CelerData

CASE STUDY

AIRBNB BUILDS A NEW GENERATION OF FAST ANALYTICS EXPERIENCE WITH STARROCKS

Introduction

Airbnb is a world-famous community-based online marketplace for lodging. It provides a platform for hosts to share accommodations with travelers around the world. There are over 6M active listings available on Airbnb platform today, 4M+ hosts provide listings on Airbnb, and over 1B+ Airbnb guest arrivals all-time (as of Sept. 28 2021).

How to efficiently use data to power business at such a large scale has always been a challenge facing the Airbnb Data Infrastructure team. As the requirements for query latency and data freshness demand grows, the legacy data architecture is inadequate to meet new data analytics requirements. StarRocks, a new generation distributed OLAP solution that delivers real-time analytics, came to the rescue.

On March 15, Jingwei Lu, software engineer at the Airbnb Data Infrastructure team, gave a live broadcast on how Airbnb uses StarRocks to power real-time analytics in three typical business scenarios: Tableau dashboard, metrics store (the Minerva project), and trust analytics.

See the full video of the meetup: https://www.youtube.com/watch?v=AzDxEZuMBwM.

Pain Points of Airbnb

Similar to other tech companies, Airbnb's data analytics architecture consists of four layers: data layer, data warehouse layer, OLAP layer, and user layer. The OLAP layer is nearest to the user layer and therefore, must be flexible and powerful enough to meet demanding data analytics needs.

Previously, Airbnb used Apache Druid and Presto as the OLAP solution. However, the two are not satisfactory in query latency and data freshness. Query latency refers to how soon a user can obtain the query result, and data freshness refers to how soon the data can land at the OLAP layer.

Query latency and data freshness of an OLAP system directly affects business logic at the user layer. StarRocks' excellent capability in these two aspects unfold more possibilities for the Airbnb Data Infrastructure team. In this meetup, Jingwei Lu explains the challenges facing Airbnb in three typical analytics scenarios and the value StarRocks brings to Airbnb.

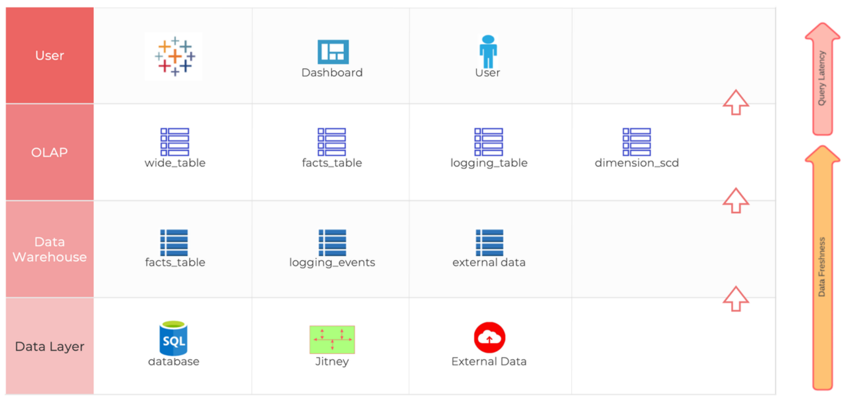
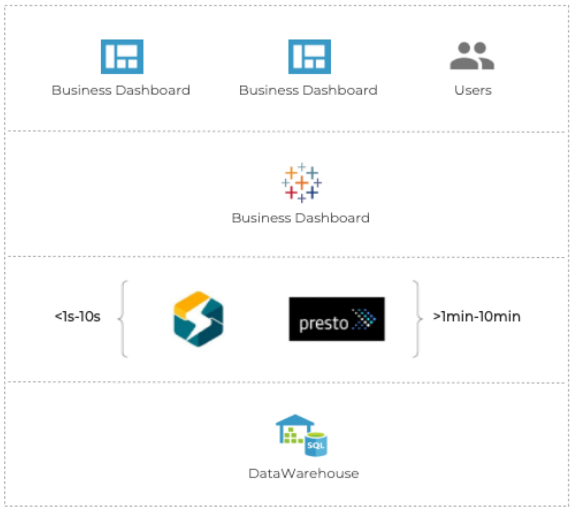


Tableau Dashboard

Tableau is a simple, powerful BI tool for data visualization. It is popular among business personnel at all levels. Airbnb has to deal with two challenges if they want to use Tableau. First, Tableau automatically generates complex SQL queries when business personnel perform drag-and-drop operations to generate BI reports.

