PaytmLabs SDE Challenge – Design Question

By Anand Soni

Abstract: This document discusses various designs Consideration in designing the backend system and concludes with architecture design for Google analytic like backend System.

Problem Statement: Design Google Analytic like backend system, which supports the following operations: -

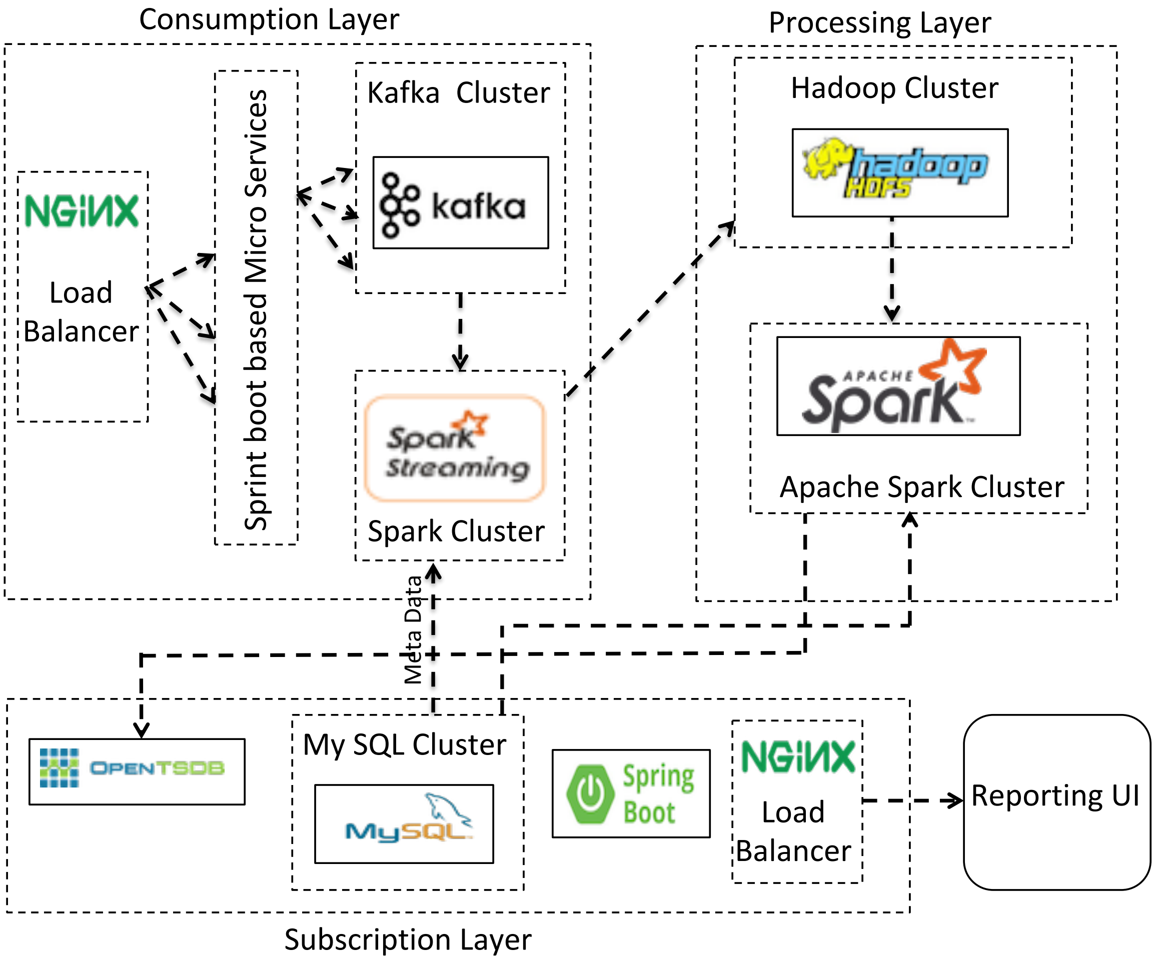
1. Handle large write volume: Billion of write events per day.
2. Handle large query volumes, approximately millions.
3. Read/Query patterns are time-series related metrics.
4. Provide metrics to customers with at most 1-hour delay.
5. Minimum downtime.
6. Re-process historical data in case of bugs in the processing logic.

