Md Sarfaraz Hussain Follow
Nov 12 · 4 min read · Listen

Data Lakehouse — A new paradigm to build Data Platform

Data Lakehouse is a new generation of open platforms that unify Data Warehousing and advanced analytics capabilities of Data Lake.

Why do we need a Lakehouse?

Lakehouse provides -
i. direct access to open data formats, such as Apache Parquet
ii. provide first-class support for machine learning and data science
iii. state-of-the-art performance

As we have seen the problems with traditional Data Warehouses and Data Lakes in my last blog (recommend to read the blog and then come back to this), but what we would love to have to cater to all those problems?

We simply want the best of Data Warehouse and Data Lake.

Goodness of Data Warehouses	Goodness of Data Lakes
Clean and Structured data	Massive scale-out
Transactional Reliability	Open formats
Fast SQL queries	Mixed workloads

Advantage of a Lakehouse

So, as a result of the above considerations, we have table formats like Delta Lake/Apache Hudi/Apache Iceberg which provides the Scale of Data Lake, Reliability & Performance of Data Warehouse, and provides low latency access that can be used for streaming as well.

Simply, Data LakeHouse = Data **Lake** + Data Ware**House**

- Lakehouse can help address several significant challenges with Data Warehouses, including data staleness, reliability, the total cost of ownership, data lock-in, and limited use-case support of AI/ML.
- Lakehouse combines the key benefits of Data Lakes and Data Warehouses allowing it to be low-cost storage in an open format accessible by a variety of systems/frameworks like Spark, Flink, Presto, Trino, etc.
- Lakehouse is an especially good fit for cloud environments with separate compute and storage: different computing applications can run on-demand on separate computing nodes (e.g., a GPU-based cluster for ML) while directly accessing the same storage data.
- The first key thing we need for implementing a Lakehouse is to have the system store data in low-cost object storage (like S3, GCS) along with a transaction layer like Delta Lake, Hudi, Iceberg, etc.
- These table formats or transaction layers are a combination of actual data along with metadata information i.e. Apache Parquet (which stores the data) and JSON/Avro (to keep metadata).

Standards for Lakehouse

In a Lakehouse, we incrementally clean up the data to improve the quality of the data and as we move ahead in the process from Bronze to Silver to Gold (it's just a naming standard and you can eliminate any of the layers based on the use-case i.e., there are no hard and fast rules — consider it as the value of data that changes during the process). With the Lakehouse architecture, multiple tables of data are kept in the same data lake. We can write a batch and stream data to the same table. Data is written to a series of progressively cleaner and more refined tables.

- The data from Streaming and Batch jobs are first captured in the Bronze table in their raw formats.

- This is just whatever comes out of Kafka, any kind of ingestion service, etc.

- It's the dumping ground for raw data which can hold historical data of many years.

- Stores all data including all the corrupted, missing data, etc.

- If we keep all the raw data in the Bronze layer, it now becomes very easy to solve cases where we might have made a mistake (a bug in existing code) or we come up with some new business rule that we want to compute from scratch.

Silver Layer

- This layer is the "single source of truth" that users can trust.

- We enforce schema validation in this layer and if any record does not comply with the schema we move it to a quarantine table.

- Data is cleaned and processed to make it queryable.

- We perform some of the cleaning, filtering, transformation (like adding new columns, updating column value, etc).

- We also have the flexibility to perform some parsing, joins, lookup/augment with more information, etc.

- Data is normalized so that it is easier to query.

- We materialize this intermediate data so that we can serve multiple purposes like –

a. This clean data can be used by multiple people in different downstream applications.

b. Training ML models on it.

c. Used for debugging. When something goes wrong in the final analysis/result, we always have this table that we can query with the full support of SQL.

Gold Layer

- Any downstream application (Spark, Presto, Hive, etc.) can read this data as well.

Data Lakehouse Architecture

The idea is to build a decoupled infinite-storage (Google Cloud Storage/AWS S3) and infinite-compute (Spark, Presto) architecture, which can handle the data volume scale at large. Having a data platform with an architecture which has separate compute and storage layers allows for a separation of concerns: running compute clusters for the analytic jobs and storing data at the lowest cost per terabyte.

I hope you have liked this blog! That's all from this blog! Happy Learning! 😊

Further Reading:
1. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics
2. Apache Hudi — The Data Lake Platform

Connect me on LinkedIn — Sarfaraz Hussain

Thank you! Stay tuned!! 😊