The OLAP layer must be powerful enough to process and optimize such complex SQL. **Apache Druid cannot meet this requirement and therefore, cannot directly connect to Tableau**. Second, data records generated in BI scenarios can reach hundreds of millions, even hundreds of billions, which is beyond the processing capabilities of MySQL and Tableau's native storage.

Airbnb also tried Presto, only finding that the latency of complex queries in some cases exceeds 10 minutes. Such long latency is unbearable for users. It undermines interactive experience and renders data analytics meaningless.



All these challenges are addressed after StarRocks is introduced.

- StarRocks has an excellent query optimizer, which can efficiently process complex SQL queries generated by Tableau and work out an optimal query plan.
- StarRocks offers blazing-fast query speed. It can return results within seconds even for queries against hundreds of millions or even hundreds of billions of data records, **reducing the response time for complex queries from more than 10 minutes to seconds**.
- StarRocks is also compatible with the MySQL protocol and supports ANSI SQL. It has 98% compatibility with the SQL syntax generated by Tableau, which ensures easy connection with Tableau.

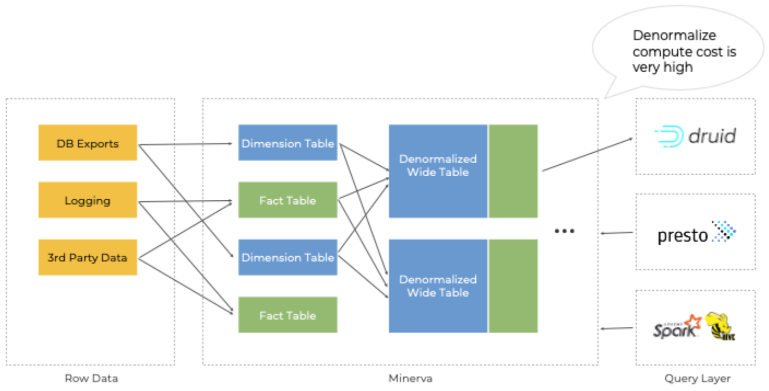
StarRocks makes it possible for **Airbnb to deliver interactive analytics experience for Tableau users in the case of huge data volume.**

Minerva Metrics Store

Minerva is a unified metrics management platform within Airbnb. It is designed with the vision to allow users to "define metrics once, use them everywhere". Minerva stores over 12,000 metrics and 4,000 dimensions. It is positioned for various data consumption scenarios, such as A/B testing, data exploration, and data analysis. For more information about Minerva, see this blog post:

https://medium.com/airbnb-engineering/airbnb-metric-computation-with-minerva-part-2-9afe6695b486.

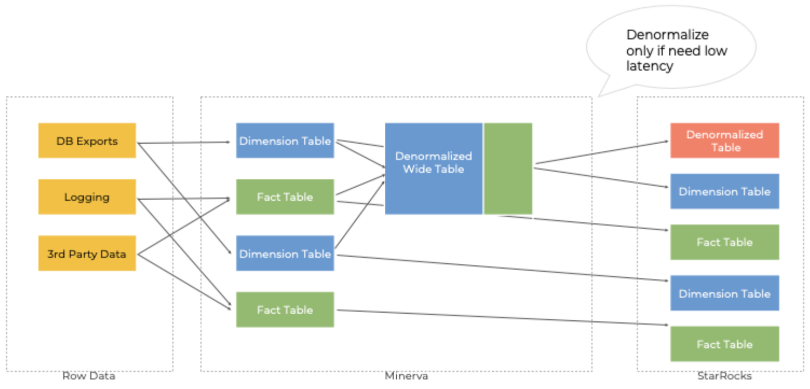
Previously, to accelerate queries, engineers have to denormalize data in Minerva and then use Apache Druid, Presto, Apache Spark, or Apache Hive to process queries.



