Design Question -- Google Analytic like Backend System

Requirements

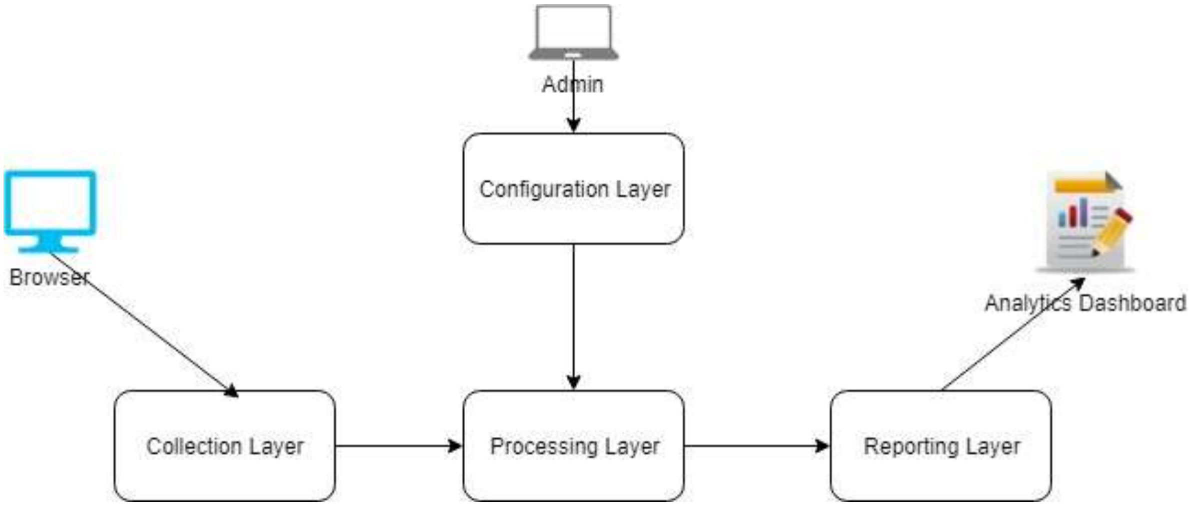
1. Handle large write volume: Billions of write events per day.
2. Handle large read/query volume: Millions of merchants wish to gain insight into their business. Read/Query patterns are time-series related metrics.
3. Provide metrics to customers with at most one-hour delay.
4. Run with minimum downtime.
5. Have the ability to reprocess historical data in case of bugs in the processing logic.

Logical Layers

The entire system can be broken down into the following components:

1. Collection Layer – In this layer, the user interactions with the website is recorded and sent to the processing layer for processing.
2. Configuration Layer – This layer is responsible for letting the end user/admin manage how the data is processed.
3. Processing Layer – This is the main logical component that is used to process the data collected from the client websites.
4. Reporting Layer – This is the API Gateway layer that will serve API requests from the end user to view processed data. It is also responsible for acting as a reverse proxy for load balancing and DDoS protection.

High Level Diagram



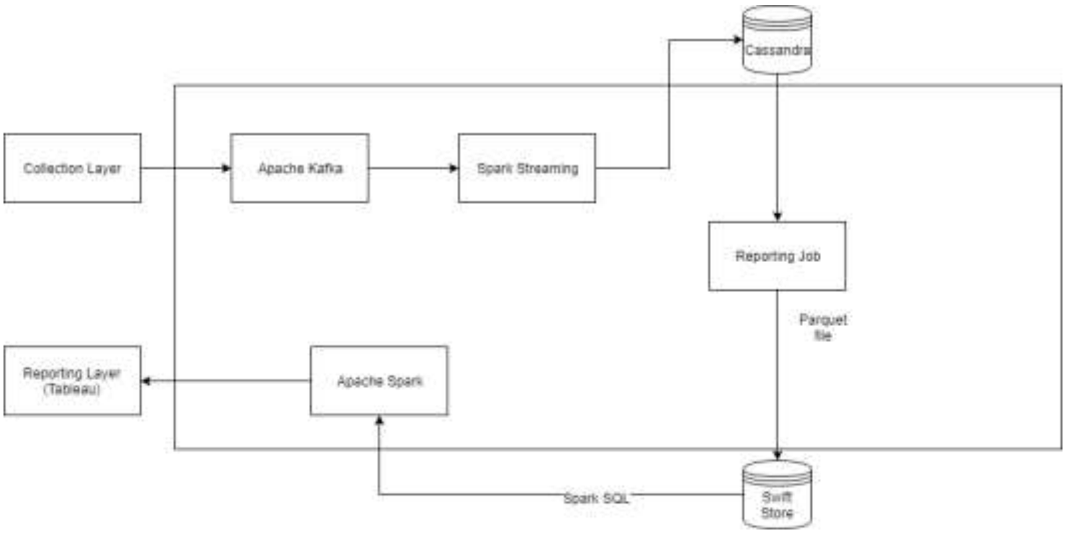
Collection Layer

Whenever a visitor visits the website, the webserver returns the webage along with JS tracking code that extracts and updates the cookies for the user. In addition to cookies, it also records and sends HTTP request for any webpage and other details such as the type of REST call made, browser information, hostname, referrer, etc. This information is sent back to the Processing layer. A measurement protocol is applied before data is sent to downstream services. In our design, we will use the collection layer will push data to Apache Kafka. Apache Kafka was chosen because it is open source and Kafka is a distributed, scalable fault-tolerant messaging system which by default provides a streaming support. Alternately, we could use AWS SQS for this purpose but the downside is that it is not open source and would cost $0.40 per million requests.

Configuration Layer

The configuration layer will consist of express server running NodeJS to serve requests from the configuration website. The configuration layer will set configuration information in the processing layer for processing and reporting data.

Processing Layer



The processing layer will consist of Apache Spark Streaming. It will be responsible for processing a batch of data from Kafka topic. The data will be cleaned and persisted in Apace Cassandra. Just like Apache Kafka, Cassandra is distributed, fault-tolerant and highly scalable.

When data is queried, Apache Spark is used for querying, joining and aggregating data from Cassandra tables. Apache Spark was chosen for both processing data from Cassandra and processing data from Kafka because Apache Spark achieves high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.

Spark batch job are scheduled to run every hour which read data from table in Cassandra and write aggregated data in swift storage as parquet format.

Reporting Layer

The reporting layer will leverage Tableau for visualization. Tableau will be connected to the Swift storage and the queries will be made by Spark SQL.