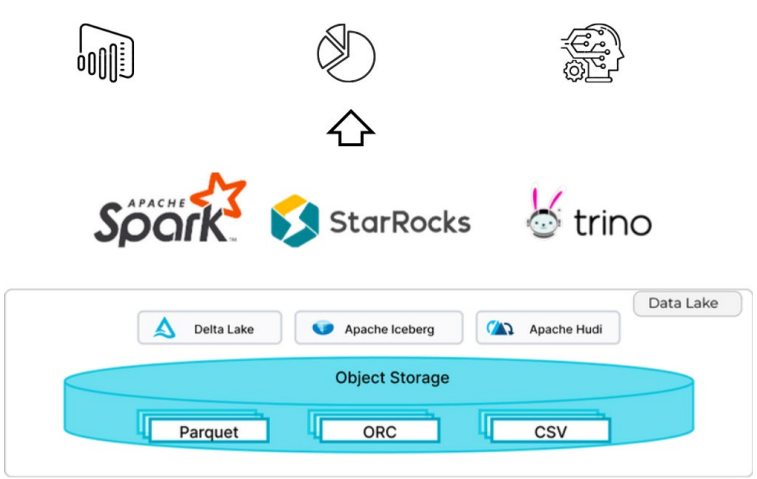
DATA+AI SUMMIT BY databricks

DATA WAREHOUSE PERFORMANCE ON THE DATA LAKEHOUSE

Sida Shen, Product Manager, CelerData

Eric Sun, Head of Data Platform, Coinbase

Data Lakehouse



- ACID transaction properties
- Schema evolution
- Compaction
- Near-real-time analytics
- SQL

- Data warehouse on open & **standardized** storage
- Unify batch and near-real-time workloads on single source of truth data
- Easy data governance, simple architecture, flexibility, cost-effectiveness

THE REALITY?

THE REALITY

Users are forced to copy data out of the lakehouse

Query engines are not fast enough

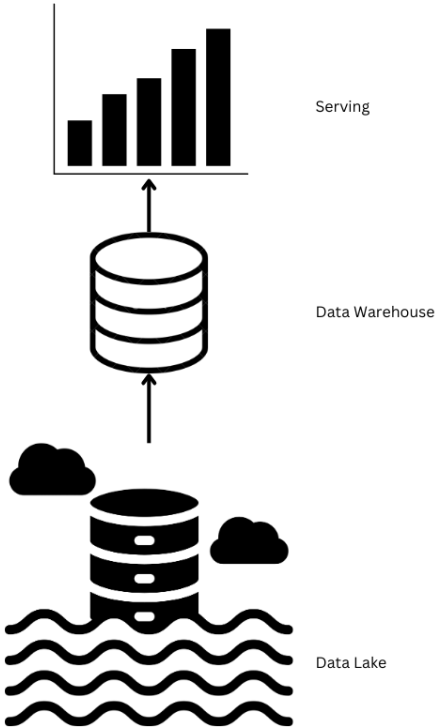
- Not optimized for high concurrency low latency workloads
- Still on older technologies
- Not able to handle demanding analytics workloads such as customer-facing analytics

The users turn to costly workarounds

- Over-engineering or overspending on their existing query engine for barely passable performance. Unsustainable nor future proof.
- Forced to move workloads to a proprietary data warehouse purely for query acceleration

THE COST

Stop moving your workload and data just for better query performance



- Cost of maintaining a proprietary data warehouse
- Cost of data ingestion
- Challenges from matching schema, data type, SQL, etc.
- Data governance challenges from duplicating the data

WHAT ARE THE CHALLENGES?

CHALLENGES

From existing query engines

Query engines used for the data lakehouse today are not built for data warehouse workloads, some of them are:

- Optimized for long-running batch workloads, not running low-latency high-concurrency queries.
- Not optimized for query performance.

CHALLENGES

From querying an external storage system like a data lake

Fetching data/metadata can easily become the bottleneck:

- Network IO overhead
- Slow and unpredictable performance of data lake storage devices
- Data/metadata files can be unoptimized for query performance

ACCELERATING QUERY PERFORMANCE

- A hierarchical caching framework is necessary
  - Overcomes the unstable performance of data lakes
  - Saves IO costs from fetching data from external storage
- MPP in-memory data shuffling
  - Optimized for low latency instead of batch workloads
- System level optimizations
  - Find a C++ query engine to fully utilize SIMD instruction sets

DATA WAREHOUSE PERFORMANCE ON THE DATA LAKEHOUSE IS EASY!

