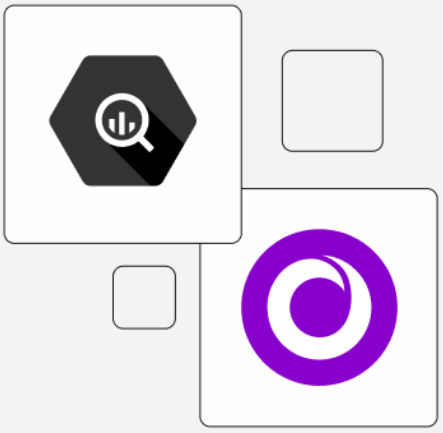
Technical Comparison Report

Google BigQuery vs. SingleStoreDB



Abhinav Srivastava, Director, Product Marketing

Quick introductions: BigQuery and SingleStore

Google BigQuery™ is a cloud-based, fully managed data warehouse service offered by Google as part of its Google Cloud Platform (GCP). It's designed to enable OLAP queries using the processing power of Google's infrastructure. SingleStoreDB is a high-performance, distributed SQL database system that supports both OLTP and OLAP workloads (aka HTAP), offering proprietary features like Universal Storage (rows and columns), full SQL support, vector capabilities and high data ingestion rates. GoogleBigQuery and SingleStoreDB share certain similarities, but differ in several ways.

Similarities

1. **Separation of storage and compute.** Both BigQuery and SingleStoreDB provide separation of compute and storage, and are highly scalable.
2. **Columnstore.** Both BigQuery and SingleStoreDB use columnstores for efficient table scans and operations like Sums, Averages, Counts or Groupings. BigQuery has a proprietary columnar format called Capacitor while SingleStoreDB's is called Universal Storage.
3. **Fast data ingestion**. Both products support bulk data loading as well as streaming data ingestion — although BigQuery places certain technical limitations (BigQuery limits Load jobs per table per day to a max of 1,500 and this limit cannot be raised. BigQuery limits Load jobs at 100,000 per project per day).

Key differences

1. **Use cases.** BigQuery is useful primarily as an OLAP data warehouse, although it can be combined with other products for other use cases. SingleStoreDB's native OLTP, OLAP, vector and multimodel capabilities make it suitable for diverse use cases including customer 360, real-time analytics, enterprise generative AI applications, user-facing analytics, recommendation systems, fraud detection, IoT and more.
2. **Query latency.** BigQuery has high query latency and as such is best suited for standard OLAP reporting and archiving activities. BigQuery's architecture discourages OLTP-style queries. Small read-write operations in BigQuery take about 1.8 seconds, while the identical action in Bigtable takes only 9 milliseconds.

3. **Concurrency**. BigQuery limits the number of queries and jobs that can be run concurrently (100), which can be a constraint for organizations with high levels of concurrent users or jobs. SingleStoreDB can cater to 100,000+ QPS on rowstore and 1,000+ QPS on columnstore.

4. **Indexing**. BigQuery doesn't support indexes (barring search index for string/ array/JSON data). While it's designed to scan large datasets quickly, the lack of indexes mean that certain query patterns may not be as fast as in other databases that support this feature. BigQuery customers typically need to think about partitioning or clustering to improve query performance. SingleStoreDB supports hash indexes for disk-based columnstore tables, and hash and ordered indexes for in-memory rowstore tables. There can be multiple secondary indexes on a table. SingleStoreDB supports **row-level indexing** (skiplist index, hash index, column group index) whereas BigQuery is limited to pruning the read set on file or file block level. SingleStore columnstores support a sort key that can provide some of the benefits of indexing for making range, equality, and list filters faster.

5. **CRUD**. BigQuery is primarily designed for analysis and not for high-frequency transactional updates. It's more suitable for appending data than for updating or deleting rows frequently. BigQuery places a daily limit on table updates and a limit on data size per request. SingleStoreDB on the other hand, is an HTAP database that provides extreme performance on Insert/ Upsert as well as OLAP operations.

6. **Streaming inserts**. BigQuery is not meant for high throughput streaming inserts. If you do not populate the insertId field when you insert rows, you are limited to 1 GB per second, per project and with a maximum of 100,000 rows per second. Additional throughput requires working with BigQuery admins to get quota increased. SingleStoreDB customers run **millions of inserts per second**.