As data volume surges, such an architecture will cause huge resource consumption due to the following reasons:

- Engineers must denormalize tens of thousands of metrics every day, even on a large number of metrics that will not be queried, which consumes lots of computing resources.
- Each time a single dimension changes, all the historical data in the previous few years must be calculated again, which not only wastes computing resources, but also undermines service stability.
- Druid is not good at aggregating high-cardinality fields. Engineers have to slice a large dimension table into several small tables, which involves complex operations and wastes computing and storage resources.

Airbnb is in urgent need of an OLAP system that can break the dilemma and allow for more flexible modeling.



After thorough and in-depth testing, Airbnb found that StarRocks works well with various data models:

- In wide-table scenarios, StarRocks can deliver sub-second query performance.
- In multi-table scenarios, StarRocks can return results within seconds, even for on-the-fly joins of largest tables.

With StarRocks, what Airbnb needs to do is pick a data model that best suits their business needs. StarRocks will take care of the rest.

- For common queries, Airbnb can leverage StarRocks' multi-table join capabilities for on-the-fly joins.
- Only for some rare queries that require sub-second response, Airbnb engineers need to denormalize data into a flat table and store the table in StarRocks for query.

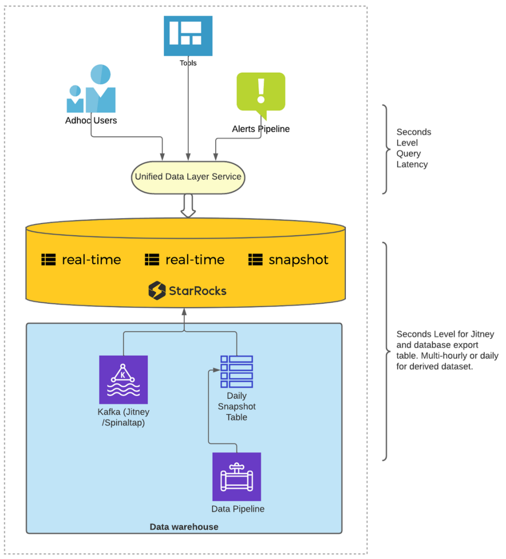
This frees Minerva engineers from tedious and complex denormalization tasks. When a metric is updated, there is no need to backfill data or rebuild tables, saving a lot of resources.

Trust Analytics

To ensure the legitimate rights and interests of users and the platform, data scientists at Airbnb need to identify violations on the platform in a timely manner to prevent possible losses. This puts forward new requirements for the OLAP system:

- **Real-time update**: Business data needs to be obtained at the earliest opportunity and be merged with historical data to achieve real-time data landing and query.
- **Complex queries**: Due to the uncertainty of violations, the dataset to query and the query method cannot be predicted. Therefore, the OLAP system must be able to run complex ad-hoc queries on real-time updated data and return results within seconds.

Druid and Presto cannot meet these two requirements.


Tools
Adhoc Users
Alerts Pipeline
Unified Data Layer Service
real-time
real-time
snapshot
StarRocks
Kafka (Jitney /Spinaltap)
Daily Snapshot Table
Data Pipeline
Data warehouse
Seconds Level Query Latency
Seconds Level for Jitney and database export table. Multi-hourly or daily for derived dataset.

Airbnb built a new Trust Analytics architecture on top of StarRocks. In this architecture:

- Kafka logging events and data tables in the data warehouse can be ingested into StarRocks in real time. StarRocks' primary key model can be used to implement real-time data updates and ensure data freshness.
- On top of StarRocks, users can run various types of complex queries to gain data insights within seconds, ensuring flexible and real-time experience.
- The daily alerting function is added. The system only needs to run SQL queries at scheduled time.

StarRocks shortens the time required for trust analytics from days to seconds, enabling real-time trust analytics.

New Generation of Data Analytics: Low Query Latency + High Data Freshness

StarRocks is well suited for Airbnb's query latency and data freshness requirements, unfolding more business possibilities for Airbnb.

