W205 Exercise 2 Architecture

Matthew Nelson

Application Idea:

This twitter application is designed to intake streaming data from twitter (live tweets), parse and count the individual words from these tweets and store the resulting tally's in a postgres relational database. This database is then queried to serve up high level information regarding the words and their counts through two python scripts (designed to be interacted with at the command line). One script shows the counts for all words, or any specific word if passed as an argument. The other script shows all words and their counts where their counts are between two integers passed as an argument.

Architecture:

- Amazon AWS EC2 Instance for distributed computing. A pre-setup AMI titled "UCB MIDS W205 EX2-FULL" was used. Postgres, streamparse, Hadoop, and python are already installed.
- An Apache Storm application for streaming data is used.
- Streamparse is previously installed on the Amazon AWS EC2 instance utilized for this exercise. Streamparse is used to run python in an Apache Storm.
- Python is the main language used throughout this architecture (excluding the clojure file for the Storm topology).
- A Twitter API, accessed through a personal Twitter account, is used to gain access to current tweets.
- The Tweepy library allows us to access the Twitter API data through Python.
- Postgres is the relational database chosen to store our processed stream data, and is located on a static EBS volume on Amazon AWS which is attached to the EC2 instance.
- The PsychoPG library is used to interact with postgres through Python.

File Dependencies:

tweets.py requires valid Twitter API login credentials

tweetwordcount_setup requires that the table tweetwordcount has not already been created in the tcount postgres database, otherwise it will fail.

psycopg2 must be installed

tweepy must be installed

To run the Application:

1. Launch a "UCB MIDS W205 EX2-FULL" Amazon AWS instance.

2. Install psycopg2
3. Install tweepy
4. Change to w205 user
5. Start postgres
6. Clone github repository https://github.com/matthewpnelson/MIDS_W205_E2.git
7. Update Twitter Credentials in /home/w205/MIDS_W205_E2/extweetwordcount/src/spouts.tweets.py
8. Run tweetwordcount_setup.py
9. Navigate to /home/w205/MIDS_W205_E2/extweetwordcount
10. Run streamparse topology (sparse run)
11. Either:
    a. Let run and open a new window to run query programs
        i. finalresults.py [optional word argument]
        ii. histogram.py [integer1, integer2]
    b. ctrl-c to cancel streaming data collection and run query programs
        i. finalresults.py [optional word argument]
        ii. histogram.py [integer1, integer2]
12. Safely shut down postgres and EC2 instance

Directory & File Structure:

Main File Structure

MIDS_W205_E2
- extweetwordcount
  - src
    - bolts
      - **parse.py** — Tweet parsing bolt to split incoming tweets into individual words. Filter out the hash tags, RT, @ and urls as well as all leading and lagging punctuations (\"?><,'.:;)!(&%). Converts all words to lowercase to ensure proper comparability.
      - **wordcount.py** — Wordcount bolt to count individual words from parse bolt.
    - spouts
      - **tweets.py** — Spout (using tweepy) to collect individual tweets and send to parse bolt. Valid Twitter App credentials must be inserted into this file.
  - topologies
    - **tweetwordcount.clj** — Storm topology linking tweets spout, parse bolt and wordcount bolt and utilizing streamparse

- **finalresults.py** — Serving Python script to query the tweetwordcount table in postgres. Provides the count of a single word if that word is provided as an argument at the command line, otherwise it will print out all words in alphebetical order with each words corresponding count at the time of query.
- **histogram.py** — Serving Python script to query the tweetwordcount table in postgres. Provides the word and word count of all words with a count within the bounds of two integers passed as arguments at the command line in the form [int1,int2]. Sorted by highest to lowest count.
- **tweetwordcount_setup.py** — This script must be run prior to running the extweetwordcount topology within Streamparse/Storm. It creates a postgres database called tcount and a table within tcount named tweetwordcount.

Postgres Database & Tables

Postgres Database Name: tcount

Table Names: tweetwordcount