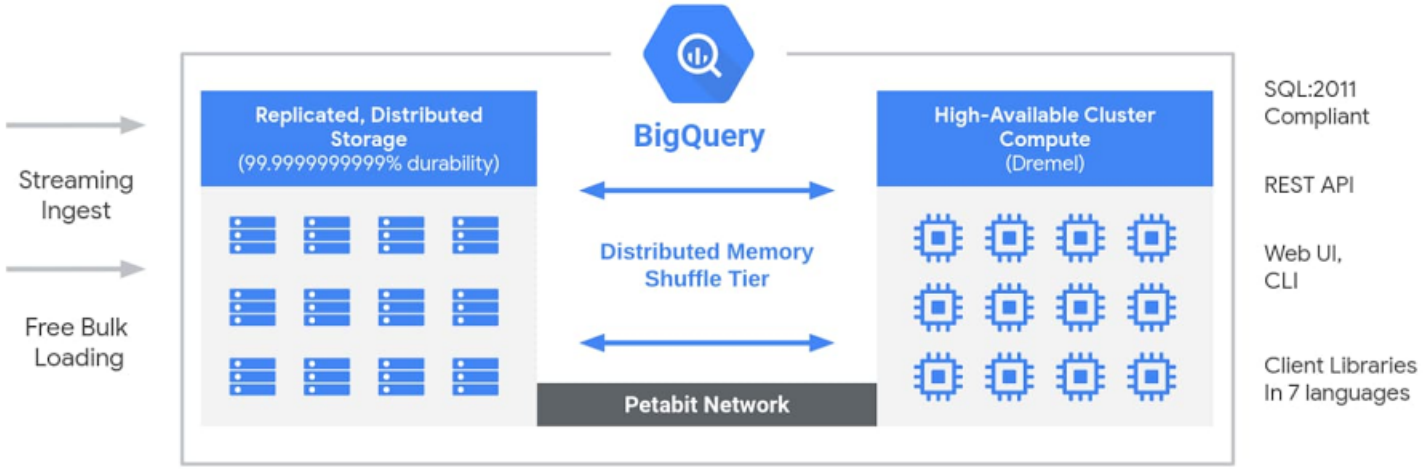
7. **DML limitations on fresh data.** While data ingested into BigQuery is available for queries reasonably soon, depending on the mode, DML statements (UPDATE, DELETE, MERGE) are not able to modify newly inserted rows for ~30 minutes, presumably until the data is compacted into Capacitor files. (See BigQuery documentation). Recently ingested data may also experience excessive query latency. In contrast, data ingested into SingleStoreDB lands in memory and is then immediately available for low latency queries.

8. **Costs**. Like other cloud data warehouses, BigQuery employs a consumption-based pricing model (on-demand or reservation). Without proper cost management, expenses can escalate rapidly if you run a lot of queries — or if your queries process large amounts of data. Streaming data into BigQuery can also have costs that can add up if you're ingesting significant amounts of real-time data (bulk loading data into BigQuery is free, but subject to resource contention with all other users without reservations). SingleStoreDB too offers a consumption-based pricing model, but costs are more predictable.

9. **Deployment.** BigQuery is available on Google Cloud or on other clouds as BigQuery Omni. SingleStoreDB is available on-premises, on Kubernetes or as a fully managed cloud service on AWS, Azure and GCP.

