kinetica

The Speed Layer Design Pattern for Analytics

The speed layer handles high-velocity data streams that require immediate processing along with integrated historical data to drive real-time decision-making.

Organizations that require both real-time insights and in-depth analytics often encounter the conundrum of using streaming tools for real-time data processing with basic analytics or a data warehouse with advanced analytics but at high latency.

Historically, builders and users of data platforms were forced to choose between a basic analytic insight in real time or a comprehensive analytic insight well after an event occurred. In many cases, one of those options is perfectly fine.

Increasingly however, organizations want to make near-real-time decisions on a larger corpus of data. This leads to better fraud prevention, threat detection, fleet management and other use cases arising from the proliferation of sensor and machine data. Recently, a new architectural design pattern known as a "speed layer" emerged that overcomes the tradeoff between advanced analytics and real-time insights.

Streaming

Steaming tools like Kafka, NiFi and Kinesis are used for building real-time data pipelines and streaming applications. They are designed to handle high-throughput, low-latency data movement. They are used to collect and process data from various sources such as sensors, logs and social media feeds, and deliver this data to multiple applications and databases simultaneously.

While streaming tools provide some basic analytical capabilities, the focus is on moving data from one system to another and processing data in real time, rather than performing complex analytics on the data. They can perform some basic forms of analytics, such as filtering and aggregating data over a narrow window in real time through built-in, lightweight stream-processing libraries. For example, a financial institution might use a streaming tool to quickly detect anomalies and take immediate actions to prevent losses or mitigate risks.

However, they are not optimized for complex querying or analysis of historical data. A key limitation of performing analytics with streaming tools is that they lack contextual data. They do not store historical data or contextual information necessary to see the full picture. They don't provide more advanced analytical capabilities like high cardinality joins, n-way joins across large fact tables, time-series functions across a larger historical window, or other feature generation and inference techniques used in advanced machine learning.

Without access to contextual data or advanced analytics capabilities, streaming tools are limited in their ability to provide insights and drive meaningful business decisions. Therefore, organizations that require in-depth analytics or querying of large data sets may need to use other tools commonly associated with data warehouses and data lakes.

Streaming Technologies

Streaming Technologies have no historical data or ability to join to other data sets, making this option ***too limiting.***

ksqlDB

Spark Streaming

Data Warehouses and Data Lakes

Data warehouses and data lakes are designed to store and analyze large volumes of historical data across multiple subject areas, which allows for more advanced analytics and modeling. Context can be key, and these environments integrate data from different sources to build a composite view and triangulate across different systems of record. The results are more comprehensive than those from streaming tools.

However, data warehouses typically have higher latency, which means that insights and analysis may take longer to produce. Data warehouses and data lakes incur latency when providing insights due to ETL (extract, transform and load) or ELT (extract, load and transform), and query complexity. ETL/ELT involves batch loading of data, table locking during the ingest process and data engineering to, among other things, allow faster and easier querying of the data through denormalization. Data warehouse and data lake queries may involve multiple tables, aggregations and joins, which all take time.

There are, of course, a variety of techniques to optimize this, but there's no getting around the fact that there's significantly more time to insight compared to streaming once the complexity of the question increases.

Data Warehouses and Data Lakes

Data Warehouse and Data Lake market options do not enable data streaming analysis in a usable latency profile, making this option ***too slow***.

snowflake

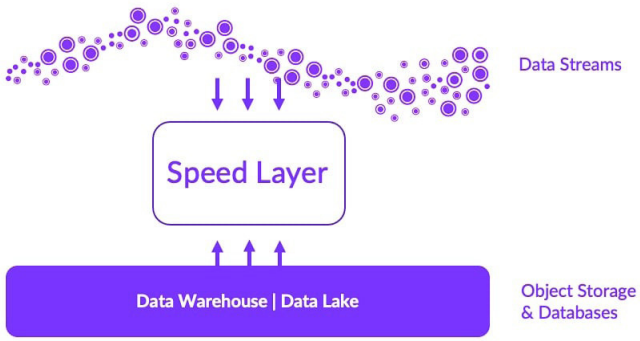
amazon REDSHIFT

databricks

Speed Layer

In a modern data architecture, speed layers combine batch and real-time processing methods to handle large and fast-moving data sets. The speed layer fills the gap between traditional data warehouses or lakes and streaming tools. It is designed to handle high-velocity data streams that are generated continuously and require immediate processing within the context of integrated historical data to extract insights and drive real-time decision-making.

A "speed layer" is an architectural pattern that combines real-time processing with the contextual and historical data of a data warehouse or lake. A speed layer architecture acts as a bridge between data in motion and data at rest, providing a unified view of both real-time and historical data. This approach provides a hybrid solution that offers the benefits of both real-time processing and contextual analytics.



To deliver on its promise, a speed layer typically includes two components:

Extreme ingestion:

The speed layer must be able to ingest data from multiple sources in real time from both the streaming source and the data warehouse and lake. This involves native connectors to streaming tools, distributed headless ingest that removes central bottlenecks and a lockless architecture that ensures data is available for query as fast it can be streamed.

Data is often moved from a slow object storage or disk (where data warehouse and lake data are persisted) into memory for fast querying. Key to note here is that when data is persisted in the speed layer, it is transient. The long-term record remains the data warehouse and lake.

Extreme query speeds:

The speed layer must provide a way to query and analyze real-time data in real time, typically using new breakthroughs in query acceleration such as vectorization. In a vectorized query engine, data is stored in fixed size blocks called vectors, and query operations are performed on these vectors in parallel, rather than on individual data elements.

This allows the query engine to process multiple data elements simultaneously, resulting in orders of magnitude faster query execution. This is in contrast to conventional distributed analytic databases that process data on a row-by-row basis, which is much slower and requires more computational resources. In addition to improving query performance, this approach can also reduce the amount of compute and data engineering required, making them more efficient and cost-effective.

>> *You can experience the benefits of a speed layer for free on Kinetica Cloud, with easy-to-use workbooks featuring prebuilt Kafka topics.*