WHAT IS STARROCKS?

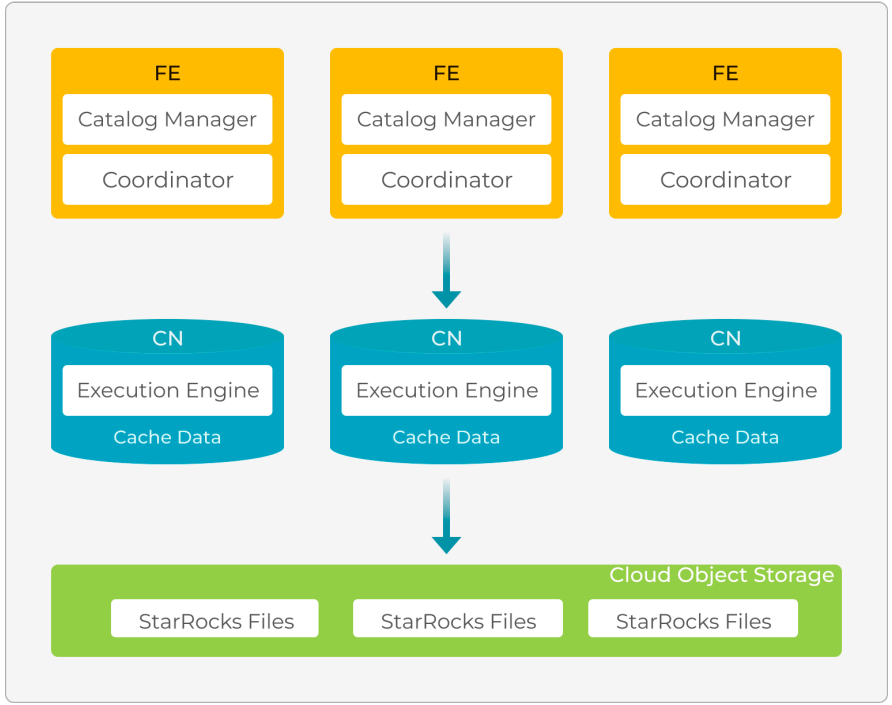
Data warehouse performance on the lakehouse

- Linux Foundation open-source Lakehouse query engine
- Newer generation query engine
  - MPP architecture
  - C++ SIMD optimized
  - Sub-second query latency with high concurrency
  - Run demanding data warehouse workloads on the data lake

