SingleStore™

Case Study

Fortune 25 FSC

FINTECH

MADE ON SINGLESTORE

Fortune 25 Financial Services Giant Gains Vector-driven, Real-time Investment Insights Across 2+ Petabytes of Data with SingleStore and AWS

About

A leading global financial institution providing investment banking, securities, asset management, and consumer banking services.

Industry

Fintech

Use case

AI DASHBOARDS ULTRA-FAST QUERIES

Solutions

SingleStore Helios®

Overview

Financial services platforms are highly data-intensive, processing massive volumes of data in near real time to support an array of use cases including risk analysis and management, complex analytics, trading, regulatory compliance, client reporting, customer care and billing, and more.

A Fortune 25 financial services company with offices in all major financial centers worldwide delivers a broad range of services to its large and diversified client base. This global enterprise needed a data platform that could keep pace with the demands of business today and future-proof its data infrastructure for future growth, while reducing support requirements and costs. For those reasons and more, the bank chose SingleStore as its data foundation for use cases and systems across the enterprise. This story:

- Drills down on two of those use cases where the evolution and learnings are most meaningful
- Discusses two other use cases where SingleStore is deployed
- Explores another use case where the company is poised to adopt a smart Cloud Data Strategy with SingleStore

Challenges and Goals

Few customers today are happy to wait for service. Institutional investors, financial brokerages and customers who invest a minimum of $10 million to open an account — as this company's private wealth management clients are required to do — typically demand the highest quality of service. Any issues or prolonged delays can cause them to migrate their considerable wealth to other financial institutions. Portfolio dashboards at many financial firms do not support this level of QoS. They are often powered by legacy databases that cannot reliably deliver millisecond query responses, especially during volatile market events.

Private Wealth Management Portal for high-net-worth Individuals

This financial services company had a legacy data platform supporting its Private Wealth Management Portal used by high-net-worth individuals. The system was usable when market volatility was limited and the number of clients accessing the website was low. However:

- When its user base grew larger than what is perhaps a shockingly low threshold of more than a few dozen users; and
- During periods of dramatic market fluctuations, when investors wanted to view their portfolios and quickly execute trades to reduce their risk and exposure;
- Investors experienced long delays in accessing the system. This led to a rising tide of complaints and indeed, outright panic from clients in the wake of market losses. This led to churn; and when churn occurs among a pool of clients who have invested a minimum of $10 million to open an account, losing even one of them has a major impact on the business.

Equities Trading Platform for institutional investors and brokers

The company also had a legacy, batch-oriented data processing framework supporting its Equities Trading Platform used by large institutional investors and brokerage companies. The platform supports functions including not only trading but also regulatory compliance and billing.

The platform required data extraction, transforming and loading (ETL) into data stores, and users would access those data stores to obtain the data they needed. This increased latency and rendered real-time monitoring and decision-making impossible under existing data loads, and the database could not handle increased transaction volumes.

To prevent attrition of assets under management (AUM) the company needed to deliver great interactive experiences to both individual and institutional/brokerage users under all conditions. In addition to revenue retention, however, it was also crucial to address revenue generation. Trying to deliver good user experiences with its existing data infrastructure was so resource-intensive that it hijacked skilled resources needed to build new services. So the second part of the company's dual data focus was to more effectively leverage data in order to build new revenue-generating services.

Technology Requirements

For its Private Wealth Management Portal, this financial services giant sought a resilient, fault-tolerant translytical platform that could process large volumes of data to support billions of market data events each trading day in near real time, while being resilient and fault-tolerant to reduce skyrocketing support costs. Distributed joins performance was a must, as was a SQL interface to avoid proprietary technology lock-in.

For its Equities Trading Platform, the company wanted a data platform that offered streaming data processing. This would enable real-time analytics, monitoring, detection, and event processing, while also enabling internal teams and end customers to benefit from the freshest data and rapid time to insights. Stream processing characteristics include:

- Data processed as a stream of events, rather than batches or micro-batches
- High-throughput processing for handling high data volumes
- Low-latency processing for high data velocity
- Hierarchy flattening (stateful processing) for efficient joins and fast analytical queries

Why SingleStore?

