Problem statement:

Design a Google Analytic like Backend System. We need to provide Google Analytic like services to our customers. Pls provide a high level solution design for the backend system. Feel free to choose any open source tools as you want.

The system needs to:

i. handle large write volume: Billions write events per day.

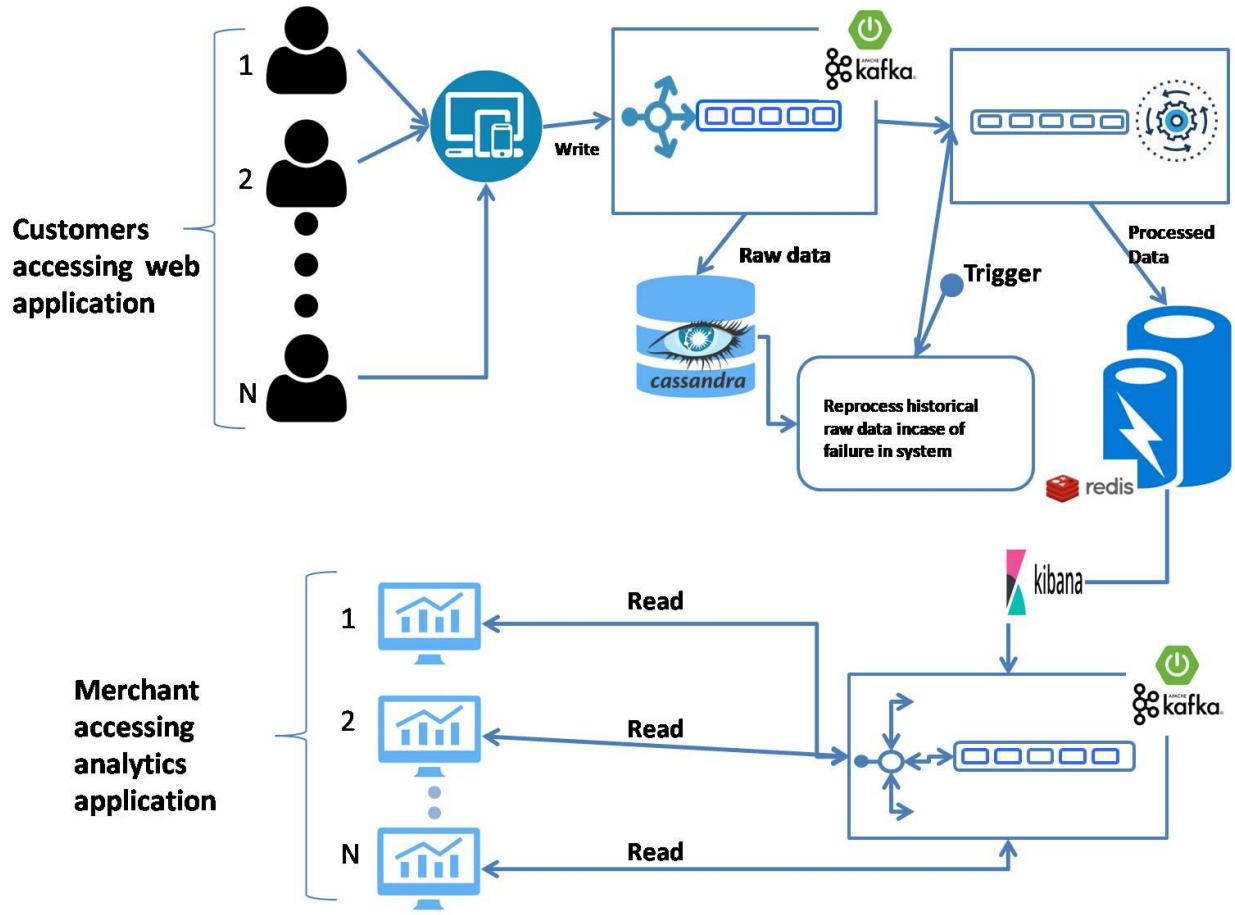
ii. handle large read/query volume: Millions merchants want to get insight about their business. Read/Query patterns are time-series related metrics.

iii. provide metrics to customers with at most one hour delay.

iv. run with minimum downtime.

v. have the ability to reprocess historical data in case of bugs in the processing logic.

High Level Design Solution Design:



Explanation:

- N users are accessing the web application page, the events recorded from each user are sent to the messaging system (Apache Kafka used here) via collectors (example: http).
- The load balancer pushes the event based on the traffic to the queue.
- Each event de-queued from the input queue is stored as raw data in Cassandra data base.
- Now, if there is a failure / need to replay the historical data, a trigger can be used to reprocess the raw data and store in the Redis data base.
- The raw data is passed for aggregation in the aggregation queue.
- After the data is being processed it is stored in the Redis data base.
- When N merchants are requesting for the analysis of the events by different customers, the processed data stored in Redis database uses Kibana to display the analysis on the screen.

Handle large write volume

- To handle large data write, I have used Apache Kafka along with Cassandra database
- Cassandra database follows multiple master nodes which mean that there is no dependency on a single node and the write throughput is high.

Handle large read/query volume

- To handle large read/query volume, I have used Redis data base.

- Since the data is being processed and updated frequently for real time analysis, redis is the perfect choice.
- It has cache memory available too, due to which reading data becomes fast.

Provide metrics to customers with at most one hour delay

- The retrieval of data from Redis is very fast, so providing the metrics using Kibana tool will also be quick.

Run with minimum downtime

- Apache Kafka is used so there is no down time with a performance of 1 million/sec in a single server.

Have the ability to reprocess historical data in case of bugs in the processing logic

- Using Apache Kafka is a log structured system. So when we require reprocessing, we retrace the message last written by Kafka Producer and reprocess the raw data available in the data base.