BigQuery architecture

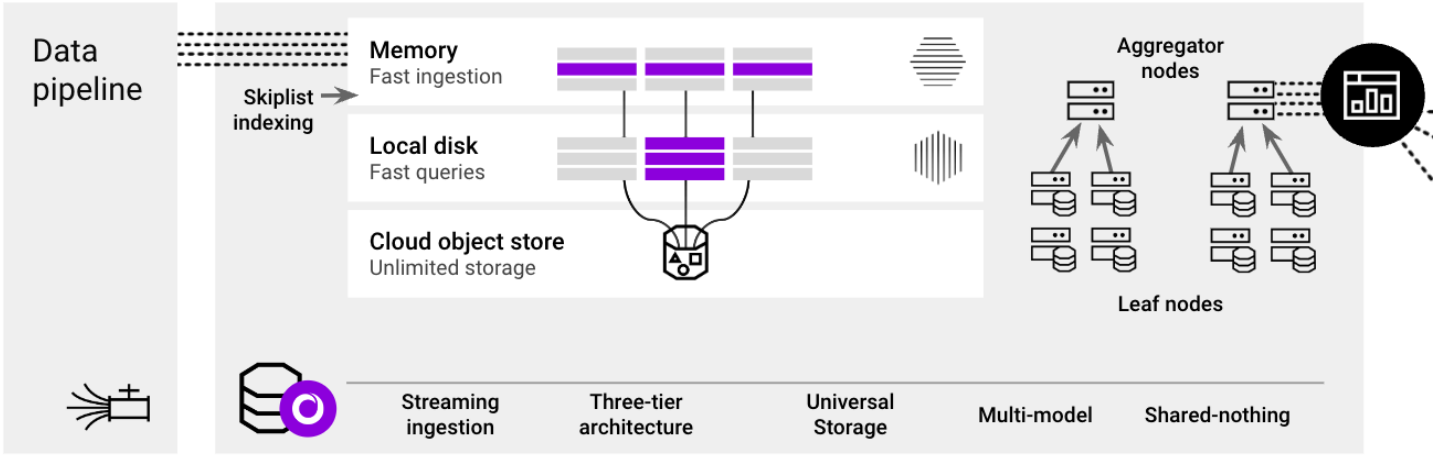
- **Decoupled storage and compute.** Allows compute to be scaled independently of storage.
- **Multi-tenant**. Under the hood, BigQuery employs a vast set of multi-tenant services driven by low-level Google infrastructure technologies like Dremel, Colossus, Jupiter and Borg.
- **Serverless**. BigQuery abstracts away the complexities of infrastructure management, allowing users to focus on analyzing data and gaining insights
- **Large-scale datasets**. Built on Dremel, BigQuery allows for massive parallelism, making it suitable for running large-scale analytics.



BigQuery high-level architecture

(Source: An overview of BigQuery's architecture by Rajesh Thallam, Solutions Architect @Google, 2020)

SingleStoreDB architecture



SingleStoreDB's three-tier, distributed SQL architecture

SingleStore architecture consists of several components that work together to provide high performance, scalability, and fault tolerance. The key components of SingleStore architecture are as follows:

- **Three tiered storage.** SingleStoreDB offers both in-memory rowstore and on-disk columnstores, enabling it to handle both OLTP and OLAP use cases while minimizing ETL and data duplication.
    - **Rowstore**: Rowstore tables are in-memory tables optimized for high-performance OLTP workloads.
    - **Columstore (Universal Storage)**: Universal Storage is the default and recommended option, supporting both online transaction processing (OLTP) and hybrid transactional and analytical processing (HTAP) workloads. This helps customers lower their total cost of ownership (TCO).
    - **Bottomless (Unlimited) storage**. In addition to rowstore and columnstore SingleStoreDB supports spilling data to cloud object stores.
- **Distributed architecture.** A SingleStoreDB cluster is made up of nodes, which hold partitions of data and are responsible for query processing. Each node holds several partitions of data. Each partition is either a primary which serves both reads and writes, or a replica which only serves reads and is used for high availability.
    - **Aggregators**: Aggregators are nodes responsible for query routing and metadata management in the cluster. They handle incoming queries, distribute them to the appropriate leaves, aggregate intermediate results, and return the final results to clients. Each cluster has one master aggregator and can have multiple child aggregators.
    - **Leaves**: Leaves are nodes that store subsets of the cluster's data. They function as storage and compute nodes, handling data processing operations. SingleStore automatically distributes data across leaves into partitions, with each leaf containing several partitions. Adding more leaves to the cluster improves storage capacity and query performance.
    - **Master Aggregator**: The master aggregator is a specialized aggregator that handles cluster monitoring, failover, basic cluster operations, and Data Definition Language (DDL) operations. There is only one master aggregator in a cluster, and it orchestrates the overall cluster operations.
    - **Worker Nodes**: Worker nodes are responsible for executing tasks and processing data within the cluster. They run web properties, back-end services, scheduled jobs, and data stores. Each worker node can host multiple leaves.
    - **Partition**: A partition is a subset of a database's data held within a leaf node. Each partition contains a vertical slice of data, distributed based on a hashing algorithm on the primary key or randomly for keyless sharded databases. SingleStore splits the database into partitions when creating a database, distributing them evenly among available leaf nodes.

