Unique word analysis through live Twitter streaming

1.0 Purpose

Application depicted in Fig 1.0 will leverage STORM architecture for live stream analysis of twitter streams and highlight the unique words and its associated count.

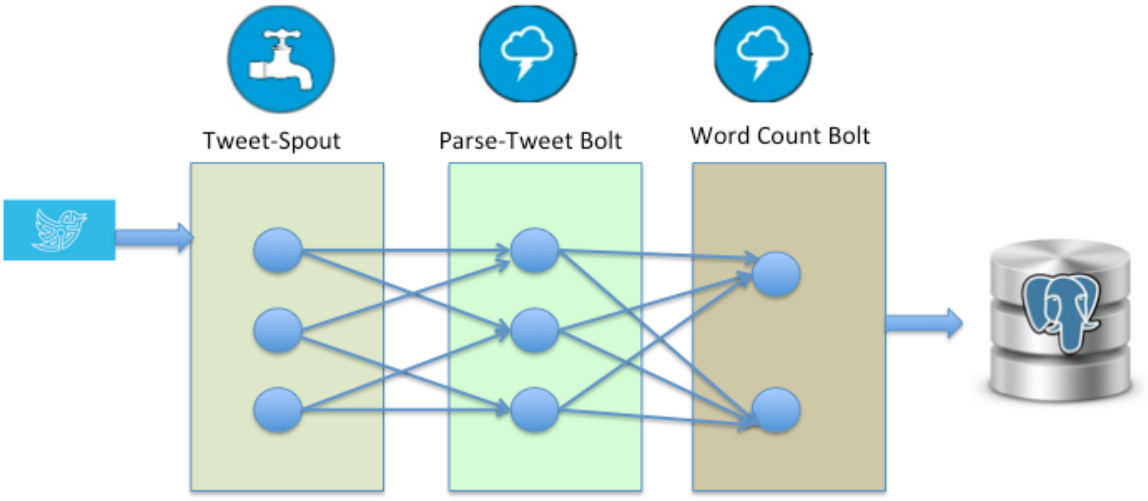

Fig 1.0 Application Topology

2.0 Application Topology

Basic premise of the application is to establish authorized connection with Twitter API's and extract live twitter streaming. Once, the Twitter connection is established and the tweets are extracted real time, all valid words will be extracted from tweets. Application will then establish a connection with already created postgres database in order to store the unique words from tweets and the associated count. Open source Storm architecture is being leveraged to perform the live stream analysis of tweets.

Application architecture can be further broken down into three main premises.

a) **Tweet Spout** – Establish an authorized connection through Twitter API's in order to extract live English tweets. Pass the tweets into the queue to parse-tweet bolt.

b) **Parse-tweet Bolt** – Take the tweets from tweet spout and decompose the tweets into valid words. Parse tweet bolt will filter out words starting with hash tags (#), user mentions (@), retweets (RT), urls (http) and any leading/lagging punctuations. Parse tweet will also check for the words containing only asci

characters. Once a valid word is established, it will then be passed to word count bolt.

c) **Word count Bolt:** Take the valid words from parse tweet bolt and store the data into an already created postgres database. Word count bolt will first establish the connection with "tcount" postgres database. If the bolt encounters the word for the first time, then it will insert a new entry into the "tweetwordcount" table containing both the word and the count. If the word was already encountered then the bolt will update the "tweetwordcount" table for the word with the updated count.

3.0 Dependencies

Following dependencies needs to be satisfied in order to run the application.

1) **"tcount"** postgres database needs to be created before the application can be run.
2) **"tcount"** postgres database will contain the a table called "tweetwordcount" having the following data schema.
    a. **Word –** String field and as the primary key.
    b. **Count –** int field.

4.0 Directories and File Structure

Application will contain two directories, tweetwordcount and scripts where as the former will have all the source code and later will have the python scripts in order to display the results of unique word analysis.

a) **tweetwordcount** directory will have two main sub-directories src and topologies where the src sub-directory will contain the source code for spouts and bolts. Topologies sub-directory will contain the topology file in order to construct the architecture topology depicted in Fig 1.0

b) **scripts** directory will contain three python scripts – finalresults.py finalresultsall.py and histogram.py.

c) **finalresults.py** will take a word as an argument and return the total number of word occurenes.

d) **finalresultsall.py** will run with out an argument and return all the words in the stream and their total count of occurences.

e) **histogram.py** will take two integers k1, k2 and return all the words that their total number of occurences in the stream is more or equal than k1 and less or equal than k2.

5.0