This Fortune 25 financial services institution uses SingleStore on AWS as the foundation for use cases that represent the core of its business.

Private Wealth Management Portal for high-net-worth Individuals

For this use case the team conducted head-to-head performance testing between SingleStore and its other finalist, SAP HANA. It chose SingleStore based on:

- Performance: The team wanted a solution that could deliver 15-200-millisecond response times on queries with a high concurrency of users. SingleStore met this benchmark and was on average 3X faster than SAP HANA.
- Low standard deviation: Performance was key, but so was low standard deviation to ensure consistent performance and customer experience under all conditions. SingleStore's standard deviation benchmarks were consistently better than SAP HANA.
- Manageability and cost: SAP HANA is an appliance that is costly to manage and maintain, and lacks flexible options such as horizontal scale-out and ready connectivity to client ecosystems. By contrast, SingleStore is a distributed SQL database that offers high-throughput transactions (inserts and upserts), low-latency analytics and useful context from real-time vector data.

SingleStore meets you wherever you are in your cloud journey, giving you the flexibility to deploy wherever and however you need it:

- As a fully managed cloud service with SingleStore Cloud
- Self-managed on-premises with SingleStore Self-Managed
- As a hybrid solution leveraging both

Equities Trading Platform for institutional investors and brokers

For this use case the enterprise put multiple products to the test and chose these solutions:

- SingleStore as the distributed relational DB for near-term or "hot" data and Apache HDFS (Hadoop Distributed File System) for long-term data storage
- Apache Kafka as distributed log store and message queue
- Apache Flink as stream processing framework
- REST API and web socket API for data distribution

The team chose SingleStore because it is built from the ground up to support distributed stream processing, providing an efficient, reliable, scalable data processing paradigm that leverages low-cost commodity hardware to provide market-leading big data processing:

- High availability, resilience and failover
- Horizontal scalability
- Checkpointing, recoverability and supportability
- Performance and load balancing
- Data quality and access controls

Why SingleStore

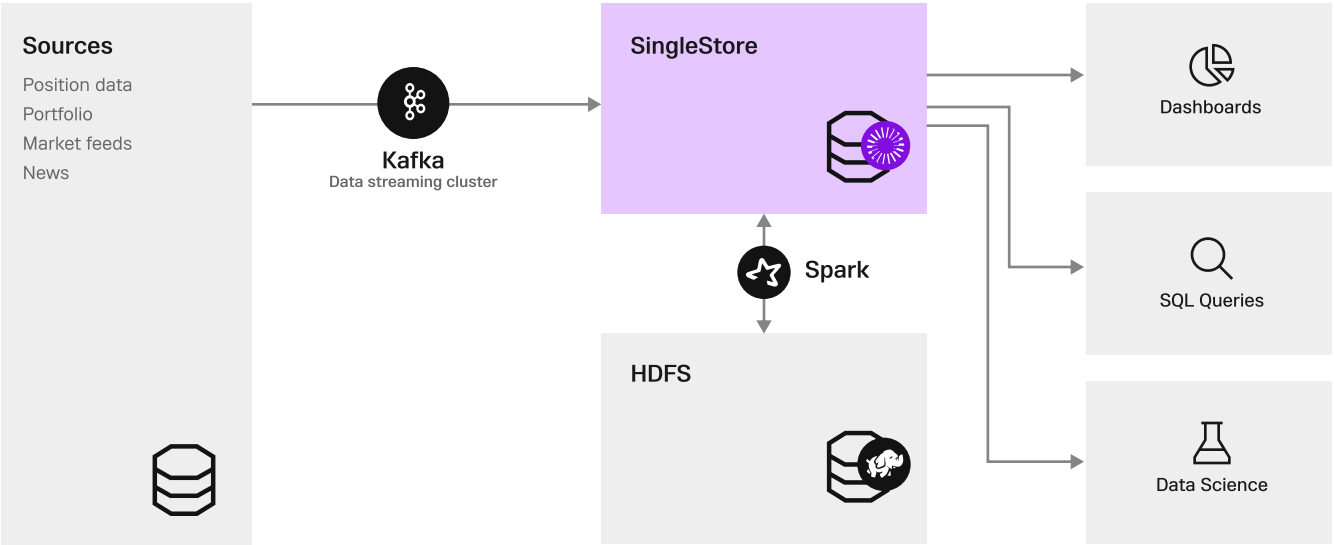
As outlined in the sections that follow, this financial services giant now supports its Private Wealth Management, Equities Trading Platform and other mission-critical use cases with SingleStore, and is looking to SingleStore as it adopts a smart Cloud Data Strategy to accelerate another use case.

