Sponsored by: ahana

EMA TECHNICAL
CASE STUDY

Uber

The Criticality of Analytics at Uber

Everyone knows Uber, but not everyone knows the clockwork that makes Uber tick. Everyone knows Uber as a shared service for point-to-point transportation, but not everyone knows Uber as a data and analytics company.

Every day, more than 18,000,000 riders key their location and destination into the Uber mobile app to request a ride or a meal. Unbeknownst to them, each transaction is plugged into complex algorithms built on insight gleaned from petabytes of data. The results come back in seconds with the most optimal recommendations for nearby drivers and pricing based on supply and demand economics.

Uber's DNA as an Analytics Company

The Uber digital business model is simple. A customer needs to get from their current location at point A to a specific location, point B. The customer has a mobile device to request a ride from the Uber network of drivers. Connecting the customer with the nearest driver happens instantaneously and is not that difficult.

Every transaction presents an amazing opportunity for Uber. An increase or decrease of just ten cents for every transaction amounts to $1.8 million dollars every day, or $657 million per year. The combination of location analytics with pricing optimization makes Uber a transportation company, a logistics company, and an analytics company. It is data and analytics that drive Uber to growth, profitability, and expansion into new business areas.

Uber's Pursuit of Hyperscale Analytics

The scale of Uber's analytical endeavor requires careful selection of data platforms with high regard for limitless analytical processing. Consider the magnitude of Uber's footprint. The company operates in more than 10,000 cities with more than 18 million trips per day. To maintain analytical superiority, Uber keeps 256 petabytes of data in store and processes 35 petabytes of data every day. They support 12,000 monthly active users of analytics running more than 400,000 queries every single day.

To power this mammoth analytical undertaking, Uber chose the open-source Presto distributed query engine. Teams at Facebook developed Presto to handle high numbers of concurrent queries on petabytes of data and was designed to scale up to exabytes of data. Presto was able to achieve this level of scalability by completely separating analytical compute from data storage. This allowed them to focus on SQL-based query optimization to the nth degree.

HIGHLIGHTS

Vendor name:
Uber

Product name:
Presto

Product function:
Distributed Query Engine

Vendor contact:
Girish Baliga

Pricing information:
Open-source

Availability:
Generally available

Uber

What is Presto?

According to the Presto Foundation, "Presto is an open-source distributed SQL query engine for running interactive analytic queries against data sources of all sizes, ranging from gigabytes to petabytes."[1]

To achieve maximum scalability and to support a broad range of analytical use cases, Presto separates analytical processing from data storage. When a query is constructed it passes through a cost-based optimizer, then data is accessed through connectors, cached for performance, and analyzed across a series of servers in a cluster. Because of its distributed nature, Presto scales for petabytes and exabytes of data. Presto's cost-based query optimizer contributes significantly to both performance and scalability with advanced technology like join enumeration, where the join order is optimized, or join distribution selection, where workloads are either broadcast to multiple servers or partitioned for optimal performance.

To speed time to production, Presto also includes a library of smart connectors designed for basic data processing and pushdown of workloads that perform better on source platforms. In addition, Presto offers a large library of analytical functions and the ability to create user-defined functions specific to each organization.

The Evolution of Presto at Uber

In the Beginning

Uber began their analytical journey with a traditional analytical database platform at the core of their analytics. However, as their business grew, so did the amount of data they needed to process and the number of insight-driven decisions they needed to make. The cost and constraints of traditional analytics soon reached their limit, forcing Uber to look elsewhere for a solution.

Uber understood that digital superiority required the capture of all their transactional data, not just a sampling. They stood up a file-based data lake alongside their analytical database. While this side-by-side strategy enabled data capture, they quickly discovered that the data lake worked well for long-running queries, but it was not fast enough to support the near-real time engagement necessary to maintain a competitive advantage.

