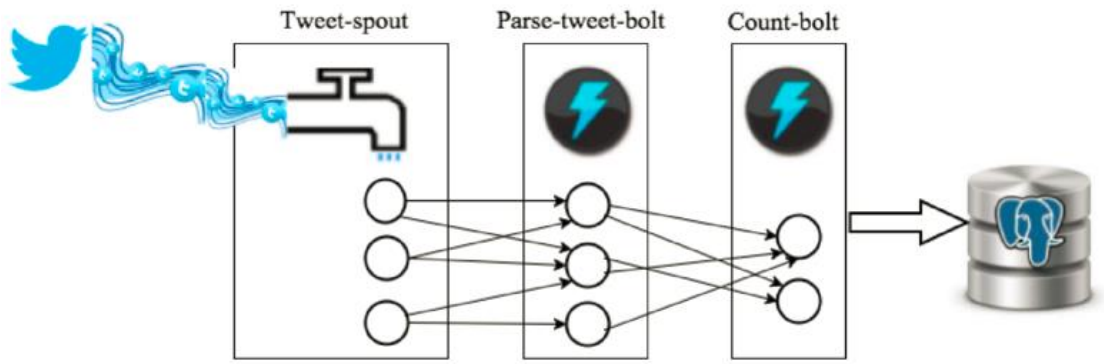
Shih Yu Chang
W205 Storing and Retrieving Data - Exercise 2
July 16th, 2016
AMI ID: i-0105eff1e0869d183
AMI Name: EX2-1 of UCB AMI for W205

Overview:

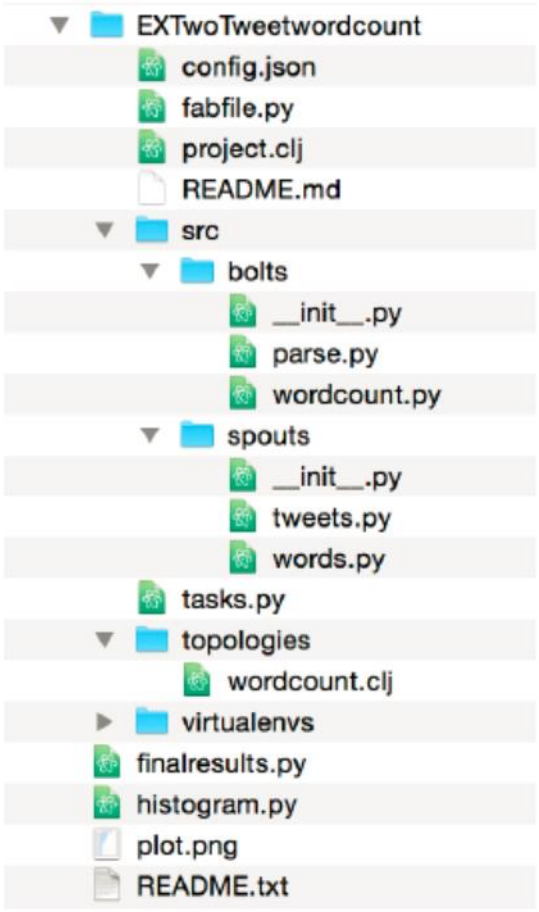
This exercise captures a tweet stream in real time, parses the tweets and stores the resulting word counts in Postgres DB. Technologies involved in this exercise include Aparche Storm, Python, AWS, Twitter API, Postgres, tweetpy, streamparse and psychpg2.

Architecture:

The application architecture is shown below. The pipeline structure can be composed by following components: first, a live stream of tweets is captured by the Tweet—spout component; this stream is then parsed and distilled into individual valid words by parsing at tweet—bolt component; next, unique word counts are tallied by the bolt component with counting function; finally, word counts are stored in the Postgres database.



Files Directory:



Files finalresults.py and histogram.py are used for dynamic querying of the Postgres DB populated by this exercise. plot.png, which presents the most frequently captured words when running aforementioned pipeline. README.txt, which includes application execution instructions. This exercise also uses the default structure of steamparse quickstart project with brief summary shown at following table.

File/Folder	Contents
config.json	Configuration information for all of your topologies.
fabfile.py	Optional custom fabric tasks.
project.clj	leiningen project file, can be used to add external JVM dependencies.
src/	Python source files (bolts/spouts/etc.) for topologies.
tasks.py	Optional custom invoke tasks.
topologies/	Contains topology definitions written using the Clojure DSL for Storm.
virtualenvs/	Contains pip requirements files in order to install dependencies on remote Storm servers.

Dependencies:

Python: Python 2.7 must be installed for program execution. Python packages required include: tweepy (Python package for integrating with Twitter API), streamparse (package for creating and running Apache Storm applications) and psycopg2 (package for interfacing with Postgres).

Postgres: Postgres must be running on the AMI prior program execution. This includes a DB and associated table for tweet word counts being stored.

Tweeter API: After having a Twitter account, the credentials and keys in this exercise must be specified in tweets.py file. This API enables us to extract tweets contents.

Execution Procedures

1. Clone directory from Github
2. Create a Postgres DB called “tcount” with table “tweetwordcount”
3. Navigate to “/EX2Tweetwordcount” and run command “sparse run”
4. Keeping running during tweets extraction, and stop by Ctrl-C.
5. Access Postgres DB to view captured tweet words and their counts.