized for star schema, while Clickhouse and Druid achieve performance enhancements through denormalizing related tables.

StarRocks supports the following data models:

- **OLAP Star** and **Snowflake schema**. StarRocks is a distributed relational column-oriented data store. It can store data in star or snowflake schema and as pre-aggregations.
- **Denormalized**. Denormalized flat-tables speed up performance are commonly used in many analytical data stores like Apache Druid and Clickhouse. However, they also are complex to update. Hence, they can't be used for all use cases.
- **External tables**. External tables support separation of storage and compute by querying data that resides in other locations such as on a cloud object store, such as Amazon S3 using the data virtualization technique. StarRocks federates queries across MySQL, Elastic and other sources.

Among StarRocks multiple capabilities are the ability to update primary keys, and support for highly efficient multi-table joins. In addition, it analyzes query behavior and recommends materialized views to speed up query performance.

Ingestion

Real-time analytics use cases, like anomaly detection, forecasting demand and promotions require availability of data as soon as it is produced. However, most analytical databases are designed to ingest data in batch, unlike the operational data stores. In the batch mode, consumers do not have access to the latest data.

The second problem with other real-time analytical solutions is that they are also not designed to handle fast changing data. Many of the databases are immutable and they append changed records to the destination. Others perform complex operations to update or delete existing records thereby increasing the data latency.

StarRocks supports real-time ingestion of streaming data in addition to batch data. StarRocks has five types of import loaders to create streaming data lakes - stream, broker, routine, Spark, and 'insert into.' Broker loader is ideal for HDFS under a TB and Spark for large HDFS deployments. The stream loader is used to import tabular files into the database from an external source through an HTTP PUT request. Finally, the routine loader manages the subscription to a pub/sub service like Kafka. Finally, the 'insert into' service allows the MySQL client with typical SQL insert statements.

The data lakes can be built using multiple technologies, depending upon the business requirements, such as:

- **Apache Flink, Spark and Kafka**. The Flink connector ingests incremental data from data sources, like PostgreSQL and MySQL using change data capture (CDC) approach. Using *Apache Hud*i on top of Apache Flink, the ingested raw data can be cleansed, filtered and transformed into the OLAP preprocessing layer. PySpark SDK allows users to write custom code to extract data from various data sources. Kafka is used to import streaming data from the change logs of data sources such as RDBMS like MySQL.
- **External tables.** Data sources external to StarRocks, such as NoSQL databases like Elastic and files on HDFS can be accessed as external tables. StarRocks, thus, eliminates the need to move data from sources like Hive, MySQL, Elastic, or any JDBC source. If the external table source is a filesystem, like Amazon S3 or HDFS, then StarRocks performs the duty of scanning and processing the data.

In addition, StarRocks can perform update and delete operations in real-time maintaining low query latency on the latest data even when the data is frequently updated.

Performance

Performance has risen as one of the most important deciding factors when evaluating new analytical solutions. While ultra low performance has been possible from operational databases, analytical database performance has lagged. Studies have shown that users lose interest when query results take more than 100ms for operational and over 3 seconds for analytical data stores. The holy grail of analytical solutions is to provide sub second latency.

Benchmarks serve as a guidepost, although users typically dismiss them as they perceive that the benchmarked workloads don't represent their use cases. Benchmarks represent the performance of databases under steady and stable conditions, while actual workloads tend to be erratic. The second problem with benchmarks is that they are easy to be "engineered" by the vendors in their favor.

A better criterion is **cost performance**, which optimizes not just performance, but also reduces the cost to achieve high performance. For example, in StarRocks, raw data is processed at a scheduled time into summary data, and then stored in its internal, optimized format, as described earlier.

StarRocks is an MPP analytical database. It splits a query into multiple logical execution units, called Query Fragments on one or more physical execution units. This allows queries to run in parallel on highly scalable multiple nodes. Combined with the vectorized query engine (see below), MPP architecture allows StarRocks to choose the best execution strategies that are optimized for individual CPUs as well as across many MPP nodes.

