Architecture document

This file describes the architecture for a twitter word stream application.

Directory and file structure:

All executable files live inside the exercise2 folder in the git repository https://github.com/pranaysuri92/W205/tree/master/exercise_2. The file extweetwordcount under exercise_2 has three major components to it: Topology files, src tiles, and executable files. The topology file, in the topologies folder, gives the outline of where the data is passed to for processing. The src folder contains the spout files and the bolt files. The spout files provide the source of streams and the bolts consume these streams and process the data. The data ends up in a postgres database. The executable files link to this database and pull the results depending on the parameters passed to them.

Application idea:

The idea of this application is to collect real time data from twitter, split the data into separate words, and count the occurrences of these words. The results will be stored in a postgres database at an aggregated level: word and count of that word. Then we have two executable files that access this data to either represent this data in a histogram format or show pull results straight out of the database.

Description of the Architecture:

The topology sets the architecture of the system. The data is coming from the twitter data stream and collected by the tweets spout. This is a live stream that will only capture English tweets. The topology then sends the output to the parse bolt. The parse bolt splits these tweets into separate words. It then processes out any special characters and any attempts to clean words of any stray punctuation. After this level of processing is finished the topology sends the output to the last bolt called wordcount. The job of the wordcount bolt is to keep a running count of how many times a word has been caught by our application. This result will be logged into our postgres database with the word as the primary key. The words are inserted or updated to postgres using psycopg.

After the application is running there are two executable files that hit the database directly. The first file is called finalresults.py and the second file is called histogram.py.

The finalresults python script accepts a word as an input. If an input is given then it returns the number of times that word has been used up to that point. If there is no input or if there are multiple inputs then all results will be printed out in alphabetical order.

The histogram python script accepts two integers as input. The script will return all words that have counts that are between the two numbers. The first number acts as the lower bound and the second input acts as the upper bound.

File dependencies/running the application:

Before you can execute any files make sure that you have postgres running and you run psycopg-sample.py located in the exercise_2 folder to set up the database. After this file is running you can run the storm topology in extweetwordcount. After these two tasks are done the executable files (histogram.py and finalresults.py) can be run to access the postgres database. More detailed step by step instructions can be found in the readme file inside extweetwordcount.