1. **About This Application:**

This application was designed to capture tweets from Twitter in real time, set up a real time processing pipeline, and then analyzing that data. Our particular analysis focuses on counting the frequency of words in those tweets. The application to parse tweets was built using Streamparse, which builds off of Apache Storm, with a PostgresSql database located on an Amazon EC2 instance to store information. The goal of this project was to demonstrate feasibility and build a working application. Future abstractions of this project can be found in Section 6.

One interesting insight which comes from this analysis which can be seen in the plot.png file is that high occurrences of the terms "nazis" and "white". An extended analysis would need to be conducted to identify the typical occurrences of these words, but I would guess the word "nazis" would not typically occur in the top 20. These two words follow a high tension period in Charlottesville, VA where white nationalist rally has sparked violence of the course of the dates of August 11-13th, 2017 when this majority of this analysis was conducted. This exemplifies how twitter is used by people to voice their opinions on current events.

2. **Application Topology:**

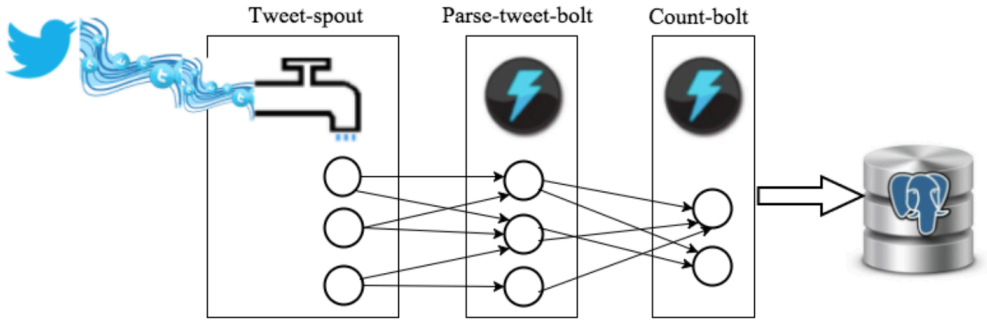

Figure 1: Application Topology

* https://github.com/UC-Berkeley-I-School/w205-summer-17-labs-exercises/raw/master/exercise_2/images/1_topology.png

3. **Application Structure:**

```
|____.DS_Store
|____.gitkeep
|____create_database.py
|____extweetwordcount
| |____.gitignore
| |____config.json
| |____fabfile.py
| |____project.clj
| |____README.md
| |____src
| | |____bolts
| | | |_____init__.py
| | | |____parse.py
| | | |____wordcount.py
| | |____spouts
| | | |_____init__.py
| | | |____tweets.py
| |____tasks.py
| |____topologies
| | |____tweetwordcount.clj
| |____virtualenvs
| | |____wordcount.txt
|____finalresults.py
|____histogram.py
|____plot.png
|____README.txt
|____screenshots
| |____.DS_Store
```

```
| |____screenshot-finalresults.png
| |____screenshot-histogram.png
| |____screenshot-twitterStream.png
|____Twittercredentials.py
```

a) **File Highlights:**

- **create_database.py**
  - Creates requisite database architecture for storing tweets.
- **Twittercredentials.py**
  - Builds Twitter Api using credential keys variables stored in the OS environment
- **finalresults.py**
  - Returns the number of occurrences of a word if specified, otherwise returns all words returned alphabetically.
- **histogram.py**
  - Returns all words that have occurred between the minimum and maximum number of occurrences.
- **Screenshots/**
  - Display the application in action for examples of proper usage
- **Extweetwordcount/**
  - **Topologies/**
    - **Tweetwordcount.clj**
      - Provides an overview to our Steam Parse for how the application will run. This includes the number of parallel instances our bolts and spouts will use.
  - **src/**
    - **spouts/**
      - **tweets.py**
        - Takes in tweets via the Tweetpy api and then filters down the tweets to only English tweets. It then passes these to the parse bolt for processing.
    - **bolts/**
      - **parse.py**
        - Takes in data from tweets.py and processes the valid words.
      - **wordcount.py**
        - Process the words passed from the parse bolt and updates the count for these words in the database.

b)Database Design:

- **Databases:**
  - **Tcount**
    - **Tables:**
      - **Tweetwordcount**
        - **Columns:**
          - **Word**
            - **Primary Key**
            - **Nullable:** False
            - **Type:** Text
          - **Count:**
            - **Nullable:** False
            - **Type:** Int

4. Requirements:

All installation instructions are for Linux. Please review other documentation if you are not using Linux to run this code.

- **Python**
  - **Purpose:** Used to run the finalresults.py and histogram.py functions which interact without database
  - **Installation**:
    - Most Linux distributions include python by default. As a result specific installation instructions will not be provide.
  - **Additional Package Requirements:**
    - **NLTK**
      - **Purpose**: This is python package is used to screen what are known as stop words, which are frequently filtered out in the processing of natural language data. Examples include "The","I", "A", "You", etc.
      - **Installation**:
        - ***Via Command Line*:** sudo pip install -U nltk
        - ***Then Via Python***:
          - import nltk
          - nltk.download()
            - This will launch then nltk download window
              - Hit d for download
              - Install stopwords
    - **Tweetpy**
      - **Purpose**: This package is used to access streaming twitter data.
      - **Installation**
        - ***Via Command Line***: pip install tweetpy
      - **Additional Requirements:**
        - To utilize the Tweetpy api you need to set up a twitter app, which can be done using the following steps:
          - **Login to Twitter: https://www.twitter.com**
          - **Create New Twitter App: https://apps.twitter.com/**
          - **Click on Keys and Access Token and then Create My Access Token**
          - **Export these keys to your system via the command line. This will enable them to be used by your application.**
            - **Comand Line Instructions:**
            - $export TWITTER_CONSUMER_KEY="YourConsumerKey" $export TWITTER_CONSUMER_SECRET="YourConsumerSecret" $export TWITTER_ACCESS_TOKEN="YourAccessToken" $export TWITTER_ACCESS_SECRET="YourAccessSecret"
    - **psycopg2**
      - **Purpose**: This package is used to interact with our Postgress database via Python.
      - **Installation:**
        - ***Via command line***: pip install psycopg2==2.6.2

- **Postgress**

    - **Purpose:** Database technology used to store our results
    - **Installation**:
        - ***Via command line***: sudo apt-get install postgresql postgresql-contrib
- **Streamparse**
    - **Purpose:** Used to capture and process streaming tweet data.
    - **Installation**: pip install streamparse
        - Additional information can be found here: http://streamparse.readthedocs.io/en/stable/quickstart.html#dependencies

5. Running the Application:

- For instructions on running the file please set the README.txt file, which can be found on this git hub repo.

6. Future Developments:

- **Remove Non-Word Characters**:
    - Despite filtering stop words, we had certain symbols like "&" slip through into our data. On the next iteration of this project we should work to remove these none. Perhaps we could use a dictionary to only capture viable words
    - Additionally, we should look at misspellings and try to accurately characterize those words so we get a proper count.
- **Add a Time Element**:
    - To add a higher dimension to our analysis we should add a time element to our process, which would allow the user to identify occurrences through time and track trends in Tweet topics.