Trino Fest

branch

"Maximizing Cost Efficiency in Data Analytics with Trino and Iceberg"

Gopi Bhagavathula

branch

Intro

In today’s fast-paced world, businesses are increasingly reliant on data to drive decision-making, but the cost of maintaining high-performance data infrastructure can quickly spiral out of control.

At Branch, we realized that our existing architecture, was not only expensive but also becoming unsustainable as data volumes grew for one of our business units.

Intro (continue...)

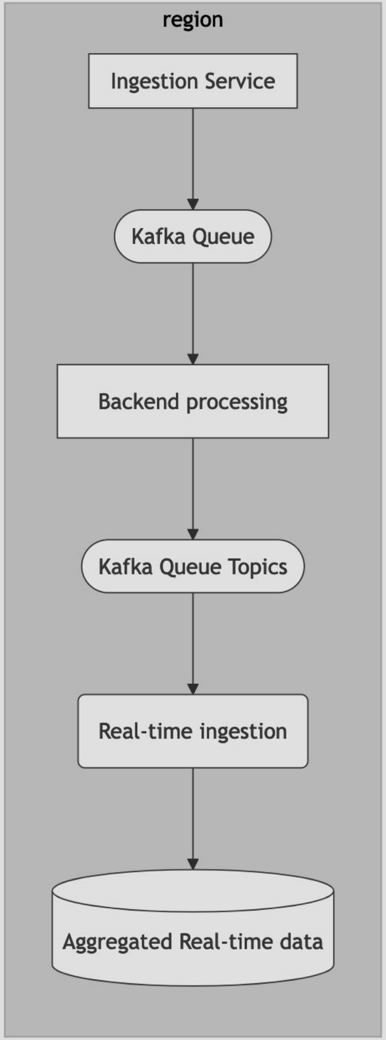
Faced with rising operational costs, we made a bold decision to revamp our internal data analytics by adopting **Trino** and Apache Iceberg. This transition allowed us to cut down on significant overhead, improve scalability, and still maintain a high level of performance—even faster than we initially expected.

With a strategic trade-off of sacrificing some real-time capabilities for cost efficiency and slight latency in query performance, we gained a streamlined, robust infrastructure that continues to support our analytical needs without breaking the bank.

Intro (continue…)

Our journey of migrating from real-time analytics to Trino and Iceberg taught us that the right combination of tools can transform data analytics for one of our internal business units, offering the perfect balance between **cost savings, performance, and scalability**. We achieved 7-figure savings with a few “compromises”.

