Project: Counting Words in the Twitter Stream
Date: April 2016
Author: Cory Kind, https://github.com/coryamanda

Application Overview:
Twitter has changed the way individuals communicate in the modern age. Through Twitter, individuals share not just interesting comments or celebrity photos, but news that is important to all of us, users and non-users alike. As a result, understanding the conversations happening on Twitter is a very important goal for many businesses, individuals, governments, and journalists. The goal of this project is to collect tweets from the public Twitter stream, parse them into individual words, store the counts in a Postgres database, and make the results available to the user interactively through Python query scripts. This data can then be used to help users capture interesting trends on Twitter in close to real-time.

Example in Action:
To test this application, I let it run for a bit on the evening of April 13, 2016. The most common words, unsurprisingly, were function words (the, a, and, of, and so on). The most common content word at the end of that data collection was KOBE. Unbeknownst to me, I was collecting data during basketball great Kobe Bryant's record-breaking final game for the LA Lakers. My hope is that this application can be used to help inform people about global conversations that they may otherwise have missed…like the many I evidently miss about basketball. :)

Directory and file structure:
Within the EXTweetwordcount directory, a successful application will contain the following files and directories:

```
(py27environment)[root@ip-172-31-63-101 EXTweetwordcount]# ls -alF
total 72
drwxr-xr-x  8 root root  4096 Apr 15 04:28 ./
dr-xr-x--- 18 root root  4096 Apr 15 05:02 ../
drwxr-xr-x  4 root root  4096 Apr 10 02:12 _build/
-rw-r--r--  1 root root   428 Apr  9 21:11 config.json
-rw-r--r--  1 root root   456 Apr  9 21:11 fabfile.py
-rw-r--r--  1 root root    33 Apr  9 21:04 .gitignore
-rw-r--r--  1 root root  1750 Apr 15 04:28 hello-stream-twitter.py
drwxr-xr-x  2 root root 12288 Apr 15 04:24 logs/
-rw-r--r--  1 root root   529 Apr  9 21:11 project.clj
-rw-r--r--  1 root root  1337 Apr  9 21:11 psycopg-sample.py
-rw-r--r--  1 root root     0 Apr  9 21:11 README.md
drwxr-xr-x  3 root root  4096 Apr 15 04:24 _resources/
drwxr-xr-x  4 root root  4096 Apr  9 21:04 src/
-rw-r--r--  1 root root   456 Apr  9 21:11 tasks.py
drwxr-xr-x  2 root root  4096 Apr 15 05:02 topologies/
-rw-r--r--  1 root root   657 Apr  9 21:11 Twittercredentials.py
drwxr-xr-x  2 root root  4096 Apr  9 21:04 virtualenvs/
```

Note that running the bash-setup.sh file will move all of the relevant files from the GitHub clone into the newly created EXTweetwordcount directory for you.

The relevant subdirectories can be seen below:

```
├── project.clj
├── psycopg-sample.py
├── README.md
├── _resources
│   └── resources
│       ├── bolts
│       │   ├── __init__.py
│       │   ├── parse.py
│       │   └── wordcount.py
│       └── spouts
│           ├── __init__.py
│           ├── tweets.py
│           └── words.py
├── src
│   ├── bolts
│   │   ├── __init__.py
│   │   ├── parse.py
│   │   └── wordcount.py
│   └── spouts
│       ├── __init__.py
│       ├── tweets.py
│       └── words.py
├── tasks.py
├── topologies
│   └── tweetwordcount.clj
├── Twittercredentials.py
└── virtualenvs
    └── wordcount.txt
```

The most important files in the application are the bolts (parse.py, wordcount.py), the spout (tweets.py) and the clojure topology that brings them together (tweetwordcount.clj). Fundamentally, the spout collects the tweets. The first bolt parses them into words, and the second bolt counts and stores them. Credit for the figure below to the UC-Berkeley School of Information.

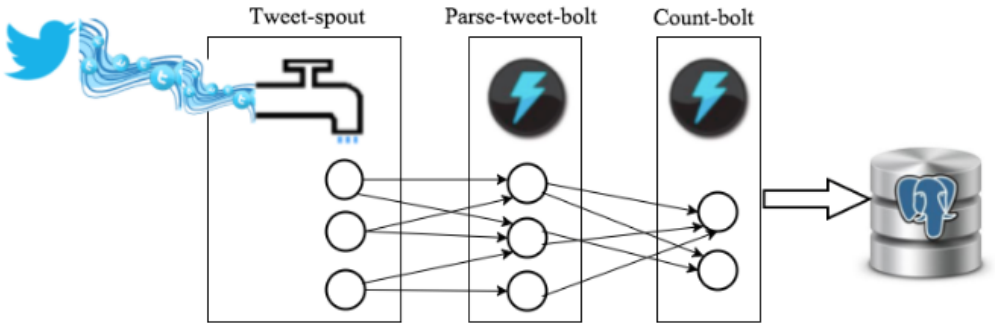

Figure 1: Application Topology

Note that in order for the application to collect tweets, you need to add valid Twitter credentials to Twittercredentials.py. If you don't have Twitter application credentials, you can find them at https://apps.twitter.com/ once you have a valid Twitter account.

Relevant software and dependencies:

- Python 2.7.3
- Postgres 8.4.20
- Streamparse 2.1.4 (in AMI)

The bash script also installs psycopg2 and tweepy if not already installed.

More Detail:

More information is available in the ReadMe files, of which there are three:

- w205-spring-16-labs-exercises/exercise_2/Readme.txt
- w205-spring-16-labs-exercises/exercise_2/screenshots/ReadMe.txt
- w205-spring-16-labs-exercises/exercise_2/serving/ReadMe.txt