

Case Study

Top 10 Retail

 RETAIL

TOP 10
RETAIL

MADE ON SINGLESTORE

One of the World's Top 10 Retailers is Saving Millions, and Solving Billions in Data Issues, with SingleStore

About

One of the world's top 10 retailers operates thousands of stores offering tens of thousands of items across multiple categories.

Industry

Retail

Use case

ULTRA-FAST QUERIES

TCO

MODERNIAZE

Solutions

SingleStore Self-Managed

Challenges and Goals

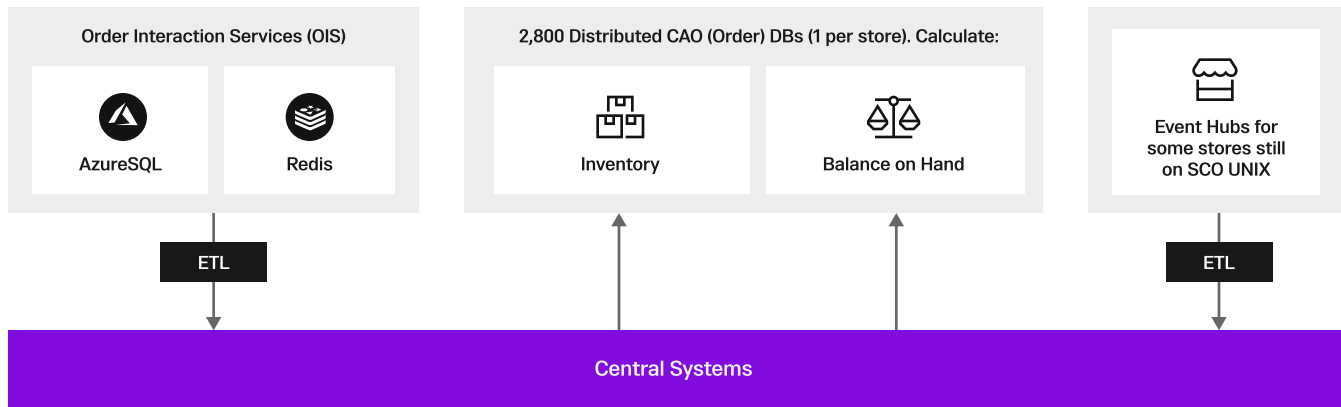
This massive organization had clearly found success, but that success was being eroded due to challenges with legacy systems, data, hardware, and design that were costing it dearly and slowing crucial processes to a crawl.

Technical Issues

This retailer's inventory forecasting process was not only incredibly complex and time-consuming, but was also fragile and failure-prone, which made the situation even worse. It was also costly and complex to scale.

For example, the inventory forecasting required a 1.5 billion-row operation that took up to eight hours to perform. This inventory operation routinely failed, which would take down a store's entire inventory system, and it took up to five hours to alert and dispatch the five technicians it took to fix the problem.

As shown in the image that follows, the inventory forecasting process was incredibly complex and required extensive data movement:



- The retailer's Order Interaction Services (OIS) system was running on Microsoft AzureSQL, with Redis as a caching layer to try to obtain faster performance. It performed calculations that were ETL'd over to an on-premise database, part of its Central Systems infrastructure, along with data from another system that was also ETL'd.
- The resulting data was pushed to thousands of Ordering databases (one per store) to calculate in-store inventory and balance on hand (BoH).
- Once this calculation was complete at each store, it had to be pushed back to Central Systems via an additional ETL process, where the data resided in OIS as the single source of truth, and at long last the retailer could finally have accurate inventory and BoH counts across its entire operation.
- The stores were still using SCO Unix (GA: 1983) for some of their systems, and the retailer had to employ event hubs to combine all the geographically dispersed data.

All of this data movement across multiple systems and bottlenecks made the inventory forecasting process extremely complex and failure-prone, preventing the retailer from scaling its data solution — and its business.

Business Impacts

This retailer quantified the business impacts of its legacy data architecture:

- It was losing 5.4 million labor days per year related to these failures, which translated to \$950.4 million in wasted labor cost per year.
- Its data downtime was making it impossible to effectively manage BoH, and every outage knocked out the mobile app for employees, which severely limited their ability to serve customers in-store.
- Another key component of any retail operation, on-shelf availability, had become a \$1+million headache for the company. If inventory is somewhere in the system but not on the shelves, or is simply out of stock and unavailable, the bottom line is that items are not on shelves available for purchase, and that costs the retailer money literally every minute.

The Way Forward

There was also some potential good news in the company's calculations: it determined that every 2% increase in BoH accuracy would result in \$10 million in additional sales annually. It crafted a blueprint for a Unified Data Store, highly performant for both data ingest and access. This would streamline operations and make it faster and more accurate for all of its in-store employees to perform tasks related to inventory and SKU management, product expiration, new product ordering, and more.

This would directly support its corporate goals around deepening what the company perceives as its four competitive moats — fresh product, its store brands, data and personalization, and a seamless ecosystem — as well as reducing operating costs to improve margins.

It would also support business unit initiatives such as:

- Sourcing products at the peak of flavor and freshness
- Reducing transit times from distribution centers
- Simplifying in-store tasks and improving scheduling
- Ensuring that stores are always stocked with products shoppers want

Plus, store associates who are empowered with the right data when they need it are far more likely to be friendly and happy when dealing with customers — the exact opposite of, “I’m sorry, our mobile app is down so I am having trouble checking on X for you.”