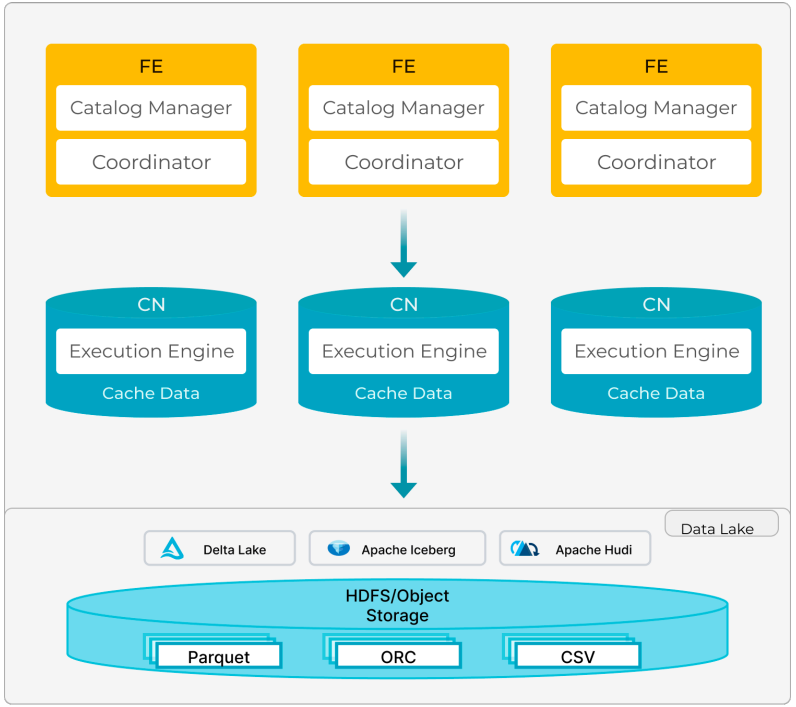
DATA WAREHOUSE VS DATA LAKEHOUSE

The performance difference between data warehouse and data lakehouse

StarRocks as a Data Warehouse

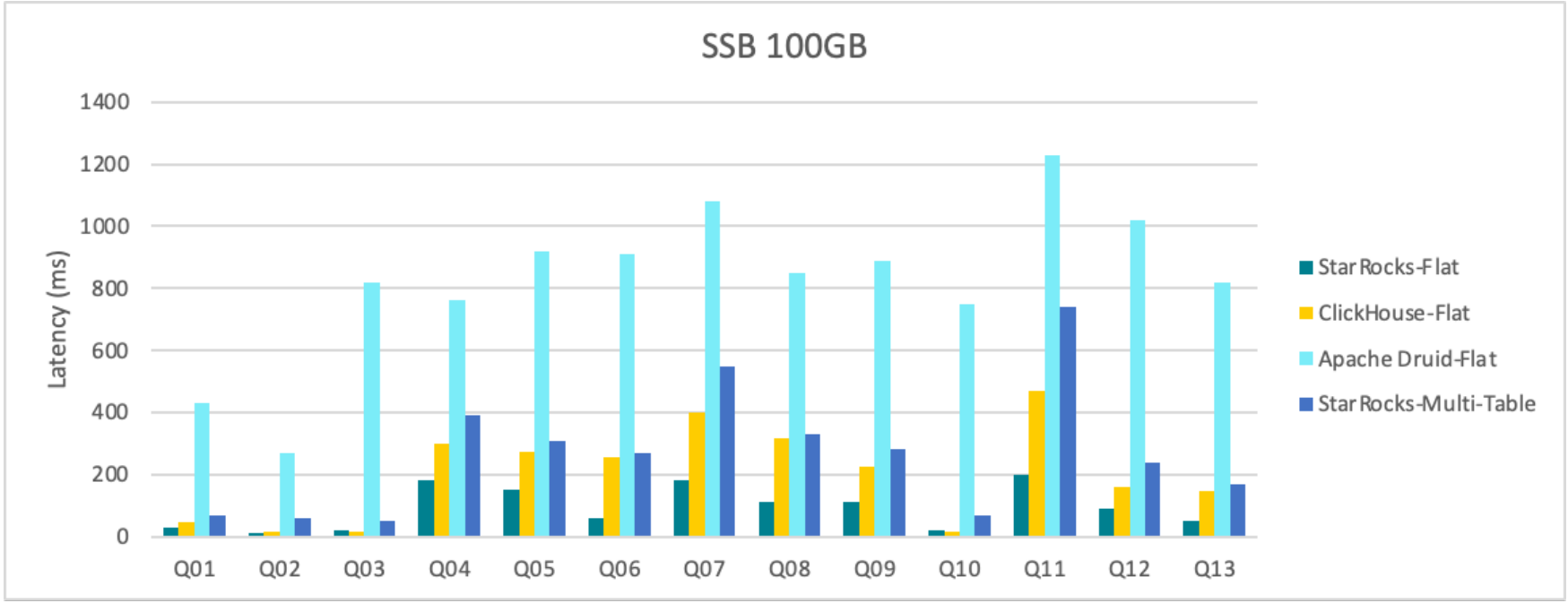


StarRocks On Open Data Lake



SETTING THE BASELINE

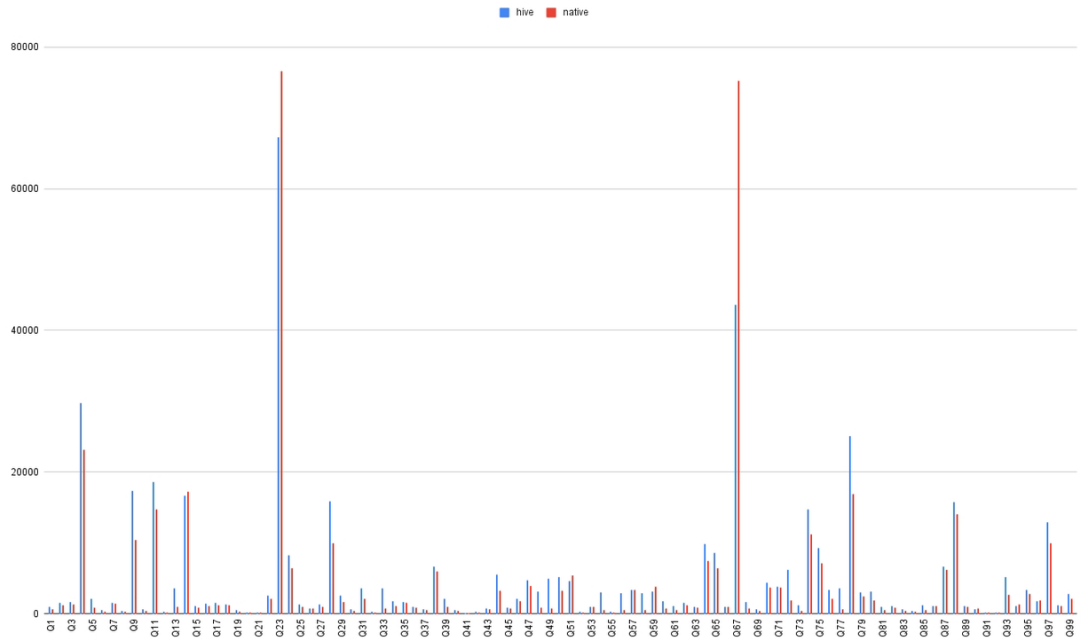
StarRocks is a highly performant data warehouse



