David Skarbrevik
MIDS W205 - Section 4
Professor Uri Schonfeld
Fall 2016

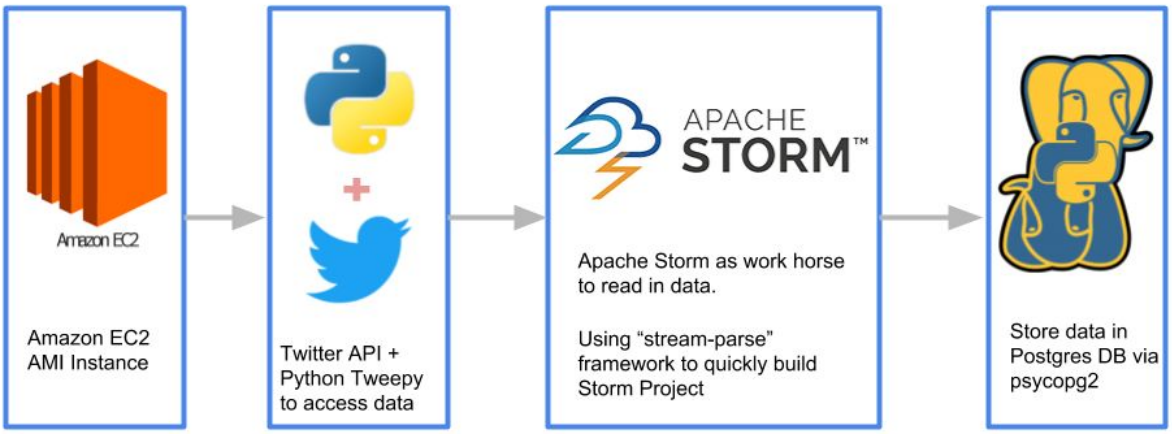
Streaming words from Twitter

Purpose of this Application:

The goal of this project/application is to use Twitter's streaming API to analyze tweets in real time. All tweets are split into individual words and each word is counted in a continuous stream until the application is stopped. While the user of this application can query the resulting database in any way they wish, there are two python scripts included in this application that can be used to see the total count of a given word or the words that have a count total within a given range. Also as a demonstration use-case of this application, there are two bar charts included showing the top words found during a 1 hour long sample stream. The plots were generated using google spreadsheets.

Note to W205 course instructors/graders:
Some base files were changed beyond what was expected for this assignment. Most notably, parse.py and wordcount.py were modified to better account for special situations like words with apostrophes that may have otherwise crashed the application.

Summary of Application Architecture:



Deliverables found in this Project:

Link to Github Repo: https://github.com/dskarby/Exercise-2

File Name	Brief Description
tweetwordcount.clj	Topology file for streamparse.
tweets.py	Spout file for streamparse to intake data from Twitter.
parse.py	Bolt file for streamparse to split sentences.
wordcount.py	Bolt file for streamparse to count words from split sentences.
finalresults.py	script that tells current stream count on any given word.
histogram.py	script that shows all words with total count in a given number range.
Readme.txt	Documentation file to help quickly get application running.
Plot.png	figure showing top 10 words in a sample stream.
More_Interesting_Plot.png	figure with cherry picked popular words from sample stream.
“Storm stream in action.png”	Screenshot of running sample Twitter stream.
“finalresults.py in action.png”	Screenshot of sample finalresults.py results.
“histogram.py in action.png”	Screenshot of sample histogram.py results.
“Storm spout file.png”	Screenshot of tweets.py spout file.
“Storm parse-word.png”	Screenshot of parse.py bolt file.
“Storm word-count.png”	Screenshot of wordcount.py bolt file.

Explanation of file dependencies:

Note that while any system can install/run the needed programs for this application, the following Amazon EC2 AMI was used:
****
UCB MIDS W205 EX2-FULL - ami-d4dd4e3
****
This AMI was used simply because many needed programs are already installed.

The following programs/packages/libraries are needed in order to run this application:

- Python 2.7 or greater
- Apache Storm (allows streaming of data)
- PostgreSQL (database for storing data)
- Psycopg2 (accesses PostgreSQL in Python)
- Tweepy (accesses Twitter API in Python)
- Streamparse (framework for Apache Storm in Python)

How to run the Application:

The Readme.txt file in this application includes directions for running this application, but to reiterate a few key steps:

1. `sparse quickstart <your project name>`
2. cd <your project name>
3. Look in “src” and “topologies” folders and place the appropriate files from this application in their associated folders.
4. Make sure you have a postgres database called “tcount” and an empty table called “tweetwordcount” set up before you run the next step.
5. From main folder of your streamparse project execute:
    a. `sparse run`
6. In a different terminal execute the following to see analysis of your stream:
    a. `python finalresults.py <sample word>`