Michelle Liu
W205
April 2, 2017

Architecture

Twitter Application Idea

This Twitter application accesses the twitter stream via Python Tweepy library with the user's Twitter API credentials. It streams the live twitter data, using the Streamparse package to parse those tweets and count the number of each word in the stream of tweets processed, and record the result back to the Postgres database. We then write python scripts to query the Postgres database using the psycopg2 library.

Directory and File Structure

Main folder: Exercise_2/

```
tweetwordcount/
        src/
                bolts/
                        _init_.py
                        parse.py
                        wordcount.py
                spouts/
                        _init_.py
                        tweets.py
        topologies/
                tweetwordcount.clj
        virtualenvs/
                tweetwordcount.txt
        README.md
        config.json
        fabfile.py
        project.clj
        tasks.py
screenshots/
Final results.py
histogram.py
plot.png
```

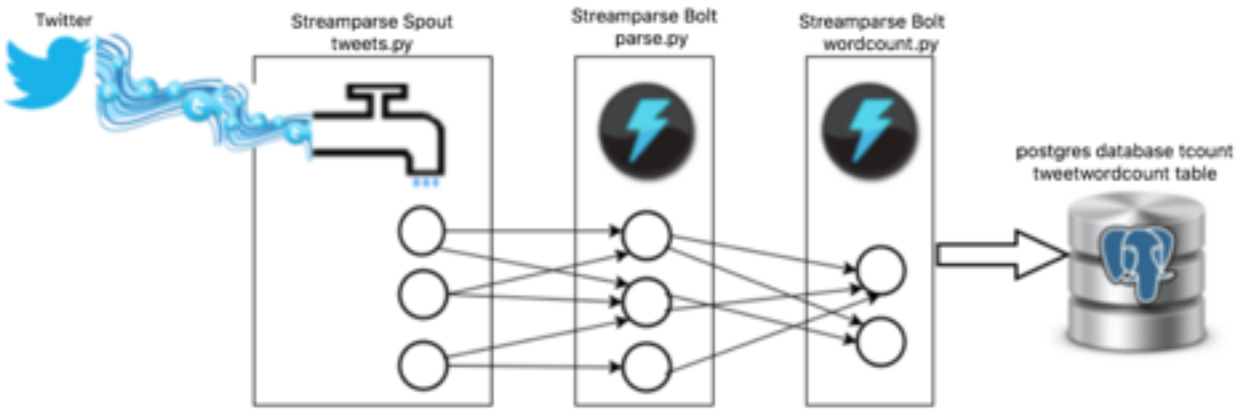
Notable Files and Description:

Mostly everything was cloned from git@github.com:UC-Berkeley-I-School/w205-spring-17-labs-exercises.git.

I modified *tweetwordcount.clj* to implement the topology shown in the assignment instruction. I didn't make any changes to *parse.py,* which takes in emitted tweets and splits them into separate words, parsing out characters that shouldn't be counted as words. In *tweets.py,* I added in my Twitter credentials to connect to the Twitter API. For Github submission, I removed the 4 tokens in the beginning of the file. Note that the end user will have to put in their own 4 tokens for the Twitter API credentials to make this project work. The major changes is in *wordcount.py,* where I connect to the database using the psycopg2 Python library, create and connect to a database *tcount*, and either update or insert to the table *tweetwordcount* with the incoming parsed twitter stream.

I have 3 Python serving scripts that I wrote to query the database and return the specific results as specified in the assignment page. *Finalresults.py* outputs a list of all words and their number of appearances in the database alphabetically if no arguments are inputted. If there is an input, then it displays the count of that word in the database. *histogram.py* produces a list of words that have a count of at least the first number and less than the second number. *top20result.py* outputs a csv file displaying the top 20 words and their counts in the Twitter stream. I later used this csv file in Excel to generate a bar chart, *plot.png.*

Description of the Architecture



(figure taken from the assignment pdf, I just added the description of the topology to show the architecture)

File Dependencies

The code needs to be run on an EC2 instance using the UCB MIDS W205 EX2-FULL AMI. The project needs Psycopg2 (version 2.6.2), Postgres, Python, Streamparse, and Tweepy installed on the instance. The Streamparse command needs to be run in the *tweetwordcount* folder. Read the README.txt file for more direction on how to run.