To address their performance needs, Uber chose the Presto query engine because of its ability, as a distributed platform, to scale in linear fashion and because of its commitment to ANSI-SQL, the lingua franca of analytical processing. They set up a couple of clusters and began processing queries at a much faster speed than anything they had experienced with Apache Hive, a distributed data warehouse system, on their data lake.

While Uber enjoyed the speed of the new query engine, they were also initially stymied by the effort required to manage Presto clusters. The Uber team spent most of their time making sure the clusters were up and running to support the business users.

[1] https://prestodb.io/

Uber

Automation enabled Uber to grow to their current state with more than 256 petabytes of data, 3,000 nodes, and 12 clusters.

Continued High Growth

As the use of Presto continued to grow, Uber joined the open-source community at the Presto Foundation as a founding member, along with Facebook. Their initial contributions were based on their need for growth and scalability. Uber became a major contributor to the automation of cluster management, workload management, the ability to process more complex queries, and security.

Automation

Due to increased speed enjoyed by users and the familiarity of an SQL-based query engine, the use of the Presto data lake grew quickly. To support growing usage, the Uber team went to work on automating cluster management to make it simple to keep up and running. Automation enabled Uber to grow to their current state with more than 256 petabytes of data, 3,000 nodes, and 12 clusters.

Uber put process automation in place to quickly set up and take down clusters. In addition, they built a proxy server and placed it between Presto and the data. This enabled them to automate query routing and isolate workloads based on priority. To meet an ever-broadening set of use cases, Uber configured specialized clusters for different kinds of workloads across their analytical mix.

Workload Management

Because different kinds of queries have different requirements, Uber makes sure that traffic is well-isolated. For example, ETL workloads are very different from Notebook sessions or third-party business intelligence applications. Uber architects their system with these different workloads in mind and they route different workloads to the appropriate clusters. This enables them to batch queries based on speed or accuracy. They have even created subcategories for a more granular approach to workload management.

Because much of the work done on their data lake is exploratory in nature, many users want to execute untested queries on petabytes of data. Large, untested workloads run the risk of hogging all the resources. In some cases, the queries run out of memory and do not complete.

To address this challenge, Uber created and maintains sample versions of datasets. If they know a certain user is doing exploratory work, they simply route them to the sampled datasets. This way, the queries run much faster. There may be inaccuracy because of sampling, but it allows users to discover new viewpoints within the data. If the exploratory work needs to move on to testing and production, they can plan appropriately.

Complex Queries

Uber faced another challenge with complex queries. High compute queries on massive data can cause Presto to run out of memory. To address this issue, Uber monitors Presto usage to provide assistance. SQL users are accustomed to working with small data sets on optimized data storage platforms where they can write any kind of SQL. Uber works to reeducate users who have never used a big data platform with massive amounts of data.

Uber

Along with directing users to sample data sets, Uber works with them to reorder their statements and optimize how they write queries for massive data. In addition, complex SQL queries benefit from the fact that the Presto architecture is distributed, supports massively parallel processing, does cost-based optimization, and processes analytics in memory. Presto can handle most queries, but some queries need to be moved to other platforms more suitable for complex queries.

Security

Early in Uber's use of Presto, they also recognized its limited built-in security capabilities. In fact, their initial evaluation was that they could not use any of the security features out of the box.

Uber adapted Presto to take users' credentials and pass them down to the storage layer, specifying the precise data to which each user has access permissions. As Uber has done with many of its additions to Presto, they contributed their security upgrades back to the open-source Presto project.

The Technical Value of Presto at Uber

Along with hyperscalability, Uber chose Presto for its support across a broad range of use cases and its leverage of modern, low-cost data storage. Understanding the technical value of Presto demonstrates two benefits of open-source software development. One, when Uber switched to Presto, they saw an immediate increase of query processing speed and quickly began to grow their internal userbase. Two, when Uber saw features lacking in Presto, they developed new technology to meet those needs and contributed much of their work back to the open-source Presto project.