Performance and scalability

Both SingleStoreDB and BigQuery are highly scalable.

- **BigQuery** can automatically scale based on query demand without the need for manual intervention or capacity planning.
- **SingleStoreDB** offers horizontal scalability, allowing nodes to be added on-the-fly. Its distributed system architecture supports high-throughput operations. SingleStoreDB decouples storage and compute with its unlimited storage architecture that leverages three data storage tiers: in-memory, disk and cloud object storage. This allows users to effortlessly scale compute resources, meeting the needs of any workload while managing the storage needs completely independently. SingleStoreDB's Workspaces feature allows isolation of workloads to power multiple workloads (read-only or read-write) on the same shared database.

Multi-model capabilities

- **BigQuery** primarily supports a relational database model, although it can support geography and JSON types as well.
- **SingleStoreDB** is a multi-model database and supports relational, geospatial, time series, JSON and text data. In addition, SingleStoreDB natively supports vector data.

Enterprise readiness

- **Both BigQuery and SingleStoreDB** support data redundancy, high availability, tiered storage, versioning, point-in-time recovery (SingleStoreDB)/ time travel (BigQuery), backups, access control/ IAM, data encryption in transit/ at rest, etc.

Extensibility

- **BigQuery** supports custom functions with its Remote Functions feature, stored procedures for Apache Spark and integrates Google Colab to allow writing and executing Python code in Jupyter notebooks.
- **SingleStoreDB** provides multiple choices to increase the extensibility of your database operations. These include user-defined functions **(UDFs)**, user-defined aggregate functions **(UDAFs)**, table-valued functions **(TVFs)** and stored procedures **(SPs)** — all of which can be created, executed, replaced or deleted with the appropriate privileges. And with Code Engine — Powered by Wasm, SingleStoreDB also supports extensibility for high-level languages including C, C++ and Rust. SingleStore **Notebooks** provide the ability to write, execute and collaborate on Python and SQL code.

Capability comparison

Capability	Why this matters	SingleStoreDB OLTP + OLAP	BigQuery Pure OLAP
Speed	**Immediate insights; responsive application experience for end users**		
In-memory performance	Responsive applications and instantaneous analytics	●	○
Streaming data ingestion	Immediate queryability of streaming data from multiple sources (files, Kafka, Spark, HDFS or object stores such as S3)	●	●
Columnstore	Low-latency (~10s of milliseconds) on complex queries	●	◒
Materialized views	Boost query performance, especially for frequently run complex queries or those that need to process large amounts of data	○ *roadmap	●
Analytics on semi-structured data	High-performance analytics on relational and semi-structured data in the same database engine	◕	◒
Scale	**Adapt to growing needs**		
Horizontal scalability	Distributed (shared-nothing) architecture that decouples storage and compute to allow scaling using low-cost infrastructure	●	●
Read replicas/ multi-region deployments	Run multiple workloads and scale compute instances across shared databases	●	●
Resiliency	Run critical applications and workloads; mitigate risks on business operations and reputation.	●	●
Deploy anywhere	Ability to deploy both as a fully managed cloud service or self-managed on-premises	●	◒
Simplicity	**Minimize complexity and costs**		
SQL-powered OLTP + OLAP with zero ETL	Minimize data movement and duplication; minimize complexity and costs emanating from sprawl; power and simplicity of SQL for CRUD and rich query operations	●	◒
Multi-model	Ability to store and query multiple secondary data formats (JSON, time series, geospatial, full-text search, etc.)	●	●
Multi-tenancy	Support multi-tenant applications each with distinct groups of users (tenants)	◒	●
Serverless	Abstracts away complexities of infrastructure management, allowing users to focus on analyzing data and gaining insights	○	●
Vector search engine	Efficiently handle large amounts of vector data for vector similarity search	◒	●
Open-source software	Community-developed software that's typically free to use and distribute	○	○

