Architecture Diagram and Explanation for Streaming Twitter Data in Real Time and Performing Analytics Over it

Use Case Understanding

A community by definition goes as, *"A community is a social unit with commonality such as norms, religion, values, customs, or identity. Communities may share a sense of place situated in a given geographical area or in virtual space through communication platforms"*. By properly understanding this definition it can be understood that a community is a group of people from versatile backgrounds but share the common interest, often because they know each other directly or because of mutual contact. People in the community communicate or interact with each other by various means, either by talking with each other or by sharing many things. Community can exist either physically or virtually.

While considering the existence of the community online, it has to be understood that the sharing, interaction, communication happens at altogether different levels. Following are the activities that happen as a part of community engagement:

1. Sharing different posts, may be photos or videos

2. Creating groups within a community and sharing messages or images or videos or documents or music.

3. Directly connecting with other members by personal messaging (one-to-one communication).

4. React to the posts shared by other people either by liking it, sharing it on their timeline or commenting on it.

5. People can even create their own new community and add new folks that are relevant to it.

Architecting Data and Analytics Pipeline

In the above section, we have got the understanding of the social community use case in detail. There are many social media platforms that are present today in the digital space and processes petabytes of data. Few popular platforms to mention but not limited to are Facebook, Twitter, Instagram, LinkedIn, etc. Here many people connect with each other and interact and communicate in many ways like Direct Messaging, Group Messaging, Video Calls, Community posts, Shares, Likes, Comments, etc.

In our use case, we consider Twitter as the social media platform and data is streamed from Twitter Timeline. Every second on these platforms millions and billions of activities happen and many people keep on performing a lot of activities. It has to be established that a lot of data is processed every second and data engineering needs to be done efficiently. When we say efficient Data Engineering, we mean that processing of data needs to be done in such a way that following assumptions are fulfilled:

1. Data Streaming when done, data should not be lost.

2. While streaming data, messages or data is broken down into chunks to process faster. These messages should not lose the order while they are received at the receiver end.

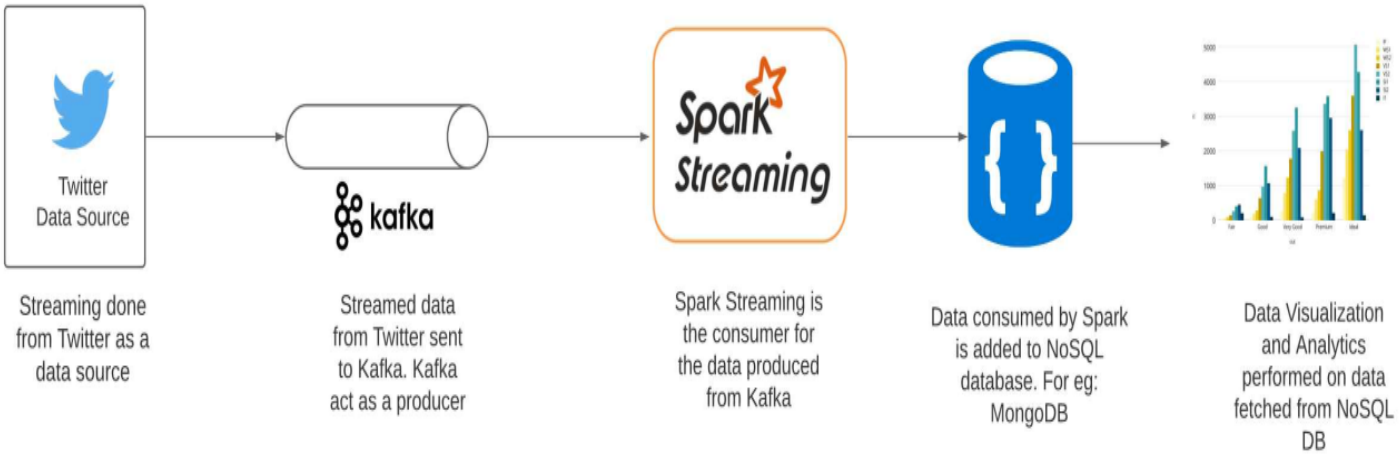
3. Stream data should be pre-processed and cleaning and transformations should be done such that relevance is established on the data.

4. Since the data is unstructured, storing such data in relational database management system would be improper and operations would not be able to performed on that data

5. Data is stored in NoSQL databases like MongoDB, Cassandra or DynamoDB. In our case, we are planning to use MongoDB as our database. This stores the streamed, cleaned, relevant and processed data.

6. Based on this data from NoSQL, different analytics could be plotted. Based on different parameters like hashtag tweets can be analyzed and sentiment analysis could be done. Another instance, based on the location of the tweets, it could be understood what exactly is the trending topic for a particular geography or a country.

Architecture Diagram



Architecture Diagram

Detailed Description of the Application Components

1. **Data Source:**

   We are using Twitter as our data source. We use the HorseBird Client (HBC) Twitter library to stream tweets from twitter. These tweets are then fed to Kafka producers.

2. **Kafka Producer:**

   Apache Kafka is used to publish the message tweets that are fetched from the twitter source. Apache kafka acts here as the ***producer*** that stores the messages that are streamed directly from Twitter.

3. **Apache Spark Streaming:**

   Apache Spark Streaming is the library built on the top of Apache Spark. This library is used to stream data. Here Apache Spark Streaming acts as the ***consumer*** that consumes the message from the producer Kafka. Messages consumed at the Apache Spark Streaming can be used to feed to RDDs and then transformations can be applied accordingly for the same.

4. **NoSQL Datastore:**

   From Apache Spark Streaming, RDDs are created. RDDs created as an output from Spark Streaming are then stored in a NoSQL database. NoSQL data is preferred here as the tweets data is unstructured and thus it cannot be stored in a relational database management system. As a part of this assignment we are exploring multiple NoSQL databases, out of which we believe we shall go with MongoDB.

5. **Data Visualisation and Analytics**

   The data being stored in the NoSQL database, needs to be analyzed to derive useful information and relevant inferences need to be drawn out of it. Hence plan to use this data from the database as a datasource for plotting graphs and performing insightful visualizations. For visualisation part, we are exploring two JavaScript libraries, namely HighCharts and Vue.js