StarRocks **uses a vectorized query engine** to batch multiple rows in columnar format and iterate over them. This allows queries to use CPUs much more efficiently and speed up reads and writes. CPUs take advantage of **SIMD** (Single Instruction Multiple Data) instructions to execute a single instruction on multiple data. Unlike other vectorized engines, StarRocks engine is fully vectorized which means all operators, functions, imports, and compactions are implemented by vectorization. This eliminates the mismatches and conversions in the data pipeline.

As previously mentioned, StarRocks's **CRUD** (create, read, update and delete) operations are **atomic and happen in-place** at field level. This makes it **mutable** and different from other solutions where new values are instead appended. When you append values with versioning you can't do a predicate pushdown because the executors first have to select which is the most recent version of each data and then do a shuffle operation before query execution. With in-place CRUD, the optimizer can push down predicates with confidence of having the authoritative value without a shuffle step.

StarRocks's **materialized views** are refreshed automatically and in real-time. This allows sub second response time to be achieved even on data that is changing in real-time.

StarRocks' native, brand-new **Cost Based Optimizer** (CBO) is customized according to its full vectorized engine and has made several improvements and innovations to deliver very high performance in multi-table join queries.

Scalability

One of the key defining factors of cloud computing is elasticity, supporting seamless, minimal downtime resource scaling, sometimes up/out, sometimes down/in. This dynamic provisioning strategy exists to meet fluctuating service demand in a cost-effective way. Scalability usually refers to an increase in capacity, and it may not be automated or even elastic.

StarRocks provides scalability across multiple dimensions:

- **Connection scaling**. Support for high concurrency is needed when services experience high peak demand. StarRocks has production deployments that support over 10,000 connection requests per second.

- **Compute scaling**. StarRocks has demonstrated linear scalability. When a new node is added to the cluster, data is redistributed behind the scene automatically with a minimum impact on the user experience. This operation is so effective that most users don't even notice that data rebalancing is happening. Isolating the resources also helps in reducing noisy neighbor issues.
- **Storage scaling**. Keeping up with StarRocks' ethos of providing optionality, it supports two types of sharding - hash-based and range partitioning. The analytical engine automatically shards data using a hash key across all the backend nodes (BE nodes). Users can create their own partitions on top of the shards.

StarRocks scaling is near-instant with no downtime. It is in the process of adding auto-scaling capabilities.

Ease of Use

IT departments and the end users want the infinite elastic scale of modern data stores and the semantics of RDBMS, respectively. These seemingly contradictory requirements were the hallmark of initial NoSQL databases. High user experience of an analytical solution translates into three areas:

- **Developer experience**. Data engineers want the process of building and maintaining pipelines to be easy, repeatable and debuggable. Complexity is the enemy of productivity.
- **Operational experience**. Administrators want the ease of reliability, upgrades, patching and performance tuning, among other operational tasks.
- **Customer experience**. End users want to continue using the front-end tools of their choice, or the SQL statements that they have built over the years. They want their existing BI tools supporting standard SQL syntax to work out of the box on the new analytical data store.

StarRocks not only provides SQL semantics but also MySQL compatibility at SQL query and client protocol levels.

In addition, as it doesn't need data to be denormalized, it significantly reduces the complexity involved in building pipelines. This approach removes the need to maintain multiple copies of data, which also reduces the number of locations where data needs to be secured. Another benefit, this approach enables data to be queried in real-time by removing the need to prepare data for consumption.

StarRocks simplified architecture helps improve system stability and lower operational cost.

StarRocks Enterprise Edition and SaaS cloud offering provide a management UI to help with auto deployment and cluster tuning. Notebook integration can be done using any MySQL client library. Finally, it provides APIs for integration.

