Apolline Chartier Kastler
Mayeul Bourillon
Judicaël TONGSI KAMWA

ING5

Big Data Ecosystem

Study case : Uber System Architecture

ECE PARIS · LYON
ÉCOLE D'INGÉNIEURS

December 31th 2021
Gr02

Big Data & Analytics

Table of contents :

1)Introduction

Launched in 2009, Uber is now the leading global carshare application. The application, available both on Android and iOS phones, connects a client with a driver, thanks to GPS localization. Uber also takes care of the payment treatment itself, the amount of the fare is debited on the client's account, then Uber receives 5% to 20% depending on the initial amount and it's finally transferred to the driver's account.

Uber faces Big Data issues as it includes 93 million monthly active users and daily trips in 71 countries (2021). The data generated is tremendous and we are going to study the problematics and solutions the company is facing. We will finally propose a solution for Uber infrastructure.

2)Data issues

a) Scalability

We want to minimize the response time and we also want to know how much data we can store.
The database should be horizontally scalable, this way we can always add more servers.

b) Availability

Drivers send data every 4-second. The infrastructure must be highly available.

c) Real-time processing

Real-time processing is used to process captured data in order to produce real-time reports. Real-time processing can then be defined as an unbounded stream of input data with very strict latency criteria for processing seen in milliseconds. This data usually has very short processing times but places more demands on the processing than batch processing. The data sources are often data in an unstructured format.

A real-time data processing architecture consists of the following logical components:

- **Real-time message ingestion**. It provides a means of capturing and storing messages in real time.
- **Processing the data streams.** Once captured the data is filtered and prepared for analysis.
- **Analytical data shop**. This phase prepares the data so that it is in a structured format to facilitate analysis.

- **Analysis and reporting**. The data is analyzed and reports on the data are produced.

3)Uber's infrastructure

a) Monolithic architecture

From 2009 until 2014, the framework was coded in Python. By then, the Data collected was in the range of several Terabytes, so it could work with a single database and a back-end and front-end service.

b) Service-oriented technology

Since 2014, due to the exponentially growing amount of data collected, a Big Data solution has been implemented. This data-driven solution ensures at the same time scalability, reliability and efficiency of data processing. This service-oriented architecture matches the demand (the user) with the offer (the driver).

We are going to study 3 different technologies used by Uber :

- **Apache Kafka.**

Kafka is a stream-processing software platform. It is used as a message queue, it can process up to millions of messages per second. Kafka is chosen for its horizontal scalability and high performance.

- **DISCO.**

DISCO helps to minimize ETA (Estimated Time of Arrival) and waiting time. To do so, it uses Google S2 library which divides the city map into tiny cells. Each cell has a unique ID. When a user requests a ride, DISCO searches for drivers in the concerned cell. DISCO calculates and classifies distances between the user and the different drivers in the cell.

- **Google Cloud Pub/Sub**

Google Cloud Pub/Sub is a messaging service for exchanging event data among applications and services. By decoupling senders and receivers, it allows for secure and highly available communication between independently written applications. Google Cloud

Pub/Sub delivers low-latency/durable messaging, and is commonly used by developers in implementing asynchronous workflows.

4) Our infrastructure proposal

The following tools are solutions for real-time data processing in Azure.

a. **Real-time message ingestion**

Azure Event Hubs. Azure Event Hubs is a messaging solution for ingesting millions of event messages per second. This captured data can be processed by many clients in parallel.

Apache Kafka. Kafka is an open source data flow processing and message queuing application capable of processing up to several million messages per second from different producers and routing them to multiple consumers.

b. **Data storage**

Azure Storage Blob or Azure Data Lake Store containers allow files to be stored in a data shop.

c. **Data stream processing**

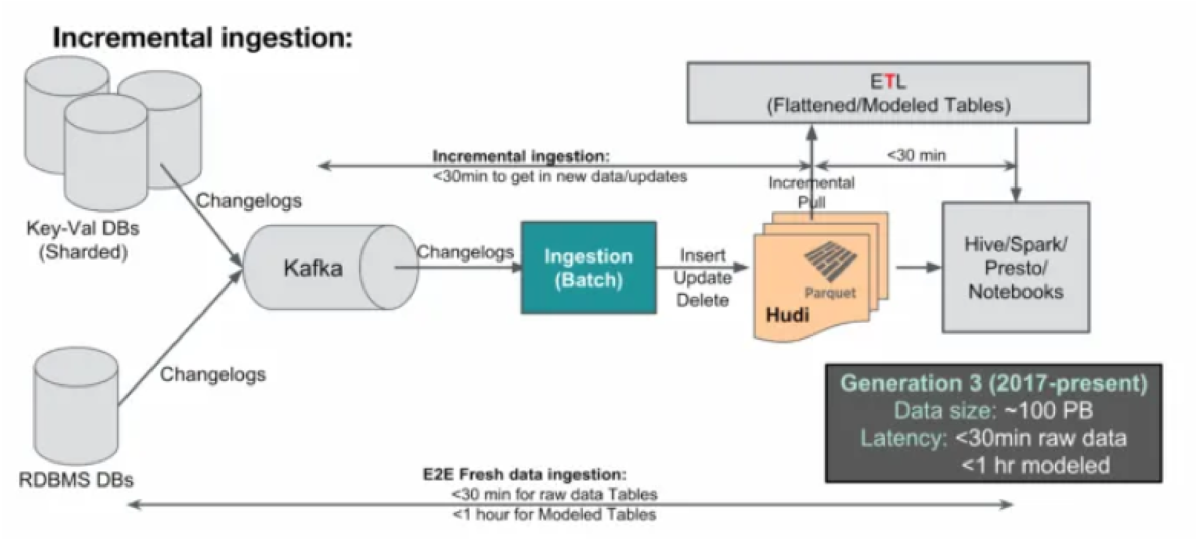
Spark Streaming is an API included in the Spark distributed platform allowing to write code in any language supported by Spark such as (java, scala, python).

d. **Analytical data store**

Spark, HBase, Hive or Azure synapse Analytics are used to store processed data in real time. Spark and Hive are solutions seen in class that allow data to be stored as a file and queried.

e. **Analysis and reporting**

Azure analysis and power BI allow analysis and reporting on data processed in real time. Power BI allows you to publish visualizations and reports in real time.


Incremental ingestion:
Key-Val DBs (Sharded)
RDBMS DBs
Changelogs
Changelogs
Kafka
Changelogs
Ingestion (Batch)
Insert
Update
Delete
Parquet
Hudi
ETL (Flattened/Modeled Tables)
Incremental ingestion:
<30min to get in new data/updates
Incremental Pull
<30 min
Hive/Spark/ Presto/ Notebooks
Generation 3 (2017-present)
Data size: ~100 PB
Latency: <30min raw data
<1 hr modeled
E2E Fresh data ingestion:
<30 min for raw data Tables
<1 hour for Modeled Tables