Analyzing Complex Data Types with Presto

As a digital native company, Uber continues to expand its use cases for Presto. For traditional analytics, they are bringing data discipline to their use of Presto. They ingest data in snapshots from operational systems. It lands as raw data in HDFS. Next, they build model data sets out of the snapshots, cleanse and deduplicate the data, and prepare it for analysis as Parquet files.

For more complex data types, Uber uses Presto's complex SQL features and functions, especially when dealing with nested or repeated data, time-series data, or data types like maps, arrays, structs and JSON. Presto also applies dynamic filtering that can significantly improve the performance of queries with selective joins by avoiding reading data that would be filtered by join conditions.

For example, a parquet file can store data as BLOBS within a column. Uber users can run a Presto query that extracts a JSON file and filters out the data specified by the query. The caveat is that doing this defeats the purpose of the columnar state of a JSON file. It is a quick way to do the analysis, but it does sacrifice some performance.

Extending the Analytical Capabilities of Presto

To extend the analytical capabilities of Presto, Uber uses many out-of-the-box functions provided with the open-source software. Presto provides a long list of functions, operators, and expressions as part of its open-source offering, including standard functions, maps, arrays, mathematical, and statistical functions. In addition, Presto also makes it easy for Uber to define their own functions. For example, tied closely to their digital business, Uber has created their own geospatial functions.

Uber

Expanding the Analytical Use Cases of Presto

Uber chose Presto for the flexibility it provides with compute separated from data storage. As a result, they continue to expand their use cases to include ETL, data science, data exploration, online analytical processing (OLAP), data lake analytics, and federated queries.

ETL

While most of Uber's ETL jobs run on Apache Spark, especially those with massive transformation requirements, there are many ETL jobs in which Presto provides sufficient performance for quick execution. For example, some users need to export query results or data sets. Other users need to prepare data as a step before their query or during the execution. These kinds of workloads are not resource-intensive. Presto enables users to aggregate terabytes of data across multiple data sources, transform the data, and export the data in a single query.

Data Science and Exploration

Data scientists have two main classes of queries: exploratory and standardized. Exploratory queries are typically newly written to help the data scientist better understand the data. They often cover large sets of data, in many different types, from many different sources. Uber combines the use of Presto query routing with data sampling to address these requirements.

Standardized queries are more refined and often used to define business metrics and share the results with other decision-makers and frontline workers. They typically query data that is already well-known to both business and data professionals. For these queries, Presto supports common data science interfaces, like Notebooks and Python, which plug into Presto.

For data science use cases, Uber users run ad hoc SQL queries whenever they want, wherever their data resides. Presto allows users to query data where it is stored, eliminating the need for ETL and a specialized data warehouse. Data preparation can be done as part of the single query. Uber uses several different Presto connectors to quickly provision access to data without any additional steps of preparation.

Online Analytical Processing

Using Presto for SQL workloads and online analytical processing is straightforward at Uber. Since support for ANSI-SQL is core to the development of the platform, and since so many data and business professionals have SQL skills, Uber connects their business intelligence reports and dashboards to Presto. This enables frontend business intelligence users to build reports and dashboards across multiple sources without dependence on data engineering and without having to aggregate the data in separate data storage.

Federated Queries

While Uber strives to consolidate most of their data storage, they do collect data in multiple data stores, including HDFS, object storage, Apache Pinot, and MySQL. Presto enables them to run federated queries across all their data sources, including their databases and data lakes on-premises or in multiple clouds. Presto allows them to aggregate answers back in the Presto in-memory analytical processing engine.

Uber

Pushing the Real-Time Boundaries of Presto

Uber has also upgraded Presto to support real-time queries and to run a single query across data in motion and data at rest. To support very low latency use cases, Uber runs Presto as a microservice on their infrastructure platform and moves transaction data from Kafka into Apache Pinot, a real-time distributed OLAP data store, used to deliver scalable, real-time analytics.