Architecture Before



region

Ingestion Service

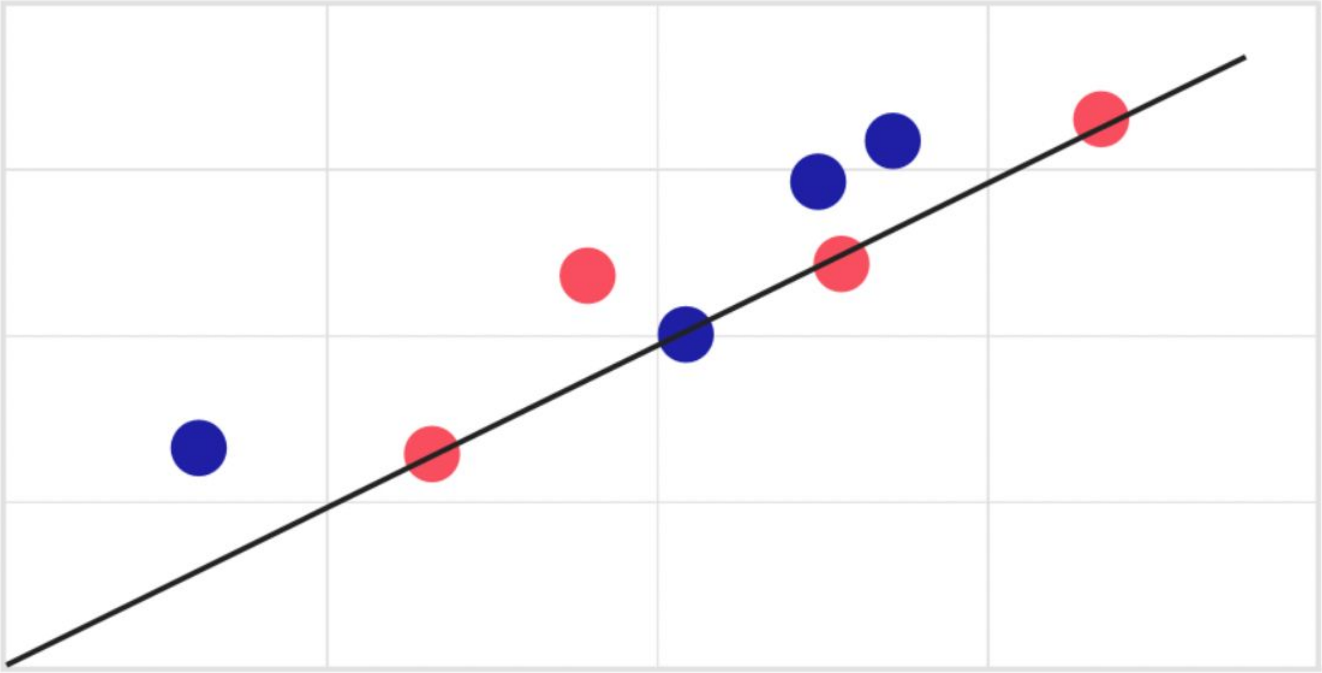
Kafka Queue

Backend processing

Kafka Queue Topics

Real-time ingestion

Aggregated Real-time data



The Challenge: Soaring Costs and Scalability Bottlenecks

Our previous architecture was built using real time for our data warehouse, which had served us well for some time. However, as data volumes grew, **operational and infrastructure costs skyrocketed**. Despite the query engine's ability to deliver sub-second query performance on real-time data, the price tag associated with maintaining this speed and scale became unsustainable.

We were spending significant amounts on:

- High infrastructure costs for scaling real time data clusters.
- Maintaining complex ingestion pipelines for real-time data.
- Additional compute and storage resources to maintain the required query performance.

Faced with this challenge, we began exploring alternatives that could provide a similar experience in terms of querying and processing, but at a lower cost. That's when we decided to look into **Trino** (formerly known as PrestoSQL) and Apache Iceberg as a potential replacement.

The Pivot: Switching from real time QE to Trino + Iceberg

The decision to move from real time query engine to a **Trino + Apache Iceberg** setup came after evaluating multiple modern data processing and storage solutions. Trino, a fast distributed SQL query engine, gave us the flexibility to query data across disparate sources. Iceberg, a table format for large analytic datasets, enabled us to manage large datasets efficiently in our data lake.

The Pivot: (continue…)

Why Trino + Iceberg?

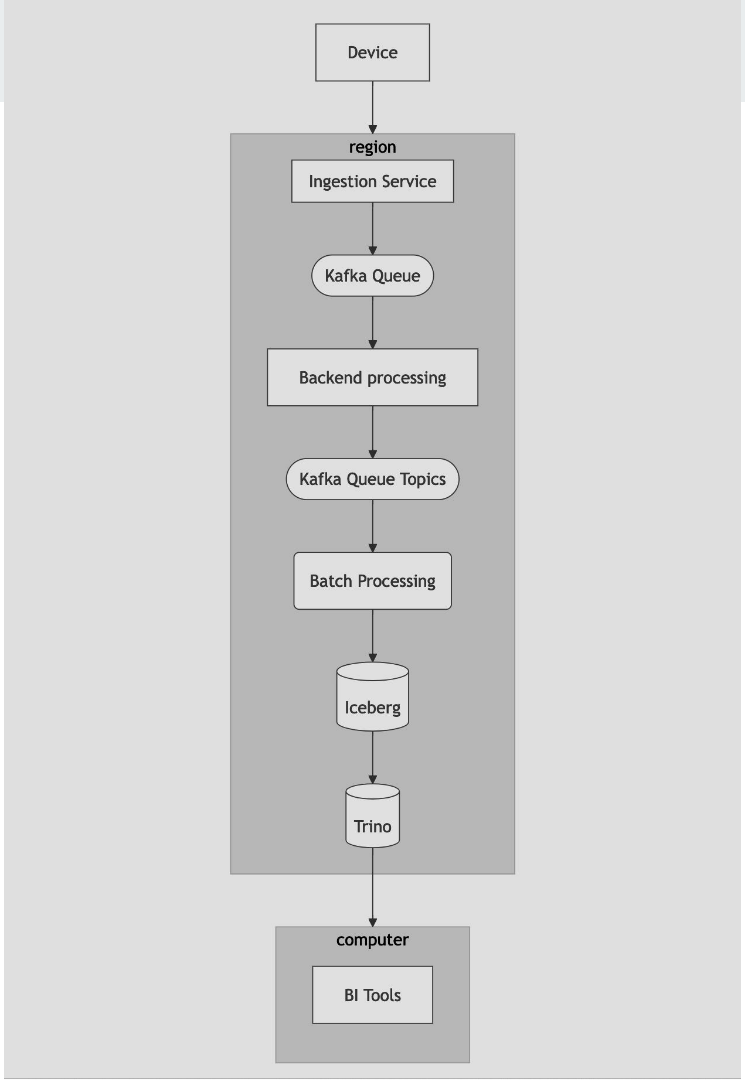
- **Cost Efficiency**: Trino and Iceberg are open-source solutions, and their integration with our existing cloud infrastructure allowed us to significantly reduce our operational overhead.
- **Flexibility in Querying**: Trino provides a **SQL query engine** that works across multiple data sources, making it ideal for our analytical queries.