Real-time Private Wealth Management Portal

SingleStore provides ultra-low-latency query responses for clients (and industry regulators) who use this customer's Private Wealth Management Portal. The Portal calculates account performance across 20+ years of data and compares active versus passive investments across funds, asset classes, and investment types. The system ensures completeness and accuracy of data from the inception date of a portfolio.

The team chose SingleStore as its distributed relational database for its near-term data supporting this Portal, leaving Apache HDFS as a data lake for long-term or "cold" data storage. SingleStore provides 20 millisecond response time across 1.4 petabytes of data (uncompressed) supporting this use case. By pushing calculations into SingleStore, the team leverages its parallelism and SIMD acceleration to deliver the rapid insights clients need.

This system runs on three SingleStore production clusters. Each cluster has two availability groups and four aggregators, organized as one parent and three child nodes, delivering high availability.

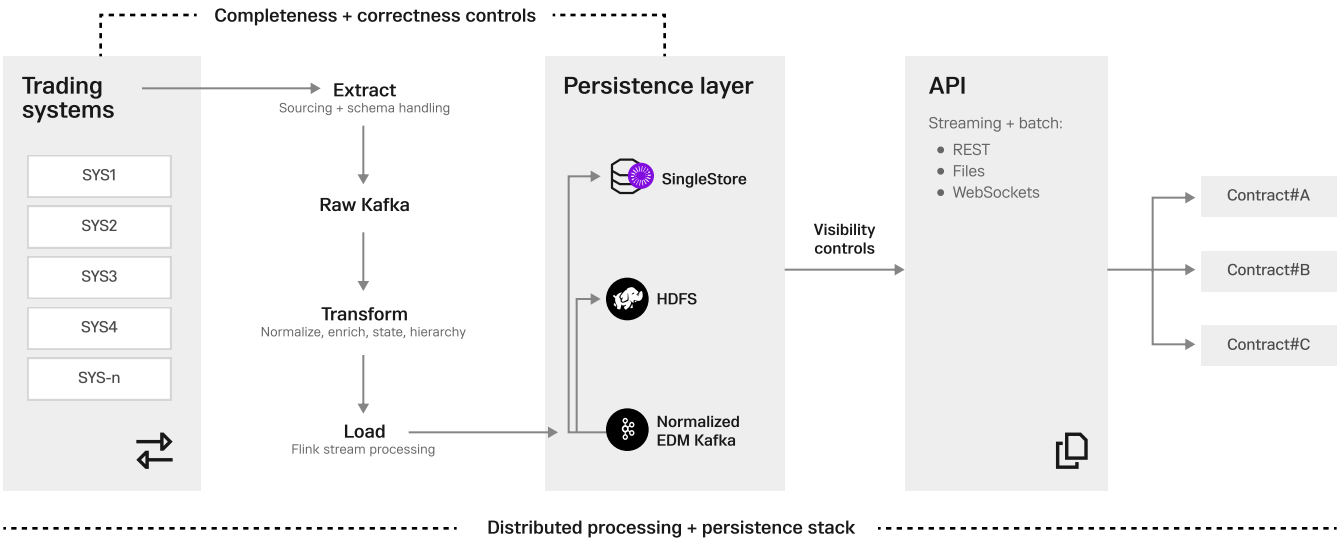


Low-latency Equities Trading Platform

With the infusion of SingleStore, this enterprise now operates a low-latency Equities Trading Platform with large amounts of data flowing into the system, and SingleStore's horizontal scalability is driving its success. The customer is currently managing 600 terabytes of data (uncompressed) in SingleStore supporting this use case.

