

Google Analytics like Backend System Design

By Xiao Huang

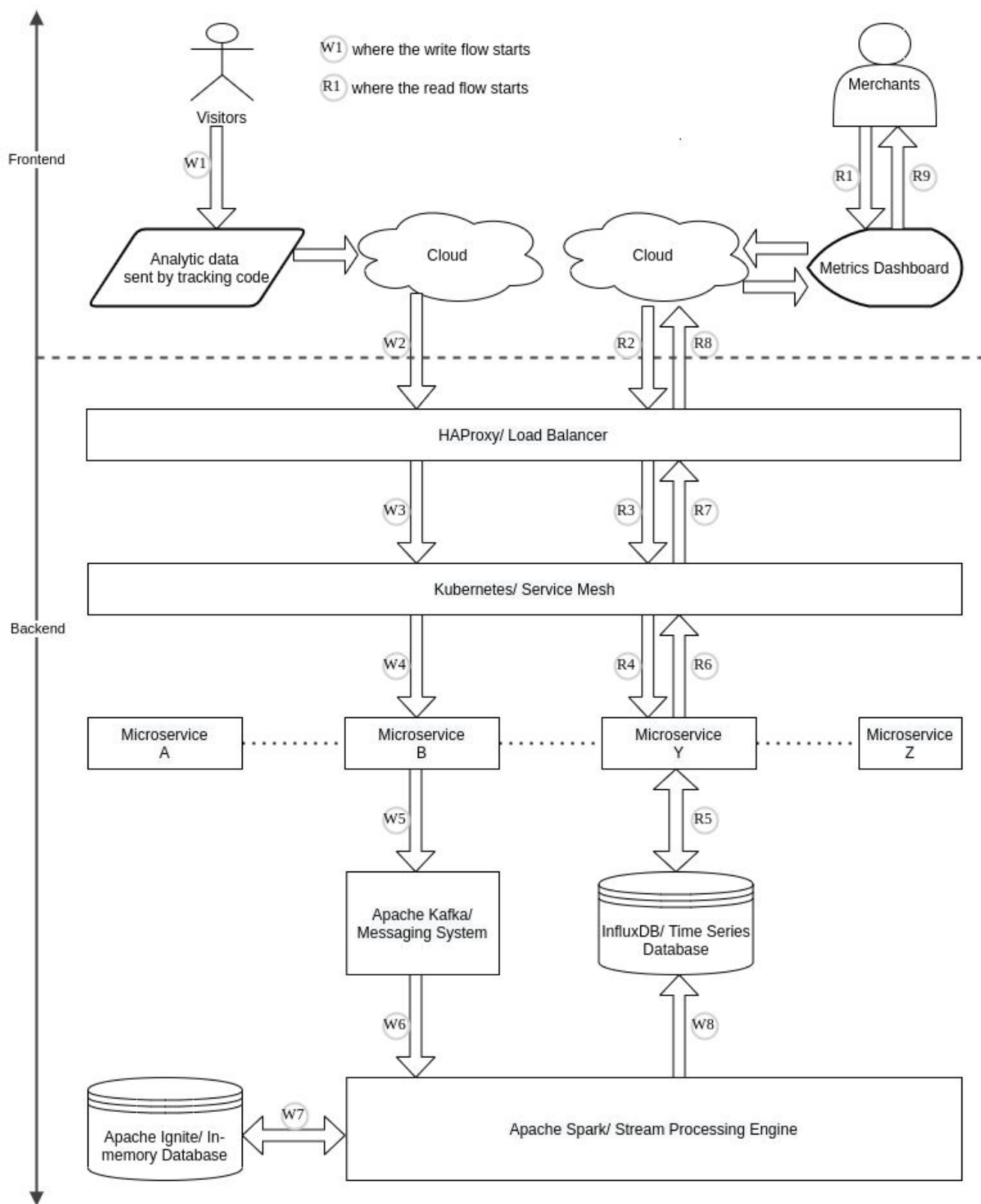
As the system need to handle the large volume of writes and reads with minimum downtime, microservices are implemented and deployed within Kubernetes cluster to provide scalability, resilience, fault-tolerance and high availability.

Furthermore, distributed, scalable, high-throughput, fault-tolerant stream processing engines, like Apache Kafka, Apache Spark Streaming, are implemented to ingest and process the large volume of data with low latency, and finally save the time-series related metrics within the time series database InfluxDB.

Additionally, in case of bugs in the processing logic, historical data stored within Apache Kafka topics can be reprocessed when Kafka log retention and cleanup policies are properly configured.

Below is the high level solution design for the Google Analytics like Backend System.

System Design Diagram



Tracking Code

The write flow starts from W1 on the frontend. Visitors access web sites tracked by tracking code, which loads an asynchronous script to assign an unique client ID. Hits containing all required analytics data are sent by tracking code through XHR requests to the backend system as shown within the write flow W2.

Metrics Dashboard

The read flow starts from R1 on the frontend. Merchants access the metrics dashboard, which sends the requests to the backend system as indicated within the read flow R2.

HAProxy/ Load Balancer

HAProxy is a popular open source software TCP/HTTP Load Balancer and proxying solution.

It routes the requests from the frontend to the Kubernetes service mesh and perform load balancing across Kubernetes service mesh to improve performance and reliability.

In addition, the high availability is provided with Keepalived.

Kubernetes/ Service Mesh

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

It provides primitives for provisioning, running, scaling and managing distributed systems.

Moreover, distributed tracing, centralized logging, centralized metrics, etc. can be implemented with Zipkin, EFK stack, Prometheus and Grafana, etc. respectively.

Microservices

Microservices are deployed within Kubernetes cluster for scalability, resilience, fault-tolerance and high availability.

On the write flow W4 and W5, it ingests the analytics data from the frontend tracking code to Apache Kafka topics.

On the read flow R4, R5 and R6, it accepts the requests from the frontend metric dashboard, then queries the time series database InfluxDB, and finally return the metrics to the front metric dashboard.

Apache Kafka/ Messaging System

Apache Kafka is an open-source distributed event streaming platform used for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.

It ingests all the analytics data sent by the frontend tracking code via microservices. Its log retention and cleanup policies can be configured properly to keep the analytics data for the enough time period so that we can reprocess the analytics data in case of bugs in the processing logic.

In addition, Apache ZooKeeper is required for leader election, metadata storage, etc.

Apache Spark/ Stream Processing Engine

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

As shown within the write flow W6 and W8, analytics data is ingested from Kafka topic, and is processed using complex algorithms expressed with high-level functions like map, reduce, join and window. Finally, processed data is pushed out to the time series database InfluxDB, from which the read flow can later query.

In addition, as indicated within the write flow W7, the performance of Apache Spark applications can be accelerated by keeping data and state in a shared Apache Ignite in-memory cluster, which is a horizontally scalable, fault-tolerant distributed in-memory computing platform for building real-time applications.

InfluxDB/ Time Series Database

InfluxDB is an open-source time series database (TSDB) and is optimized for fast, high-availability storage and retrieval of time series data in fields such as operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics.

It's perfect for storing time-series related metrics generated by Apache Spark Streaming(see the write flow W8) and retrieved by the frontend metrics dashboard via microservices(see the read flow R5).