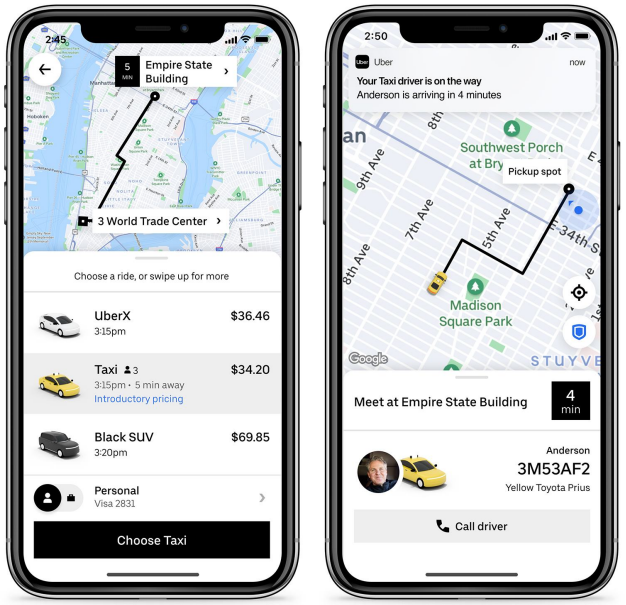
Real-time Data Infrastructure at Uber

Efficient Data Handling for High-Throughput Operations



Follow for more

Sunil Gudivada

Week 01

Real-time Data Infrastructure at Uber

Yupeng Fu
yupeng@uber.com
Uber, Inc

Chinmay Soman
chinmay.cerebro@gmail.com
Uber, Inc

ABSTRACT

Uber's business is highly real-time in nature. PBs of data is continuously being collected from the end users such as Uber drivers, riders, restaurants, eaters and so on everyday. There is a lot of valuable information to be processed and many decisions must be made in seconds for a variety of use cases such as customer incentives, fraud detection, machine learning model prediction. In addition, there is an increasing need to expose this ability to different user categories, including engineers, data scientists, executives and operations personnel which adds to the complexity.

In this paper, we present the overall architecture of the real-time data infrastructure and identify three scaling challenges that we need to continuously address for each component in the architecture. At Uber, we heavily rely on open source technologies for the key areas of the infrastructure. On top of those open-source software, we add significant improvements and customizations to make the open-source solutions fit in Uber's environment and bridge the gaps to meet Uber's unique scale and requirements.

We then highlight several important use cases and show their real-time solutions and tradeoffs. Finally, we reflect on the lessons we learned as we built, operated and scaled these systems.

for tracking things such as trip updates, driver status change, order cancellation and so on. Some of it is also derived from the OnLine Transactional Processing (OLTP) database changelog used internally by such microservices. As of October 2020, trillions of messages and petabytes of such data were generated per day across all regions.

Real-time data processing plays a critical role in Uber's technology stack and it empowers a wide range of use cases. At high level, real-time data processing needs within Uber consists of three broad areas: 1) Messaging platform that allows communication between asynchronous producers and subscribers 2) Stream processing that allows applying computational logic on top of such streams of messages and 3) OnLine Analytical Processing (OLAP) that enables analytical queries over all this data in near real time. Each area has to deal with three fundamental scaling challenges within Uber:

- Scaling data: The total incoming real time data volume has been growing exponentially at a rapid rate of year over year produced by several thousands of micro services. In addition, Uber deploys its infrastructure in several geographical regions for high availability, and it has a multiplication factor in terms of handling aggregate data. Each real time processing system has to handle this data volume increase while

Overview

Importance of real-time data processing at Uber

Key components and technologies used

Objectives of the real-time data infrastructure

Apache Kafka

Used for messaging

High throughput, low latency

Scalability and fault tolerance

Uber customizations in kafka

Cluster federation

Handles peak traffic by distributing load across clusters

Dead letter queue

Helps in identifying and addressing data processing issues

Uber customizations in kafka

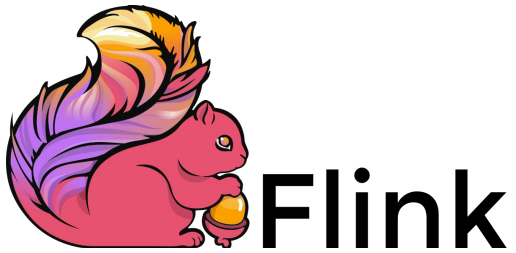
Consumer Proxy

Ensures consistent data consumption across different applications

Cross-cluster Replication

Critical for maintaining service continuity during cluster failures

Apache Flink



Real-time data stream processing

Supports complex event processing

Ensures low latency and high accuracy

Key for real-time decision-making

HDFS

Hadoop Distributed File System)

Long-term storage solution

Handles large-scale data efficiently

Ensures data redundancy and reliability

Integrates with other components for seamless operations

Real-time Analytics

Capabilities

Monitoring and decision-making

Real-time analytics and dashboards

Operational efficiency and quick response times

Benefits and Challenges

Benefits

Improved operational efficiency

Real-time insights and actions

Scalability to handle Uber's data volume

Benefits and Challenges

Challenges

Managing data consistency

Ensuring low latency

Handling system failures

Conclusion

Summary

Uber's infrastructure supports its extensive operations

Integration of Kafka, Flink, and HDFS

Continuous improvements for efficiency and reliability