Architectural Overview

The Twitter-stream application is a simple word counting tool for analyzing the Twitter-verse. Utilizing Apache storm, it ingests tweets, parses them into words, and counts the words used in each tweet. The tweets are collected using the Twitter-API via tweepy into the storm spout, fed into a parse bolt, which simply splits the tweet into words (using a simplistic algorithm of breaking the tweet at spaces), then feeds them into a word-count bolt which write them to a Postgres database via psycopg2 and logs them to screen. The overall architecture is illustrated in Figure 1.

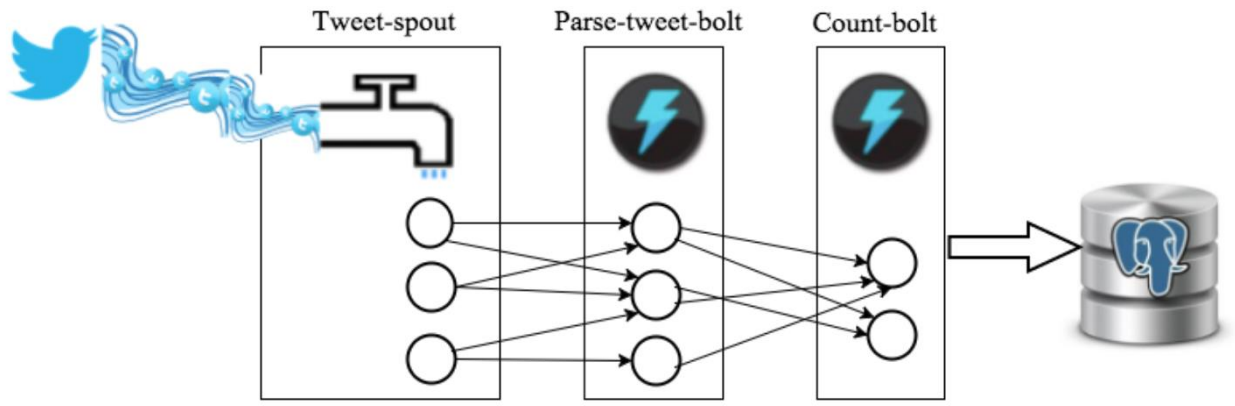

Figure 1: Twitter-stream Application Topology (Source: w205-fall-17-lab-exercises)

File and Folders

To explore the Twitter-stream application, begin at the github repository https://github.com/berkeley-agenright/w205-fall-17-labs-exercises and browse to the exercise_2 folder. All remaining folders are under there. The folder space appears in Figure 2. The green-colored elements represent unmodified files cloned from the original exercise repository. To conserve space, folders in green are not expanded. Gold-colored elements were generated from the sparse quickstart code-generator and untouched after initial generation. Purple elements represent files copied from the tweetwordcount project in the initial repository but unmodified after copy. A summary of file and folders added to the repository is given in the following table:

Filename	Comments
extweetcount/src/bolts/wordcount.py	Modified to write tweets to Postgres using the psycopg2 library
extweetcount/src/spouts/tweets.py	Modified to log into twitter using appropriate credentials
extweetcount/topologies/tweetwordcount.clj	Modified to adjust degree of parallelism
extweetcount /README.md	A brief description of how to run the architecture
screenshots/screenshot-StormRunning.png	A screenshot of the architecture executing, including log messages
screenshots/screenshot-QueryingPostgres.png	A sample query of the Postgres database
screenshots/screenshot-ShowingHistogram.png	Running the histogram.py program
screenshots/screenshot-Top10Tweets.png	Running the topcounts.py program
screenshots/screenshot-WordCounts.png	Running the finalresults.py program
Architecture.pdf	This document

README.md	A top-level readme for the project
Plot.png	A histogram of the top 20 tweet counts
finalresults.py	Displays word counts for either the entire twitter-stream or selected words
histogram.py	Displays word counts in a specific count range
topcounts.py	Displays the n top word counts

Table 1: Summary of repository additions/changes

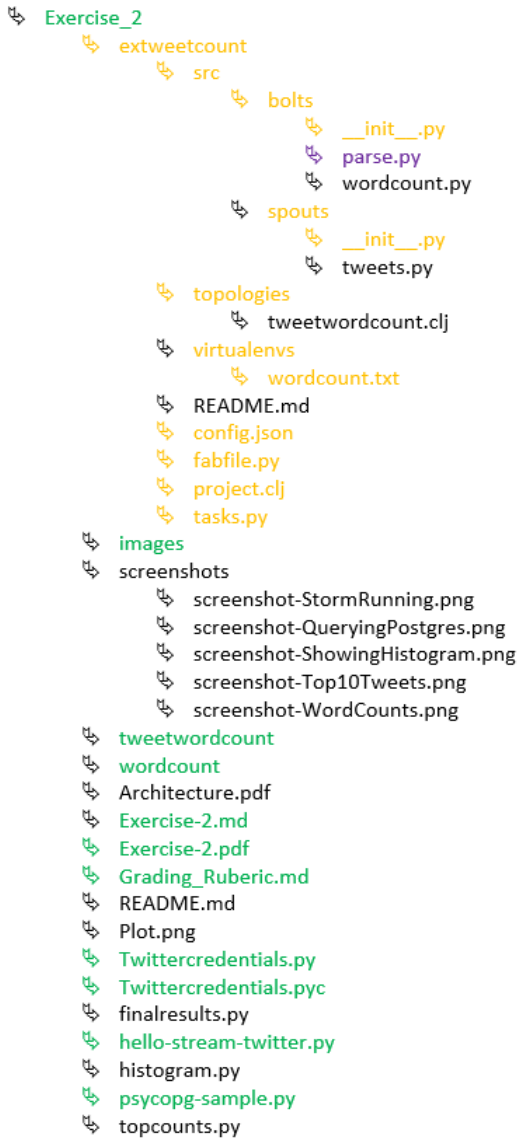

Figure 2:Folder and File Structure (green = unchanged, gold = generated, purple = copied, black = new/changed)

Dependencies

All the files in extweetcount depend on, and are initially generated from, Apache storm (http://storm.apache.org/releases/current/index.html). The tweets.py spout requires tweepy (http://docs.tweepy.org/en/v3.5.0/) as well as an appropriately configured Twitter application account.

The wordcount.py bolt requires the psycopg2 library for Postgres (http://initd.org/psycopg/). All code runs in Python 2.7, against Postgres 8.4.2.

Running the Environment

The base environment for this application is assumed to be the UCB W205 EX2-FULL AMI at aws.amazon.com, configured m1.large (https://aws.amazon.com/ec2/). In addition, we mounted an 100Gb data drive configured for Postgres. After adding tweepy and psycopg2 to the instance, the application can be run as follows:

1. Clone the git repository from https://github.com/berkeley-agenright/w205-fall-17-labs-exercises
2. Change directory to w205-fall-17-labs-exercises/exercise_2/extweetcount
3. Invoke the application using storm (sparse run)

For step 3, you may want to redirect your standard error and output to files for full log analysis.