Project Proposal
Distributed streaming platform

1. Definition

Implementation of a distributed data pipeline to process real-time streaming data, coupled with search engine and sentiment analysis. The design majorly focuses on providing large-scale data processing with low latency and fault-tolerance.

2. Justification

Increasingly people are turning to Twitter to express their views on various issues faced on a daily basis. 335 million users generating data at unprecedented scale expressing their views; some good, some bad. We are interested in using this real time tweets to find out if people are spreading positivity or negativity. Is positive tweets retweeted more or negative tweets retweeted more. And also isolate by region what is the overall mood of the people.

In trying to understand the sentiment of the tweets, we face the issue of processing large streams of real time data. Ingesting, processing and storing the data introduces new problems. And this would give us a chance to understand and build a pipeline to process the data and find out relevant information.

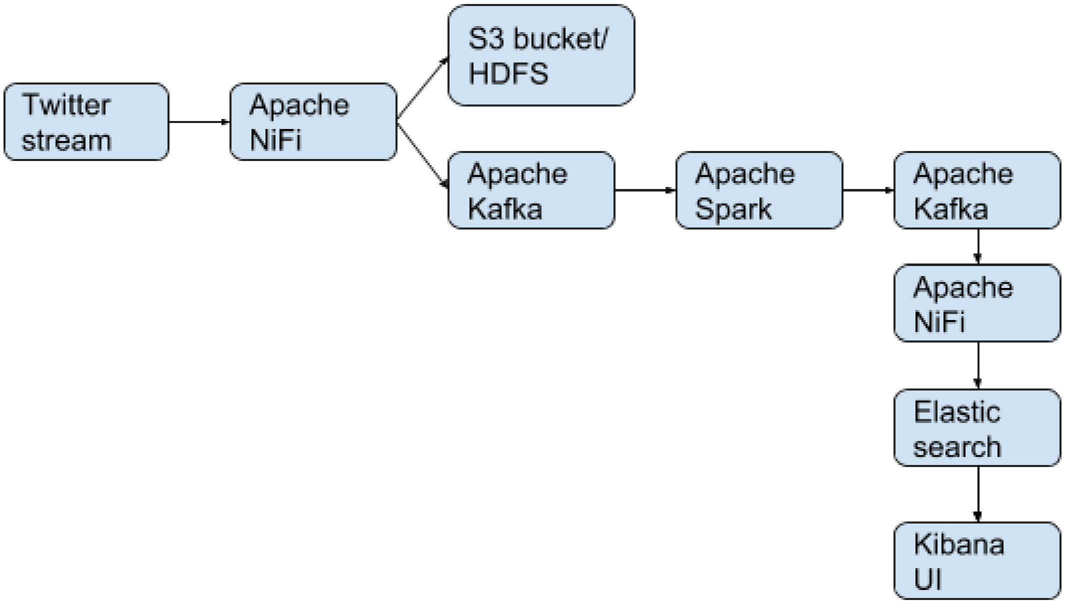
3. Overview

This project is focused mainly on building a data pipeline to process a stream of real time data to extract useful information from the raw data streams. The extracted information would be made available for visualization tool to search and visualize the required information.

This project would use real-time twitter data obtained by twitter API as data source. The twitter stream is passed through Apache NiFi to create a constant rate steam which the later components can process. Spark is used to handle this extra workload, for data enrichment and data transformation in the most optimized way possible. The transformed data would be used to perform sentiment analysis, co-occurrences and extraction of other useful information. This information is fed into Elasticsearch and Kibana to visualize the data.

This entire data pipeline would be tested under various loads to verify the robustness of the pipeline. The pipeline would also be verified for the fault tolerance and constant time processing.

4. Architecture



Twitter data is fed into NiFi which facilitates data routing and transformation. We will be using NiFi GetTwitter processor which pulls status changes from Twitter's streaming API. The NiFi GetTwitter API would be called every 1s in our case. The data from NiFi is pushed into Kafka, is a messaging queue which would serve as a buffer upon crash.

The data from NiFi could also be stored in S3 buckets to enable access to systems outside the organization, or in distributed file system like HDFS for batch processing.

Spark pulls data from Kafka and performs data cleansing, additional transformations if necessary. Analysis such as trending hashtags could be determined with a sliding window of size 5 mins and speed of 30s, co-occurrences of words, sentiment analysis using NLP library in Spark. Spark pushes the processed data into Kafka, so that multiple applications can consume data directly from Kafka. NiFi pulls the processed data from Kafka and then fed into Elasticsearch, a search engine adapted from Apache Lucene facilitating query by word. Visualization is provided by Kibana UI, a visualization plugin for Elasticsearch data.

5. Timeline

Milestone 1 : Data Source (Twitter streaming API and basic real time data stream generator)
Milestone 2 : Building the pipeline on VCL (Apache NiFi, Kafka, Spark)
Milestone 3 : Building the pipeline on AWS (Kinesis)
Milestone 4 : Data visualization (Elastic Search, Kibana) and stress test (varying loads)

6. Resources

Twitter streaming data -
https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data.html
NiFi GetTwitter API -
https://nifi.apache.org/docs/nifi-docs/components/org.apache.nifi/nifi-social-media-nar/1.5.0/org.apache.nifi.processors.twitter.GetTwitter/
Kafka -
https://kafka.apache.org/documentation/
NLP library in Spark -
https://databricks.com/blog/2017/10/19/introducing-natural-language-processing-library-apache-spark.html
Kibana -
https://www.elastic.co/products/kibana