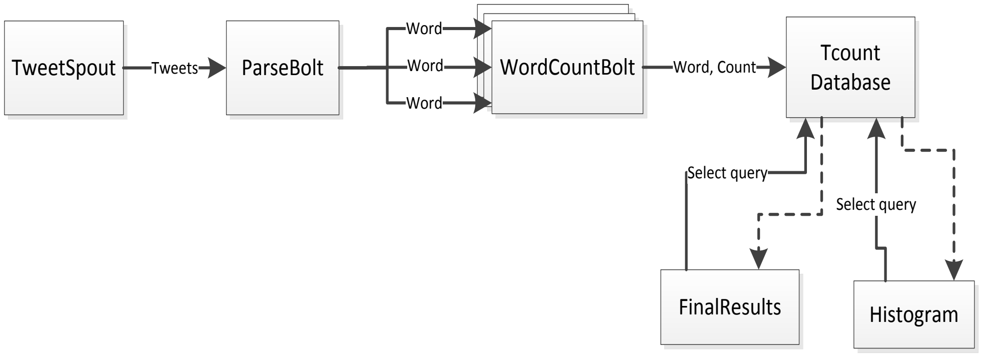
Twitter Storm WordCount Architecture

1. Overview:

This Storm pipeline reads streaming data from the Twitter API, parses each Tweet into a series of words, and then counts the total number of times each word appears. The pipeline maintans word count information in Postgres database called "tcount."

There are also two additional reporting functions called "finalresults" and "histogram." Both of these functions connect to the tcount database and print out the words and word counts that match the input criteria. The finalresults function will return the word count for its first argument. If given no arguments, the function will return the word counts for all of the words in the tcount database. The histogram function takes two integers and returns the words for which the word count is between the first and second arguments.

2. Diagram



3. Deployment

This application has the following dependencies

a. Tweepy

b. Sparse

c. Lein

d. Python 2.7

e. Psycog

This application streams data from the Twitter API and writes data to a Postgres database called tcount. To configure these endpoints, please modify the following files

- src/bolts/parse.py: Please change Twitter access and secret keys to match your Twitter API keys.
- src/bolts/wordcount.py: Please change the connection string username, password, host, and port to point to a running Postgres server with a tcount database.

4. Issues

a. To prevent syntax errors in Postgres statements, the wordcount bolt and finalresults command remove some special characters from incoming words. For example, “I’ll” becomes “Ill” and “PG&E” becomes “PGE”.

b. The tweets spout will sometimes stop streaming tweets and instead throw an EmptyQueueException. Although this behavior is usually intermittent and only lasts a one or two seconds at most, there have been occasions where the behavior persists indefinitely. If this happens, please restart the Storm python process.