Low Query Latency

Low query latency is a shared pursuit of all OLAP engines. Airbnb is looking for an OLAP engine that is highly performant in both wide-table and multi-table complex queries. With the guarantee of low latency, business personnel have the freedom to choose between data models and cultivate more analytics scenarios.

Specifically, Airbnb has three types of queries:

- Filtering or aggregation on wide tables, such as GROUP BY on multiple columns or filtering on multiple columns using the WHERE clause. Data tables may have lots of high-cardinality fields. **For such queries, Airbnb wants sub-second query latency.**
- Point lookup, such as querying detailed data by ID. **Such queries are impossible in systems that use pre-aggregation, such as Druid.**
- Complex queries, such as joins between multiple large tables and joins between large tables and multiple small tables. This type of query may also involve window functions and various data types. **For such queries, Airbnb hopes to achieve second-level return against datasets with billions of records.**

Complex queries are usually most flexible and unpredictable, which places high requirements for the **functions** and **performance** of an OLAP engine. StarRocks fits these requirements well and closes gaps in data analytics.

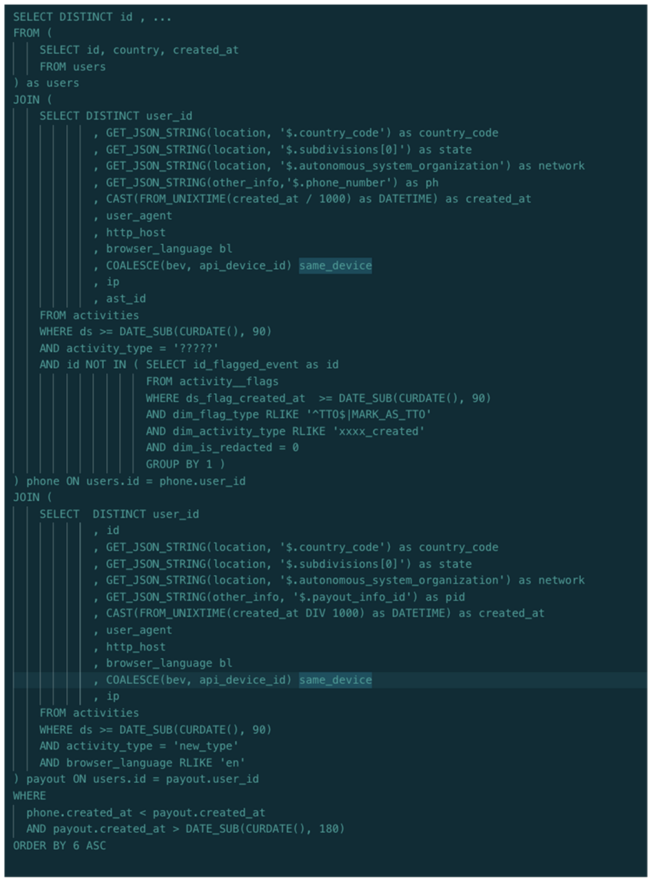
At function level:

- StarRocks supports all types of joins. It also offers collocated join, bucket shuffle join, and broadcast joins to optimize joins of large tables.
- StarRocks offers rich functions and supports window functions.
- StarRocks supports federated queries for Hive, MySQL, and Elasticsearch, broadening data analytics scenarios.

At performance level, query plan optimization is equally important as query execution speed. An excellent query plan can make perfect arrangements for complex SQL statements, including those generated by the BI system. Ultra-fast query execution is the cornerstone behind flexible data modeling.

With such performance guarantee, data scientists and analysts can focus more on business analytics. **StarRocks' cost-based optimizer (CBO) and vectorized execution engine are two important reasons that drive Airbnb to choose StarRocks.**