Why others choose SingleStoreDB over BigQuery

Complex queries completed in	Superior performance up to	Reduce TCO by	Query performance improved by
1-3ms	24x	75%	220%

- "*We looked at four players: SingleStore, BigQuery, Cloudera, and Snowflake. Several of the competitors have good features, but they all had a pitfall or two that held us back, whereas SingleStore really ticked all the boxes.*" Mauricio Aristizabal, Principal Data Architect, impact.com. Read more singlestore.com/customers/impact
- "*SingleStore can process complex queries with large data sets within **1-3ms**. The closest Snowflake or BigQuery can get us is in the 200ms range.*" — Leading Enterprise Customer
- **ZoomInfo**, a software company (with a valuation of several $B) that provides data for companies and business individuals wanted an alternative to BigQuery and were looking for 1) High performance 2) lower TCO 3) No cloud vendor lock in 4) High Concurrency 5) Elastic scalability. In their evaluation, SingleStoreDB was able to exceed their requirements, delivering **superior performance (24X)** and **lower TCO (4X).**
- A high growth company in the martech/salestech space faced challenges with costs, long data load times (5+ hours) and were looking to augment BigQuery for better performance. With SingleStore, they realized a $1.2M annual cost savings, **220% improvement** in query performance and were able to support **~4x data volumes** and concurrency.

Recommendations

- **Avoiding vendor lock in**. Customers seeking to avoid cloud vendor lock-in or hybrid cloud deployment may want to consider a data platform that provides more deployment flexibility, and is wire compatible with open-source databases.
- **Managing costs**. Customers should compare the overall costs of multiple data systems for OLTP, OLAP and AI. Costs include software licensing, infrastructure, ETL tooling and compute, staffing/ training, etc. Customers may also want to consider other costs like system resources that normally get correlated with cost of ownership and get inversely correlated with latency.
    - Cost of small updates, deletes or medium size updates or deletes touching few rows in many files.
    - Selective reads and object retrieval.
- **Ensuring performance**. Customers may want to execute a POC or run tests to compare performance (TPC-H at suitable scale, data ingestion, query latency over freshly created/ ingested data, frequent data updates, etc.)

SingleStoreDB: Transact, analyze + contextualize data in real time.

SingleStoreDB is a real-time distributed SQL database that empowers users to transact, analyze and contextualize data in real time. It delivers stellar performance for both transactional applications (OLTP) and analytical workloads (OLAP).

With support for **streaming data ingestion** (Pipelines), a unique table type that supports **both transactions and analytics** (Universal Storage), **separation of compute and storage** (unlimited storage), limitless **point-in-time recovery** (PITR) and a **distributed** (shared-nothing), **MySQL-compatible** architecture, SingleStoreDB provides fast-growing companies the ability to build and scale real-time applications and analytics, generative AI applications and more.

All this comes with **compelling price performance**. SingleStore customers often report **2-3x better TPC-H performance** at **a fraction of the cost** of multiple technologies for OLTP, OLAP, ETL, NoSQL, etc. See our benchmarks here singlestore.com/blog/tpc-benchmarking-results.

Capabilities

SingleStoreDB empowers the world's makers to build, deploy and scale modern, intelligent applications — leading to faster, real-time decisions, lasting customer experiences and more cost-efficient operations.

Transact
Drive real-time analytics with super-fast ingest pipelines and a horizontally scalable architecture.

Analyze
Columnstore capabilities and aggregate functions for the most demanding analytics.

Contextualize
Vector functions, semantic and keyword search, and Notebooks provide a foundation for building AI applications.

Universal Storage
Unique single table type for transactions and analytics. Compatibility with ANSI SQL, MySQL + MariaDB ecosystem; NoSQL

Pure speed
Fast transactions, analytics, vectorization and query compilation.

Fast ingestion
SingleStore Pipelines — load data with updates.

Unlimited storage
Separation of storage + compute in a unified database

SingleStore Kai™
Power up to 1,000x faster analytics on JSON for applications built on MongoDB

Multi-model
Support for various data types, including vectors.

Run anywhere
Hybrid, multi-cloud, SaaS, on-premises, Kubernetes operator

Intelligence
Build generative AI and other apps with Notebooks on top of SingleStoreDB