SingleStore's distributed stream processing architecture provides high performance, scalability, resilience, state management, data enrichment, and exception handling in the streaming pipeline. SingleStore along with Apache Kafka and HDFS provide persistence and storage to handle near real-time, intra-day, and historical queries and associated data retention and distribution requirements.

The Equities Trading Platform interacts with multiple upstream systems, each with their own frequently-changing data models. To effectively deal with schema changes, the design enforces a schema handshake with the upstream system before the data exchange. Data stored as Apache Avro in the Kafka topics has its corresponding schema in the schema registry, which supports Avro data schema evolution.



Enterprise-wide Reference Data Management

Reference data is a special subset of master data used for classification, governance, and ready access to data across an entire organization. Reference data describes 1:1 relationships, 1-to-many relationships, or hierarchies. Examples include business units, cost centers, currencies, financial hierarchies, customer segments, and business processes; postal, country, language and transaction codes; and more. This customer already operates most of its Reference Data Management System in SingleStore, and at time of publication is moving two remaining areas of reference data from MongoDB to SingleStore as a result of key functionality introduced in SingleStore Pro Max, The Data Platform.

Asset Management: where this Fortune 25 bank invests its own money

This financial enterprise invests its own money in a wide range of investment vehicles including hedge funds, private equity offerings, and many more, and its Asset Management infrastructure delivers the data upon which these mission-critical investment decisions are made. At time of publication, the customer supports its multi-headed solution mainly with Sybase. However, it has deployed SingleStore to support parts of the use case, and the team's vision is to adopt an effective Cloud Data Strategy by migrating most or all of its asset management use case to SingleStore Cloud.

Many moving parts create complexity and cost — and soon it will all hit a wall

This use case is built on the challenge of distributing position data about the company's core investments to many stakeholders, and joining data with data from the Reference Data Management System and other sources. Two decades of evolution, assimilating multiple formerly separate businesses under one (conceptual) roof, and various attempted data strategies have resulted in this multi-headed infrastructure:

- A database it refers to as the golden source of truth for financial asset position data, distributed across 40+ adaptive server enterprise (ASE) replicas. These replicas must connect with other datasets via replication paths or ETL channels. User teams do their own calculations by using complex join logic or retrieving the data into an enrichment framework.
- A data warehouse that ingest data from many sources and makes it available to developers, who in turn join, filter and aggregate this data with many other datasets in order to produce the desired reports and application data views.
- An API-centric approach that attempts to provide a responsive set of APIs to answer common questions about financial asset positions. These services read position data from the so-called golden source of truth and are supported by in-memory caching. They perform data enrichment and aggregation, cache the results inside their in-memory stores and expose the data to users through a set of APIs on top of the data stores. This API-centric approach results in a more constrained and well-defined access pattern and is usually reserved for real-time applications.

- Most teams consuming this position data have their own siloed data infrastructures where they use data pipelines to access and consume position data and other datasets, join and aggregate them, and maintain local data stores.

This solution features a multitude of moving parts, requiring a significant amount of data duplication and a daisy-chained front-to-back infrastructure where data travels throughout various teams across the organization. As indicated, each consuming team has its own siloed data enrichment infrastructure and data store. Operating and maintaining these data infrastructures and all of this data everywhere does provide one benefit: data isolation. However, it also creates massive financial and organizational opportunity cost — including the reality that this sidelines valuable team resources dealing with technical debt instead of freeing them to build new revenue-generating services and applications. Its golden source of truth is also losing its luster, as that database and its 40+ ravenous replicas are reaching capacity limits and will likely be unsustainable in a few years if the company's current growth trajectory continues.