This combination supports a high volume of low-latency queries. For example, Uber has created a dashboard called Restaurant Manager in which restaurant owners can look at orders as they are coming in and compare them to historical data. All of this is done on Presto and Pinot. They have created several other similar external-facing dashboards for people outside of Uber to consume.

Even though this real-time application runs on a different version of Presto, both versions support a common set of SQL functions. As a result, users can go back and forth between both versions. A Pinot connector for Presto was recently open-sourced to support queries on Pinot or to join Pinot data with other sources.

Scaling the Upper Echelons of Presto

While Uber has clearly tested the upper echelons of Presto scalability, they have also experienced its limitations. The platform scales well for massive volumes of data, as long as the queries do not become too complex with multiple joins. There are some long-running queries that run out of memory and will not complete.

There are also some limitations around extremely high volumes of queries. Volume is limited by Presto's use of a single coordinator. At this point, the only way around this bottleneck is to deploy multiple clusters. It is for this reason that Uber operates twelve different Presto clusters. Also, the open-source Presto foundation is working on a project called Fireball to enable a distributed Presto coordinator, scale out for high volumes of queries, and eliminate the single coordinator bottleneck.

Scaling Into the Future

Not only does Uber plan to continue using Presto, they also plan to continue making it better. The flexibility of the distributed Presto architecture makes it ripe for extending scalability and broadening the number of use cases.

For example, the Presto query optimizer works well out of the box, but there are some challenges. It assumes that the data is stored in a certain way with correct statistics, proper layering, and populated footers. As a result, there is some work required to format the data to make sure it works with Presto. Uber is currently working on a Parquet writer that takes Presto logic into consideration as the data is formatted.

Uber is also doing some work on technology that will better enable exploratory queries using data sampling. They are building a software component that rewrites queries for a sampling of data as a percentage of the full dataset. The engine automatically knows it is a sampling, so it rewrites the results to reflect the whole set of data.

Uber

In addition, to continue pressing both scalability and real time, Uber is working on a version of Presto that runs on Apache Flink, a framework and distributed processing engine for stateful computations over unbounded and bounded data streams. Facebook is also working on Presto for Apache Spark.

Along with Uber's contributions, the Presto community is vibrant, currently working on new innovations in several different areas.

- Project Aria enables the pushdown of entire expressions to the data source for some file formats, like ORC.
- Project Presto Unlimited introduces exchange materialization to create temporary in-memory bucketed tables to use significantly less memory.
- Apache Pinot and Druid Connectors lower the overall latency for real-time queries.
- Fireball disaggregates the Presto coordinator to scale out for high volumes of queries and eliminate the single coordinator bottleneck.

A Positive Presto Outlook

Presto has already achieved results well beyond what its creators envisioned. Based on what EMA is seeing at Uber and within the open-source Presto project, the platform will continue to expand the number of use cases and improve optimization. The community, backed by Facebook, Uber, and others, plus the commercialization of Presto at companies like Ahana, combine to ensure a future of continued excellence in unified analytics with Presto.

EMA also sees a rapid move toward the concept of a unified analytics warehouse, a unified analytics solution that adequately handles multi-structured data, multi-latency data, and a broad set of analytical use cases from data engineering to data science. Presto meets many of the requirements of the EMA unified analytics vision.

In the words of Girish Baliga, Engineering Manager at Uber, "I love the community of smart engineers at Presto. We are part of a much bigger thing. We can solve any problem that comes our way. Presto has been architected to overcome any challenge we face. We will be able to analyze data in any form, in any use case."

For organizations moving toward unified analytics, Presto makes sense for the leverage of existing platforms and the extension of analytical use cases toward dispersed, multi-structured, multi-latency data. Since not everyone has all of the digital-first software engineering resources like Uber, there are also managed services vendors who have made Presto readily available to any organization for discovery, testing, development, and production.

4093.061421