DATA WAREHOUSE VS DATA LAKEHOUSE

- Same hardware
- Hot queries
- TPC-DS 1TB Benchmark

The data warehouse solution is only 12% faster



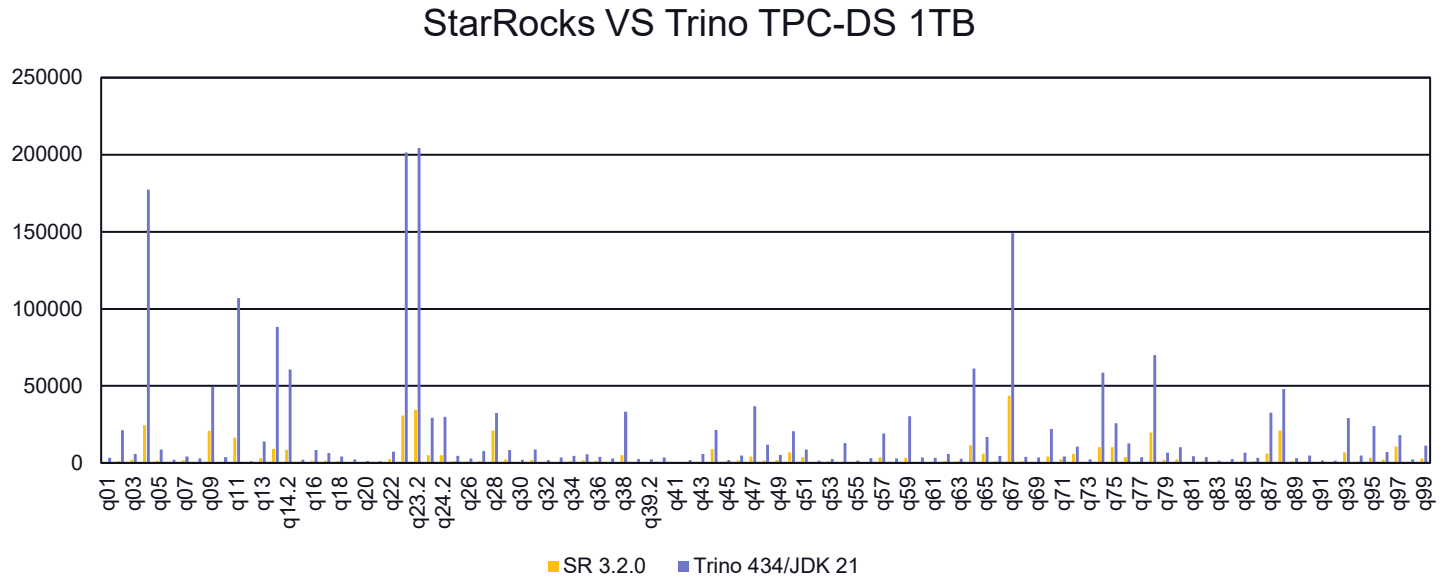
COMPARING TO OTHER QUERY ENGINES

What is Trino?

- Open-source distributed SQL query engine designed to query large data sets distributed over heterogeneous data sources
- Written in Java, forked from Presto, great at connecting to different data sources
- Trino was a game changer for data lakes
  - MPP architecture
  - Way faster than Map Reduce
  - Hours -> minutes query latency on massive amounts of data

COMPARING STARROCKS TO TRINO

How fast is a purpose-built query engine for the lakehouse?