Fortune 25 financial institution's vision: one database to rule them all (all data needs)
The team's vision is a single data platform backed by a unified database that:

- Stores all of the data its various data consumers will ever need
- Provides sufficient resource isolation so different groups can run queries without impacting each other
- Supports transactional and analytical workloads (translytical/HTAP)
- Manages both recent and historical data without the need for archival mechanisms
- Offers flexible elasticity of both compute and storage
- Can sustain continuous reads and writes
- Delivers sub-100-millisecond performance

By resource isolation, the team means that each user group can continue to do its own enrichment work as it does today, and also "write back" the results into that same place, without impacting other user groups.

The resulting unified data platform will support all of the core datasets this company requires to run its business, including Positions, Transactions, Accounts, Benchmarks, Product Reference, Product Economics, Market Data, Performance, Investment Catalog, Risk and ESG.

Real-time Risk Management; and vector data functionality — in 2017
This use case supports multiple teams across the enterprise that need real-time data to drive risk management involving commodities, credit, currencies, interest rates, and liquidity to meet trading and reporting requirements.

A key reason the team uses SingleStore for the algorithmic calculations in this Risk Management System is its ability to enable analysis at the row level rather than on aggregated data, thus vastly improving the accuracy of those calculations.

Another reason was SingleStore's incorporation of market-leading capabilities years before most of the rest of the market had even heard of them. The desire to most effectively support this use case for this customer was a driving force in SingleStore's decision to build vector data functionality into SingleStore. While it is only over the past 18-24 months that most of the world has begun to embrace vector data functionality to support generative AI applications, SingleStore began building vector functionality into the platform in 2017 to support the risk management use case at this leading financial services organization.

SingleStore's built-in vector database processing lets this Fortune 25 enterprise store and search vector data. A typical vector search locates the set of vectors that most closely match a query vector. Vectors usually come from objects: text, images, video, audio, etc. Vector database searches find data based on its content or meaning, even without exact matches. For example, vector search can enable semantic search of text where a query about "investments" can return information about "stocks" and “US Treasury bonds" without using those words, because they are similar in meaning or pertain to the same broader category of things.

Some benefits the team gained from using SingleStore for vector database processing, as opposed to a specialized vector database system, are:

- The broad array of standard modern database capabilities available in SingleStore including SQL, fast distributed and parallel query processing, full-text search, extensibility, ACID transactions, high availability, disaster recovery, point-in-time recovery, broad connectivity support and more
- Less data movement is needed between different data subsystems (e.g., caches, text search systems and SQL databases) when all data, including vector data, is stored in SingleStore
- Operational costs may be reduced since fewer data management tools and fewer copies of the data are needed
- Less specialized skills and reduced labor may be needed to run an application environment

For more information on vector databases and generative AI applications, we invite you to explore the resources linked at the bottom of this page.

Outcomes

While there is clearly much more work to do for this Fortune 25 financial services customer, choosing SingleStore for some of its mission-critical data requirements is already yielding significant results:

Using SingleStore as the foundation for distributed stateful streaming processing in its Equities Trading Platform

Enables the team to effectively handle intra-day and long-term use cases, providing better scalability, resilience and performance at reduced cost. With the ability to process a very large volume of transactions at sub-second latency, this technology can help trading firms make informed decisions in real time, leading to increased efficiency and profitability.

Parallel, high-scale streaming data ingest and high concurrency with SingleStore

Delivering 10-20-millisecond query results on up to two decades of portfolio history for its high-net-worth customers in its Private Wealth Management Portal. These clients no longer experience page buffering when they access their current portfolios or slice and dice their wealth summarization across various asset classes. They can also compare their portfolios against various benchmarks and make any changes they want on the fly.

This customer is already using SingleStore to support portions of its Asset Management Infrastructure, and the team's vision is to replace its current high-cost, multi-headed, many-moving-parts approach with a single unified data platform to deliver real-time performance addressing today's requirements while positioning the organization for future growth. SingleStore is poised to meet the need. SingleStore is a unified, distributed SQL database that offers high-throughput transactions (inserts and upserts), low-latency analytics and context from real-time vector data.