- **Iceberg's Efficient Data Layout**: Iceberg is designed to handle large datasets with minimal overhead, providing better performance for incremental processing and optimizing large-scale data processing.

- **Reduced Compute Requirements**: Unlike our previous tool, Trino does not require constant scaling of clusters to handle the growing data volumes, thus further reducing cloud compute costs.

By adopting this combination, we had **no need to maintain expensive clusters** just for real-time querying, which became a major driver in cost reduction.

Architecture



The Implementation: Technical Adjustments and Trade-offs

Sacrificing Real-Time Performance for Cost Savings

One of the key trade-offs we made was accepting a slight lag in data freshness. Our near-real-time query capabilities were crucial for some use cases, but for our **internal data warehouse**, it wasn’t a critical factor. Trino, combined with Iceberg, could still deliver very fast query results—although not real-time and with some query latency—at a **fraction of the cost**.

We introduced a **slight delay in data ingestion** (~30 to 60 minutes), which allowed for more **efficient batch processing** in our pipeline. This slight increase in lag proved to be worth it, as it allowed us to cut down the need for real-time infrastructure, **saving us nearly a 7-figure per year**.

How Trino + Iceberg Worked Better Than Expected

While we initially expected slower query response times compared to our previous setup, **Trino surprised us** with its speed. Queries on historical data were returning results much faster than anticipated, thanks to Iceberg's intelligent partitioning and Trino's distributed query execution capabilities.

- **Improved Query Performance**: Even with billions of rows, Trino's ability to push down complex queries to Iceberg's optimized data format resulted in **consistently fast response times**.
- **Reduced Maintenance Effort**: The Trino + Iceberg setup required much less tuning and management compared to the other tool's ingestion and indexing processes, further cutting down operational complexity.

The Surprise: Unexpected Performance Gains

Although the initial goal was purely cost-driven, we were pleasantly surprised by the performance of the new system. Trino, working with Iceberg, was **much faster than we had anticipated** for our use cases, especially for analytical queries over historical data. This led to additional benefits that were initially unexpected:

- **Better Query Optimization**: Trino's cost-based query optimizer worked well with Iceberg's efficient table structure, leading to significant performance improvements.
- **Support for Large-Scale Datasets**: Iceberg allowed us to scale our datasets to **petabyte-scale** without worrying about performance degradation, something that became increasingly difficult with the other tool.
- **Unified Access to Multiple Data Sources**: Trino's flexibility allowed us to query not only our Iceberg tables but also other data sources, like relational databases and object stores, all from a single platform.

The Results: over 7-figure Annual Savings & Future Growth

By transitioning to **Trino + Iceberg**, we achieved our primary goal of **saving few hundred thousand per year**. Beyond just the cost savings, the move opened up opportunities for:

- **Scalable Infrastructure**: Trino's ability to handle large-scale data processing means we're well-prepared for future growth without a proportional increase in costs.
- **Simplified Data Management**: Iceberg's support for ACID transactions and schema evolution simplified how we manage data at scale.
- **Adopting Modern Data Lakes**: The flexibility of Iceberg also allows us to experiment with modern data lake architectures, giving us the ability to integrate newer technologies as we expand.

Lessons Learned and Future Directions

- **Cost vs. Performance Trade-off**: We learned that slightly sacrificing real-time performance for batch processing led to significant cost savings, especially for internal use cases.
- **Open-Source Solutions are Competitive**: Our experience showed that open-source tools like Trino and Iceberg can outperform traditional commercial solutions at a fraction of the cost.
- **Optimizing for Growth**: Our new infrastructure is designed to scale, making us confident that we can handle increasing data volumes without exponentially rising costs.

In the future, we plan to explore **further optimizations in our data pipelines** by leveraging additional features of Iceberg, such as **time travel** and **metadata management**. We're also considering the integration of machine learning models within our Trino query engine to derive insights directly from the warehouse.

Conclusion: A Successful Transformation

Transitioning from our previous tool to Trino + Iceberg was not just a cost-saving decision, but a move that redefined how we approach our data architecture. We were able to **balance cost-efficiency with performance**, allowing us to deliver actionable insights at a significantly reduced operational cost.

This success story is proof that with the right tools and a willingness to make strategic trade-offs, businesses can optimize their data infrastructure without sacrificing performance—and in our case, even exceed expectations.

Thank you

Q&A