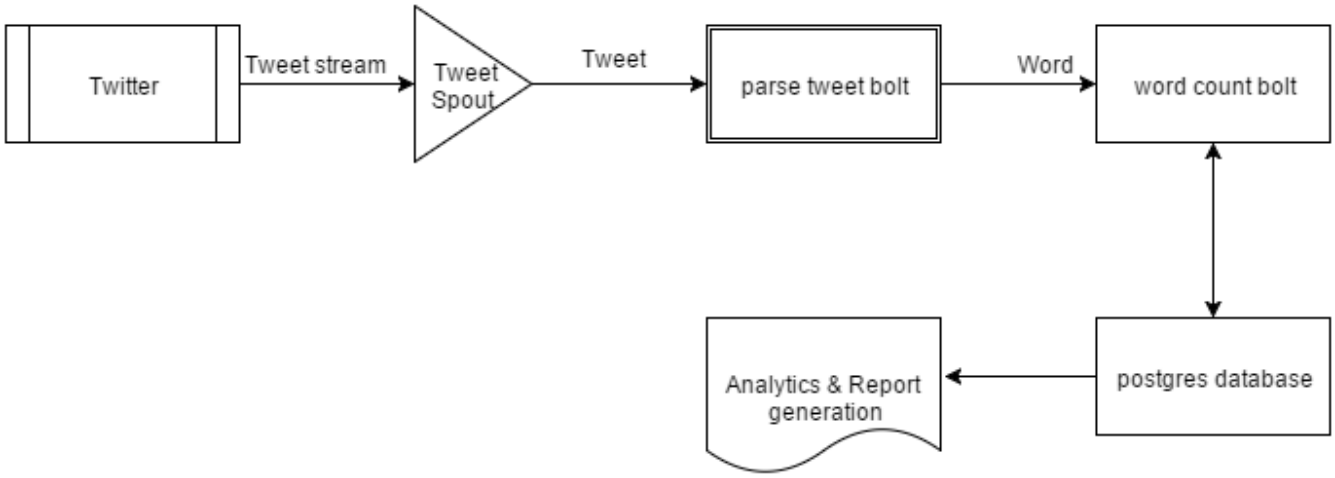
Architecture

Application Idea

This application takes the twitter stream as its input and computes the number of occurrences of each word in the input stream and writes that information to a serving database. The tweets which are essentially sentences are broken down into words and the number of occurrences of each word is then computed. The same idea can be used to process tweets in realtime to determine current trends and insights.

Application Architecture

The application uses storm to process realtime tweets. The storm component tweet-spout listens for streaming tweets and sends them to another storm component called the parse-tweet-bolt that parses the tweets and breaks them down into words. The parse bolt then sends words one at a time as its output to another bolt called word-bolt which maintains a per word counter and updates a postgres database table called the tweetwordcount table. This table was created during the initialization of the tweet-spout. Serving scripts can be used to analyze the words stored in the tweetwordcount table.



Tweet Wordcount Application Architecture

The application runs in the streamparse project EX2Tweetwordcount. The top level project directory has subdirectories for specifying the topology as well as the python source code for the bolts and spout under the src subdirectory. The credentials are stored in Twittercredentials.py.

Directory Structure of this submission

Top level Exercise_2 directory

```
Exercise_2/
    topologies/                 /* Contains Storm Topology */
    screenshots/                /* Contains the screenshots of different components working */
    src/                        /* Contains source changes for the storm spout and the bolts  */
    Serving_Scripts/            /* Contains python scripts to query postgres and gain insights */
     plot.png                   /* Analytics output: A bar chart of the 20 most frequently
                                    occurring words in a recent twitter stream.                 */
     plot.pdf                   /* pdf version of the above doc */
     Architecture.pdf           /* This doc */
     Readme.txt                 /* A readme of how to run the application */
     Twittercredentials.py      /* Credentials to get the tweets */
```

Spark Topology Directory
Specifies the storm topology consisting of one spout and two bolts and the output from each component. This file has a dependency on the sources implementing the bolts and spouts.
Exercise_2/topologies/tweetwordcount.clj

Screenshots Directory
This contains images of a working application,
Exercise_2/screenshots/screenshot-twitterStream.png
Exercise_2/screenshots/screenshot-storm-components.png
Exercise_2/screenshots/screenshot-extract-results-1.png
Exercise_2/screenshots/screenshot-extract-results-2.png
Exercise_2/screenshots/screenshot-extract-results-3.png
Exercise_2/screenshots/screenshot-extract-results-4.png

Serving_Scripts Directory

Exercise_2/Serving_Scripts/histogram.py /* This script takes two integers k1,k2 and returns all the words that their total number of occurrences in the stream is more or equal than k1 and less or equal than k2 */

Exercise_2/Serving_Scripts/finalresults.py /* This script takes a word as an argument and prints the number of occurrences of that word in the input stream. Running without an argument it returns all the words in the input stream and their occurrences sorted alphabetically in an ascending order. */

src directory
Contains the bolts and spouts sub-directories.
Exercise_2/src/bolts/
Exercise_2/src/spouts/

Bolts directory
Contains changes to implement the two bolts.
Exercise_2/src/bolts/wordcount.py /* Inserts/Updates word, count pair to the tweetcounttable */
Exercise_2/src/bolts/parse.py

Spouts directory
Contains changes to create the tweetcounttable table in postgres and the rest of the changes implementing the spout.
Exercise_2/src/spouts/tweets.py