MIDS W205 - Exercise #2

Chris Bennett

Architecture Document

Application Description

This application captures live tweets and processes them in real-time, and aggregates the results in a Postgres database. This application utilizes the Tweepy library to read the live stream of tweets from twitter. This reading of tweets occurs in the spout (tweet-spout). As tweets are read, they are passed to a bolt process that parses the tweets and extracts the words from each tweet (removing some words and punctuation). This parsing bolt (parse-tweet) passes the words to the next bolt (count-bolt), which counts and loads the words and word counts into Postgres – into the Tweetwordcount table in the Tcount database.

Directory and File Structure

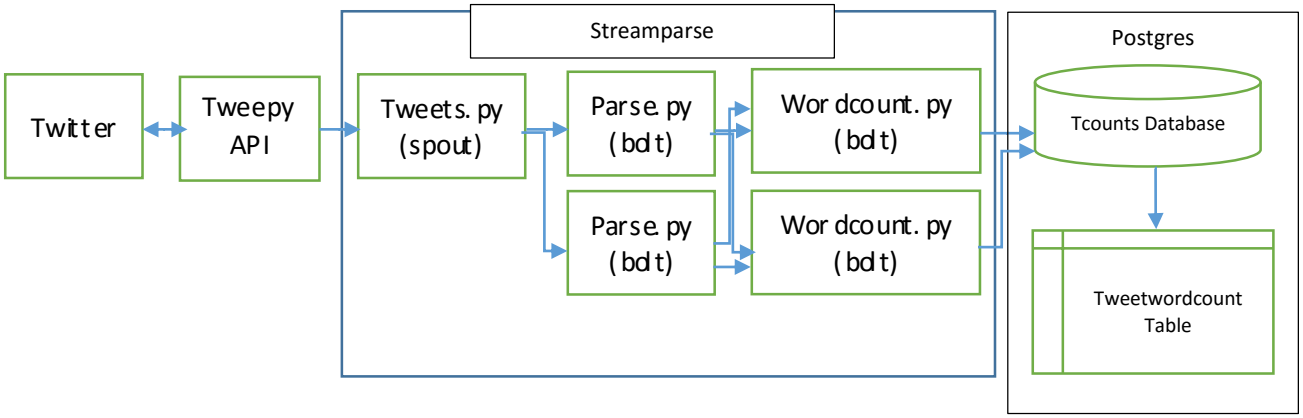
```
w205@ip-172-31-4-173:~
[w205@ip-172-31-4-173 ~]$ tree Exercise_2
Exercise_2
|-- Twittercredentials.py
|-- Twittercredentials.pyc
|-- hello-stream-twitter.py
|-- psycopg-sample.bak
|-- psycopg-sample.py
|-- scripts
|   |-- finalresults.py
|   |-- histogram.py
|   `-- setup.sh
|-- test.txt
`-- tweetwordcount
    |-- README.md
    |-- _build
    |   |-- classes
    |   |   `-- META-INF
    |   |       `-- maven
    |   |           `-- EX2Tweetwordcount
    |   |               `-- EX2Tweetwordcount
    |   |                   `-- pom.properties
    |   `-- stale
    |       `-- leiningen.core.classpath.extract-native-dependencies
    |-- _resources
    |   `-- resources
    |       |-- bolts
    |       |   |-- __init__.py
    |       |   |-- parse.py
    |       |   `-- wordcount.py
    |       `-- spouts
    |           |-- __init__.py
    |           |-- tweets.py
    |           `-- words.py
    |-- config.json
    |-- fabfile.py
    |-- logs
    |   |-- streamparse_tweetwordcount_count-bolt_3_19004.log
    |   |-- streamparse_tweetwordcount_count-bolt_4_19002.log
    |   |-- streamparse_tweetwordcount_parse-tweet-bolt_5_19003.log
    |   |-- streamparse_tweetwordcount_parse-tweet-bolt_6_19005.log
    |   `-- streamparse_tweetwordcount_tweet-spout_7_19015.log
    |-- project.clj
    |-- src
    |   |-- bolts
    |   |   |-- __init__.py
    |   |   |-- parse.py
    |   |   `-- wordcount.py
    |   `-- spouts
    |       |-- __init__.py
    |       |-- tweets.py
    |       `-- words.py
    |-- tasks.py
    |-- topologies
    |   `-- tweetwordcount.clj
    `-- virtualenvs
        `-- wordcount.txt

19 directories, 35 files
[w205@ip-172-31-4-173 ~]$
```

File Dependencies

File	Dependencies
Tweetwordcount.clj	Tweets.py, Parse.py, Wordcount.py
Tweets.py	Tweetwordcount.clj, tweepy, Twittercredentials.py
Parse.py	Tweets.py
Wordcount.py	Parse.py, postgres tcount db, postgres tweetwordcount table

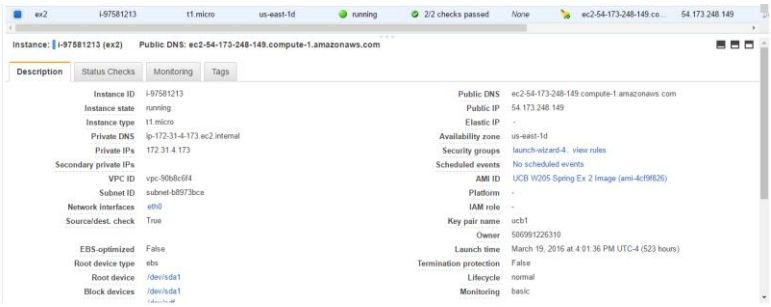
Architecture



1. Tweepy API – opens connection and pulls stream of tweets.
2. Tweets.py – Spout that receives tweets and passes them on to next bolt.
3. Parse.py – Bolt that receives tweets and parses out individual words. Also removes some punctuation.
4. Wordcount.py – Bolt that accumulates words with a count of each. Also removes some common words. Inserts/updates word and count data into Tcounts postgres database.

Information Needed to Run Application (Readme.txt)

1. Start an EC2 instance with the UCB W205 Spring Ex2 Image AMI



2. Restore Exercise_2/ files from Git repository chrisb1249/datastorage
3. Ensure postgres is installed and started

4. Create database Tcount and table tweetwordcount (word, count)
5. Install Tweepy
6. Change users to W205
7. Change directories to /Exercise_2/tweetwordcount
8. $sparse run (enter this into command prompt)
9. Let the program run for a while, then hit <CTL-C> to quit
10. Change directories to ../scripts
11. $python finalresults.py (enter this to see final results)
12. $python histogram.py x,y (enter this to see words with counts between x and y)