**File Directory:**

/src/bolts/parse.py

/wordcount.py

/src/spouts/tweets.py

/topologies/tweetwordcount.clj

/finalresults.py

/histogram.py

/root/wordcount/extweetcount

This is the test suite

**Application Idea:**

This application continuously counts the words emitted by a live stream of tweets. There are a variety of potential applications, including identifying trending topics, identifying when a corporate brand or individual is being mentioned, or measuring the overall mood of the twitterverse (for example how often “happy” words appear as compared to “sad” words).

**Description of Architecture:**

The program runs on a sparse topology, which includes a .clj file directing the execution, a src folder within which are the spouts (in this case three instances of one spout type) and the bolts (three instances of the parse bolt and two of the count bolt). The serving scripts are also located in the main folder, and call on the sparse program in execution.

**File dependencies:**

* 1. Twittercredentials.py is hard coded with access keys for the Twitter app. hello-stream-twitter.py called these during test.
  2. The Spout tweets.py is also hard coded with access information. While not dependent on another file, the code does require that certain programs be imported and called effectively.
  3. parse.py receives tuples from (and therefore depends on the correct execution of) tweets.py
  4. wordcount.py receives tuples from parse.py
  5. The mechanism for interactions b., c., and d. above are controlled by sparse processes detailed in tweetwordcount.cjl
  6. finalresults.py is reliant on the execution of the core program (b, c, d, e), as well as the storage of output in the Postgres table and the correct call from this script on this populated database
  7. histogram.py is reliant upon information from finalresults.py

**To Run the Application:**

An EC2 instance must be created and launched using the correct UCB AMI, and with the appropriate Security Group applied (including port 5248 for Postgres).

A volume of sufficient size must be attached to this instance.

Hadoop and postgres must be installed and available, and psycopg2

Git clone of <https://github.com/carleeprice/W205> must be executed, with the correct file structure (see above) on the instance.

To run the application: from the command line, navigate to folder extweetcount. Execute the command: sparse run. Because the program is continuous, it must be interrupted with Ctrl+C.

Once some amount of information has been populated in the database, again from extweetcount folder on the command line, run: python finalresults.py <word of choice>

Likewise, run: python histogram.py <number of choice> <another number of choice>

Et, voila!