ST 590 Project 2 – Part 1

Claudia Donahue, Collin Knezevich

Uber is one of the fastest growing companies in recent history. According to Statista, Uber’s total global revenue increased from $100 Million in 2013 to $6.5 Billion in 2016 – a 6400% increase over just 4 years. Today, Uber generates nearly $17.5 Billion in revenue and stores approximately 100 PB of data. As Uber began to grow rapidly, they recognized the need to efficiently store and access a large amount of data. Their initial solution was to use a data warehouse software called Vertica. While this platform was successful in allowing Uber’s many data analysts and engineers to easily access and query data, it presented a number of data problems. Firstly, data reliability was not very good under this platform. The proper schema for data was not formally defined; this combined with ingesting data through ETL jobs often resulted in duplicate data. Additionally, scalability needed to be improved, and this was a major problem given the large amount of data Uber takes in.

To address these issues, Uber switched to an Apache Hadoop-based data lake platform, which is still the basis of their big data operations to this day. The initial switch to Hadoop improved upon these issues with scalability and reliability; however further improvements would still be necessary in the future. Regardless, Uber was in a much better place regarding these issues. A major reason for these improvements is the usage of Apache Parquet, a data file format available in Hadoop that is based on a column-oriented format. Using this allowed Uber to improve its storage via better compression, resulting in better scalability.

Additionally, the Hadoop platform allowed users to use a number of tools to help them easily query and access data through a single user interface. Firstly, Presto allowed users to make interactive queries. Next, Apache Spark assisted users in accessing raw data. This tool is very flexible, allowing users to access data via SQL or via other non-SQL methods. Finally, users could utilize Apache Hive for very large queries. Having all of these technologies available is very important so that individual users will have something to use that is tailored to what they need to query or do.

There were still some limitations with Uber’s big data platform, and they sought to address these by adding on to their Hadoop-based platform. A major problem Uber faced was with data latency. Uber needed to make analytical decisions in real-time, but new and updated data only became available every 24 hours. As a result, ETL jobs were quite slow, as they needed to access the entire table every time. To remedy some of these issues, Uber added Hudi to their big data pipeline. Hudi added support for update and delete operations, vastly improving Uber’s data latency to under an hour. The ability to update data much more quickly improved Uber’s ability to make real-time decisions. Additionally, queries were now much more efficient, and had additional capabilities as well. Users could still query all records in a table at a point in time, but the queries ran much faster. In addition, users could choose to return only new or updated records in a table. Uber added a few supplemental technologies alongside Hudi. Firstly, they added Apache Kafka, which improved upon the storage of changelogs and their associated metadata. Uber also added Marmaray to their platform, which improved the efficiency of data ingestion by helping to avoid ingesting duplicate data. Data transformation is no longer done upon ingestion, but is done through Hadoop by users.

(Will do some more research on how Uber uses their data next).