Architecture

I have suggested 3 layered processing Architecture.

- Consumption Layer.

- Processing Layer.
- Subscription Layer.



Consumption Layer

This layer is responsible for capturing various metrics of the customer. Some of the capabilities for this layer are:

a. This layer should support the capturing of millions of customer metrics.

b. This layer should provide capability re-processing metrics in case of bugs.

c. This layer should allow high volume data Ingestion (Ingestion of various metrics).

Technology on this layer

Assumption

- It is assumed that all customers will have Javascript enabled platform, each page of the platform will embed the Javascript, which will interact with the Rest API of the Consumption layer.

Technology selection

- I have used NGINX as the load balancer to support the following operations:
    - Distribute the load coming on the REST API component.
    - Based on the IP address respective NGIX server will handle the request. This allows us to enable zero failure of the incoming data, in case the other one will handle one NGIX server will be down it.
- Microservices supporting REST API will be used. These microservices will be developed in spring boot and these are responsible for pushing messages into Kafka Cluster.
- I decided to go for Kafka Cluster because:
    - It supports high throughput
    - Capability to store data for as many days.

        - Rewind & Replay capabilities to handle reprocessing in case of a bug.

- For Data Ingestion, I am suggesting either of 2 Options
  a. Directly using Kafka Connectors to directly inject the Messages into Hadoop Cluster.
  b. If we will not give the good performance we will introduce scheduled based Spark **streaming** to pull the messages from Kafka Cluster and push them into the Hadoop cluster.

Based on the benchmarking among the two we will have to decide one of the data Ingestion Strategy.

Processing Layer

Processing layer has the following responsibilities:

a. Define Archival and Maintenance policies around data.
b. Enable time series data point's generation.
c. Support any kind of business logic operation around the data, if needed.

Technology on this Layer

a. The Suitable choice to store such massive data and run analytics should be a clustered environment. For our consideration we have select Hadoop cluster as:
   1. This is open source distributed processing software available on an economical cost.

    2. Easy configuration leads to less time to market of the product.
    3. Integration with other Massive data processing tools like Spark makes it a good choice for the need.

b. Another alternative suggestion if we are looking for Infinite compute and infinite storage capability could be Snowflake. However it is not open-source. Snowflake per TB per month storage cost is $23, they charge for the compute capability. **Snowflake** comes with a public cloud setup which makes environment maintenance very simplified .We don't have to worry a lot about the infrastructure. **Snowflake** makes cross-environment data sharing very simplified. A Point in time rollback feature of **Snowflake** would complement us to adhere to the requirement to re-process and rewind the data

c. For all the time series based analysis and metrics generation, I have selected OpenTSB as it is open source supports clustering.
d. OpenTSB supports Integration with Hbase, hence all the time series analytics would be done in Open TSB sitting on top of Hbase.

Subscription Layer

This layer has following responsibilities:

a. Allow user to query generated analytics.

b. Allow user to creates reports of this analytics.

Technology on this Layer

a. All the client-specific meta-data will be stored in SQL enabled environment; I have selected MySQL as it supports clustering, which will help us scale up our solution.
b. Delivery will happen over the REST API developed using Spring Boot for faster development. To make this layer fault tolerance make these nodes sit behind a Load balancer like NGINX as described in the Consumption layer.
c. We could allow user to view the reports on BI platform, an unfortunate platform like ClicData/Power BI offers free solution up to limit of 1GB, for any solution viewing beyond that will be costly.

Assumption:

Sizing estimate and capacity limits are not suggested in this document as they could only be suggested after some benchmarking.

References:

All the links that has influenced the above architecture are mentioned below

- *NGINX: https://nginx.org/en/*
- *Comparison between load balancers: https://techinplanet.com/best-open-source-load-balancers/*

- *Kafka: https://kafka.apache.org/*
- *Snowflake: https://www.snowflake.com/*
- *Hbase : https://hbase.apache.org/*
- *Hadoop: https://hadoop.apache.org/*
- *Spark : https://spark.apache.org/*
- *OpenTSB: http://opentsb.net/index.html*
- *MySql : https://www.mysql.com/*
- *SpringBoot: https://spring.io/projects/spring-boot*
- *Comparision between OpenSource BI tools :*
- https://opensource.com/business/16/6/top-business-intelligence-reporting-tools