Technology Requirements

This massive retailer’s technology must-haves included:

- A master data store for multiple inventory and ordering apps
- A highly performant data engine with a working data set encompassing tens to hundreds of terabytes
- Support for both streaming and batch data ingest 1.5 billion rows in <30 minutes
- Low-latency performance to resolve queries in <1 second
- Five 9’s of uptime
- Reducing forecast job run time from 4-8 hours to <30 minutes
- Sharded tables for queries, eliminating multiple points of failure
- Flexibility for various types of analytics including OLTP, Simple Analytic, Complex Analytic, JSON, and Aggregates
- Integrations with Power BI, Tableau, and application frameworks from .NET and Python
- Multiple Availability Zones as well as Regional Zones in Azure (self-managed)

Why SingleStore

To resolve these costly issues, the retailer kicked off a rigorous search process and PoC for identifying a modern solution for its business. SingleStore’s main competitors for this business included the incumbents, Azure Cloud Services, Azure SQL, and CosmosDB; and potential new rival Snowflake.

As shown in the image that follows, the retailer threw down the gauntlet by challenging SingleStore, after an initial TPC-H demo, to ingest 1.2 billion rows of data in less than 30 minutes, a process that was taking it 4-8 hours every night.

SingleStore’s response: ingested 2.3X the data, 3.6 billion rows, via SingleStore Pipelines in less than seven minutes.

Agreed Upon Success Metric	Previous Performance	Desired Performance	SingleStore Percentage Improvement Over Desired Performance	SingleStore Performance Gain Accounting for Both Time and Row Count
1.5B-row inventory job	4-8 hours	30 mins or less	76%	Up to 156X (98%) faster
820M-row base trailer update	1-2 hours	30 mins or less	66%	Up to 11X (91%) faster

The PoC was also designed to prove out features and capabilities for query performance as well as more complex update table processes. SingleStore's fast INSERT-DELETE-UPSERT capabilities delivered, notching a significant improvement versus the customer's existing insert-delete-rename table process.

At that point this customer gave the Microsoft Azure team multiple attempts to come up with a better solution. In the end, the team conceded that it could not match SingleStore's performance, and offered to have the customer purchase SingleStore via the Azure Marketplace as its only viable option.

In addition to its market-leading ingest speed, the giant retailer chose SingleStore for:

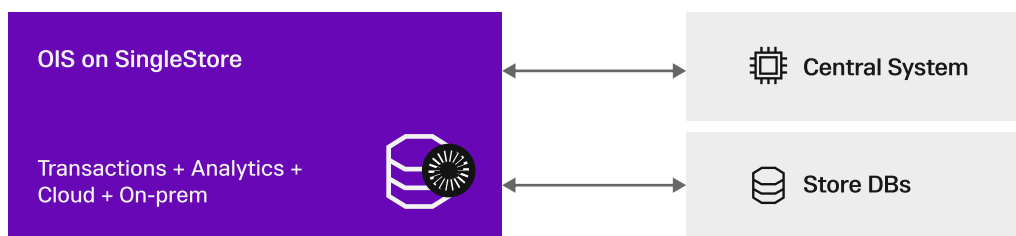
- Ability to support streaming ingest via SingleStore Pipelines, primarily from Azure Blob Storage, along with data enrichment and visualization capabilities
- Lightning-fast query performance
- Integration with Power BI, Tableau, and application frameworks from .NET, Python, and more
- Ability to directly ingest from Kafka using SingleStore Pipelines to reduce points of failure and interdependencies related to ETL processes
- Horizontal scalability
- Lower TCO

Solution

This globally recognized retailer worked with SingleStore to build its Master Data Store on SingleStore Self-Managed to provide a data service for multiple inventory and ordering applications across the organization. As shown in the image that follows, SingleStore is replacing Azure SQL, acting as the new OIS database, and its support of Kafka pipelines allows for direct import and export of data with multiple systems without the need for costly and slow ETL, including between the Ordering database and Event hubs.

This dramatically reduces the time required to perform the 1.5 billion-row inventory job while reducing points of failure within the system. Ingest occurs via raw ingestion as well as aggregated tables via stored procedures in pipelines. The majority of tables in the new system are columnar, with some reference and rowstore tables as needed.

More importantly, the queries are so fast that SingleStore also eliminated the customer's need for the Redis Cache that it used previously to try to speed up the OIS database on Azure SQL.



This innovative retailer redesigned its schema and collapsed existing table structures to optimize for SingleStore Sharded Tables, and as a result is obtaining sub-second query response time with hundreds of concurrent users.

The system includes six 16-node clusters, an 8-node cluster, and a 6-node cluster for SingleStore functionality, with three 16-node clusters and two 8-node clusters for Microsoft Power BI and Business Objects.

Outcomes

By innovating with SingleStore, this retail giant is gaining massive cost savings and improved labor productivity while serving its customers better:

156X

Faster ingest

By adopting SingleStore's solution, this customer will realize several hundred millions of dollars in cost savings related to lost labor associated with inventory job failures. The company is gaining back the nearly one billion dollars it was losing annually in wasted labor cost, and by optimizing its overall data processes, it is resolving what was another billion-dollar issue, poor on-shelf availability.

10%

Increase in BoH accuracy

With faster and more reliable inventory forecasting, this customer estimated it will be able to increase BoH accuracy by 10%, which will generate \$50 million in additional revenue by ensuring fresh stock of the products shoppers want.

68X

faster recovery point objective

This customer slashed its recovery point objective from eight hours to seven minutes: nearly 68 times faster. It eliminated ETL tooling by consuming events directly from Kafka, advancing the state of the art across its data landscape while reducing cost and complexity by getting rid of another middleware. This more efficient architecture allows for greater scalability and paves the way for expansion to other lines of business across the organization.