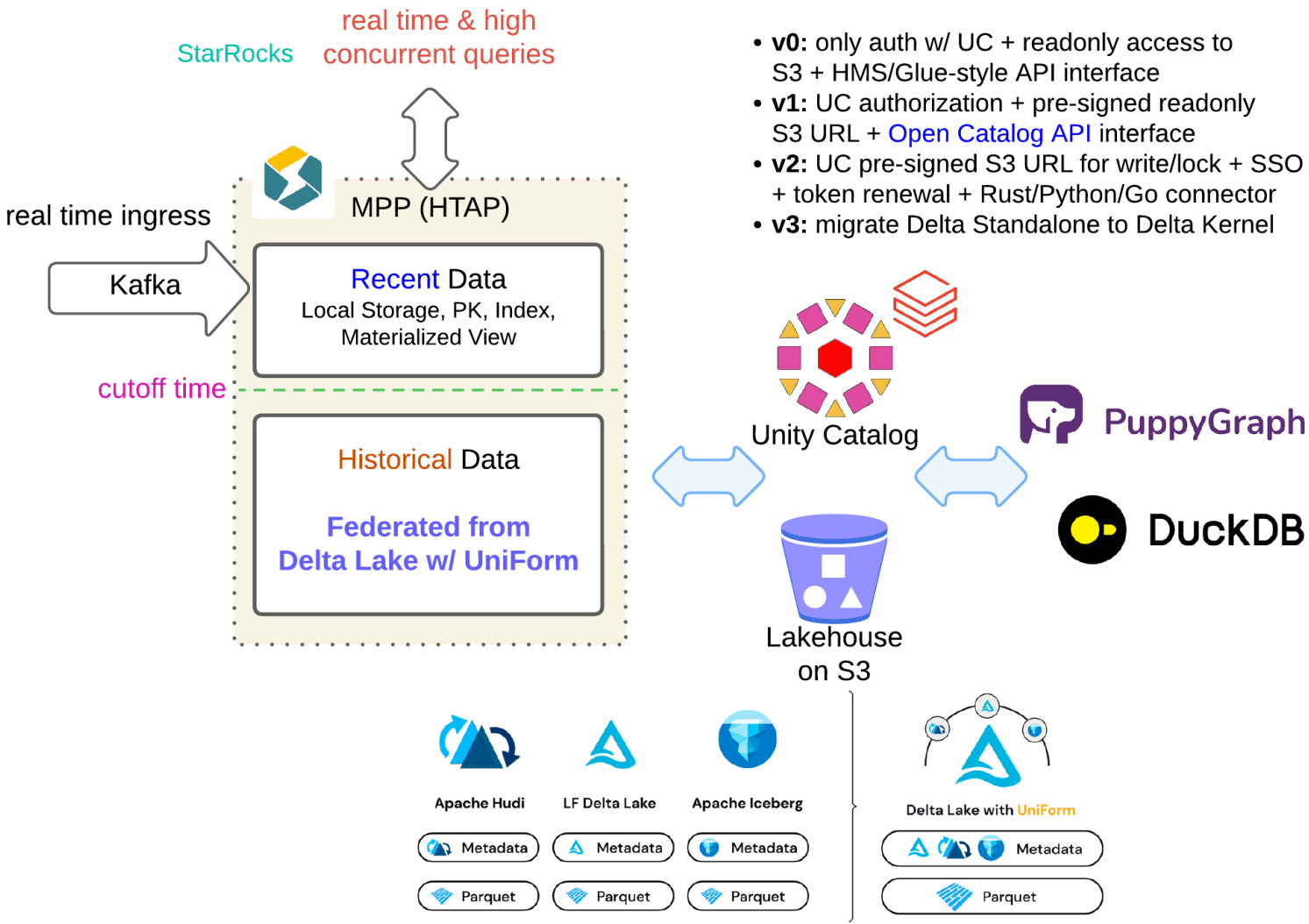


StarRocks is 4.62 times faster than Trino on TPC-DS 1TB

How can we apply these new technologies today?

DATA LAKE WITH OPEN FORMAT, UNITY CATALOG, AND MULTIPLE QUERY ENGINES

Eric Sun, Head of Data Platform, Coinbase



- **v0:** only auth w/ UC + readonly access to S3 + HMS/Glue-style API interface
- **v1:** UC authorization + pre-signed readonly S3 URL + Open Catalog API interface
- **v2:** UC pre-signed S3 URL for write/lock + SSO + token renewal + Rust/Python/Go connector
- **v3:** migrate Delta Standalone to Delta Kernel

DATA+AI SUMMIT

- Visit **CelerData** at **Booth #75**
- Join the StarRocks Slack community
- StarRocks.io
- CelerData.com