W205 - Exercise 2

Rob Mulla, Spring 2017

Introduction and Architecture Description

This document serves as an outline of the architecture and design for exercise 2. In this project an application was designed that captures live tweet information and stores the results into Postgres database. The application contains two bolts**,** the first parses the text of each tweet into its word components. The second bolt incrementally counts the number of occurrences for each unique word. The result is a database containing a row for each unique word with two columns `word` and `count`. The word column contains the word itself while the count column contains the number of occurrences of that word.

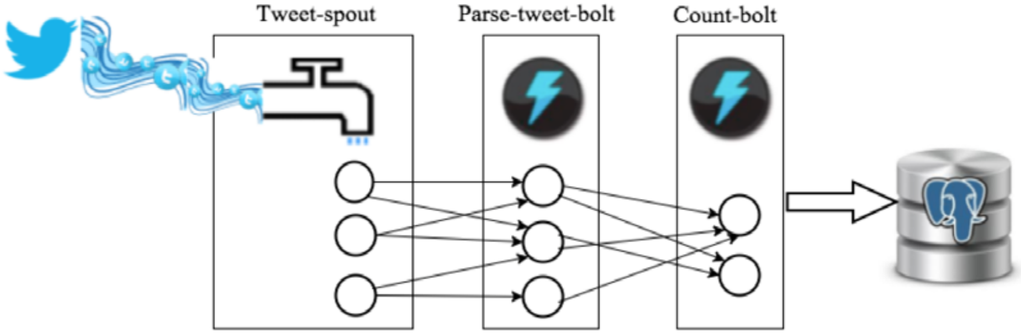

Figure 1: Application Topology

Directory and File Structure

This application is stored in the master github repository which also contains my exercise 1 submittal.

https://github.com/RobMulla/w205

All relevant files for this project can be found under the ~/exercise_2/ directory.

The following relevant files:

- **~/exercise_2** - Master file directory for this exercise
- **~/exercise_2/extweetwordcount/** - Directory containing the application architecture, bolt, spout, and topology files.
- **~/exercise_2/extweetwordcount/src/bolts/ –** Bolt configuration files
  - **parse.py** – Python bolt script for parsing the tweet into words
  - **wordcount.py** – Python bolt script for incrementally counting word occurrences. Saving results to postgres database.
- **~/exercise_2/extweetwordcount/src/spouts/** – Spout configuration files

  - **tweets.py –** Spout stores twitter credentials and pulls the live stream of tweets.
- **~ /exercise_2/extweetwordcount/topologies/** - Contains clj file with application topology
  - **tweetwordcount.clj –** Defines topology (number and connections) between spout and bolts.
- **~ /exercise_2/screenshots/**
  - **screenshot-finalresults.png** – Screenshot of the final results script running (lists all words alphabetically with their count)
  - **screenshot-histogram.png –** Screenshot of the histogram python script running (lists all words with counts between two numbers)
  - **screenshot-twitterstream.png –** Screenshot of the twitterstream application running, counting live tweet word occurrences.
- **~ /exercise_2/README.md/README.txt** – Instructions of how to install and run the application on a fresh EC2 instance.
- **~ /exercise_2/finalresults.py –** Python script which, when passed a single word returns the word count from postgres database. If no word is passed all the words are listed alphabetically with their counts.
- **~ /exercise_2/histogram.py** – Passed two numbers and returns a list of words with counts between these two numbers.
- **~ /exercise_2/Plot.png** – A plot showing the counts for the top 20 words resulting from a run of the twitterstream application.
- **~ /exercise_2/plot.py** –. Custom script created to pulling the top 20 word count to produce the Plot.png file.
- **~ /exercise_2/bar.csv** – The output of the plot.py script, which was used to create Plot.png

Details about adding twitter credentials

In order for the code to work correctly you must manually add your twitter API credentials. Instructions can also be found in the README.txt and README.md files. The tweets.py file contains a location for you to add these credentials.

```
$ cd exercise_2\extweetwordcount\src\spouts\
$ vi tweets.py
```

You will add your credentials in the location that looks like this:

```
twitter_credentials = {

   "consumer_key"        :  "",
   "consumer_secret"     :  "",
   "access_token"        :  "",
   "access_token_secret" :  "",
}
```