Following is a query case of Airbnb. In this case, a StarRocks cluster built on 20 AWS EC2 i3.8xlarge instances is used (32 vCPUs, 240 GiB of memory, 3 TB SSD); the query involves three tables (one with 500 million rows, one 6 billion rows, and one 100 million rows), 4 joins, 3 distinct counts, and JSON and REGEX functions. StarRocks uses its CBO to reorder joins and push down operators. It takes only **3.6s for StarRocks to return the result**, **whereas in Presto, the query may take as long as 3-4 minutes**.



```sql
SELECT DISTINCT id , ...
FROM (
    SELECT id, country, created_at
    FROM users
) as users
JOIN (
    SELECT DISTINCT user_id
            , GET_JSON_STRING(location, '$.country_code') as country_code
            , GET_JSON_STRING(location, '$.subdivisions[0]') as state
            , GET_JSON_STRING(location, '$.autonomous_system_organization') as network
            , GET_JSON_STRING(other_info,'$.phone_number') as ph
            , CAST(FROM_UNIXTIME(created_at / 1000) as DATETIME) as created_at
            , user_agent
            , http_host
            , browser_language bl
            , COALESCE(bev, api_device_id) same_device
            , ip
            , ast_id
    FROM activities
    WHERE ds >= DATE_SUB(CURDATE(), 90)
    AND activity_type = '?????'
    AND id NOT IN ( SELECT id_flagged_event as id
                    FROM activity__flags
                    WHERE ds_flag_created_at  >= DATE_SUB(CURDATE(), 90)
                    AND dim_flag_type RLIKE '^TTO$|MARK_AS_TTO'
                    AND dim_activity_type RLIKE 'xxxx_created'
                    AND dim_is_redacted = 0
                    GROUP BY 1 )
) phone ON users.id = phone.user_id
JOIN (
    SELECT  DISTINCT user_id
            , id
            , GET_JSON_STRING(location, '$.country_code') as country_code
            , GET_JSON_STRING(location, '$.subdivisions[0]') as state
            , GET_JSON_STRING(location, '$.autonomous_system_organization') as network
            , GET_JSON_STRING(other_info, '$.payout_info_id') as pid
            , CAST(FROM_UNIXTIME(created_at DIV 1000) as DATETIME) as created_at
            , user_agent
            , http_host
            , browser_language bl
            , COALESCE(bev, api_device_id) same_device
            , ip
    FROM activities
    WHERE ds >= DATE_SUB(CURDATE(), 90)
    AND activity_type = 'new_type'
    AND browser_language RLIKE 'en'
) payout ON users.id = payout.user_id
WHERE
  phone.created_at < payout.created_at
  AND payout.created_at > DATE_SUB(CURDATE(), 180)
ORDER BY 6 ASC
```

High Data Freshness

Data freshness is critical to data analytics. Early data acquisition and analysis mean early opportunity capturing, instant problem solving, and timely actions. Data freshness hinges on how quickly data is imported and how quickly data updates take effect.

StarRocks provides a reliable and scalable import solution that offers the following features:

- High-performance, scalable batch data import. This enables Airbnb to quickly synchronize historical data and daily snapshots in the data warehouse into the OLAP layer. Users can query data at the BI layer in real time. With StarRocks, TB-scale data can be imported within 1 hour.
- Transactional import, which ensures exactly-once semantics and simplifies O&M.
- Real-time import. Data takes effect immediately after being imported, achieving second-level freshness.

In terms of **data update**, the data update cycle of traditional data models is excessively long. Data lake products such as Apache Iceberg, Hudi, and Delta Lake also support data updates but the update cycle may take several minutes. StarRocks offers two data models that deliver dramatic performance uplift:

- Primary key model: This model updates data based on the primary key. Users can achieve real-time replacement of the entire row by performing DELETE + INSERT.
- Aggregate key model: This model aggregates index columns based on dimension columns when data is being imported. This reduces the amount of data to scan and process, accelerating queries.

A Look Ahead

StarRocks' features and performance, especially real-time updates and efficient processing of complex queries help Airbnb discover new real-time analytics scenarios, eliminate the limitations of the flat-table schema, and make data analytics accessible to more front-line business personnel.

Further, StarRocks high-concurrency analytics capabilities are much better suited for our user-facing analytics sceneries (vs. our limited flexibility solution that relies on both Elasticsearch and a KV store). Finally, StarRock's powerful data lake analytics will accelerate our data warehouse's Hive-based interactive query model and gradually replace many if not all of our Presto workloads.

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