Resilience

As analytical workloads become increasingly mission critical, the underlying data stores are expected to be high-available, fault-tolerant. StarRocks is designed with no single point of failure. When a single node fails, data can be automatically migrated without affecting overall availability. Its sharding approach provides variable level replica number configuration, automatic data balancing, and replica repair.

StarRocks provides automatic backups on Amazon S3-compatible object stores or on HDFS. The data on object stores can also be accessed through a direct query.

Unplanned downtime is one of the biggest causes of concern to admins and it leads to poor user experience. StarRocks currently provides three 9's (99.9%) availability for the self-managed version and higher for its SaaS cloud version, which will be generally available in Q4 2022.

Security

Key data security goals of all data stores should be to secure all sensitive data, set up authorizations and enable encryption.

StarRocks supports password authentication or LDAP based authentication. Its permission management system supports fine-grained permission control at table level, role-based access control (RBAC), and whitelisting. This allows StarRocks users to secure sensitive data and ensure authorizations will allow only the right consumer to see the right data.

Lifecycle Management

Robust lifecycle management should address how easily and quickly applications can be built, tested, and delivered. The goal is to accelerate the release cycle, including A/B testing. Typically, products integrate with other aspects of the architecture, such as the ETL engine, test management, and application development platform.

Key lifecycle management features include automated patching and upgrades. StarRocks features include rolling upgrades and automated patching. As mentioned earlier, it supports backing up data and metadata to a remote storage system, such as an object store like Amazon S3. The backup can be restored to a cluster at any time.

A StarRocks Case Study

Consumers, globally, know Airbnb for its community-based online marketplace for lodging. But, technologists also know Airbnb for its many open source contributions, such as Airflow for data orchestration and Superset for data visualization. In fact, Airbnb has an impressive 194 repositories on GitHub reflecting its deep open source roots. One of the newer open source projects is a metrics layer called Minerva. Airbnb uses Minerva to store 12,000 metrics and 4,000 dimensions.

Minerva consolidates the creation and serving of business metrics into a single source of truth platform that is used to derive consistent cross-dataset insights. Besides the consistency benefits, Minerva improves the scalability and reliability of its analytics. It serves over 5,000 data sets and hundreds of concurrent users. Data from sources are ingested, cleansed, enriched, and analyzed constantly. This data is highly dynamic in nature and evolves rapidly.

Minerva's requirements included near real-time data freshness, low query latency, support for complex SQL queries, and low overall cost. It started using Apache Druid and Presto for its analytics. As its needs increased, Druid could not handle complex SQL statements and Presto query latency was higher than its needs. StarRocks's streamlined architecture that doesn't require denormalized tables to be created, allows fast changing data to be analyzed on the fly. Airbnb could use its existing Tableau BI tool directly on the data as StarRocks is fully compatible with MySQL and offers full SQL support. Many queries that took ten minutes to run were able to complete within seconds.

In addition to Minerva, Airbnb is also using StarRocks for real-time fraud detection and improving BI Dashboard performance. For more details on the Airbnb case study, please read the whole story here: https://celerdata.com/hubfs/Airbnb_Case_Study.pdf.

Summary

StarRocks has ignited the analytical query world by providing a low latency MPP platform to query dynamically changing data. Being a mutable database, the changes to source data are written in real-time to StarRocks, which further optimizes read performance.

Most distributed columnar relational data warehouses are designed to query a pre-defined and optimized OLAP data model or a flattened table. StarRocks instead supports multiple data models. Its support for batch and streaming workloads is a unique differentiator. Finally, it is compatible with MySQL protocol and supports all major BI products.

About CelerData

CelerData enables enterprises to quickly and easily grow their business with a real-time analytical engine that is 3X the performance/cost of any other solutions on the market. CelerData is the only platform uniquely designed for the next generation real-time Enterprise, unleashing the power of business intelligence to help accelerate Enterprise digital transformation. Used worldwide by market leading brands including Airbnb, Lenovo and Trip.com, CelerData generates critical new insights for these data-driven companies. To learn more, please visit, www.celerdata.com