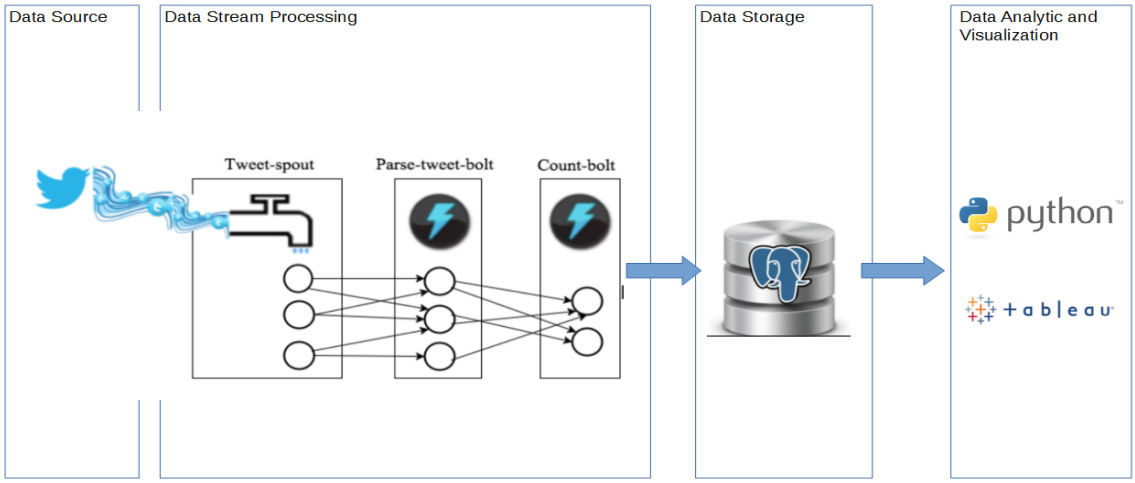
Twitter Application Architecture

I. Application Architecture



This application is designed to stream tweets from Twitter API, using ApacheStrom streaming application called streamparse to pull in the tweet, parse it by word, and process it by doing word count. The application then create a database tcount in Postgres and store/update the word count in real time in small batches. This database serve as a service layer which could be connected to analytics and BI tools like python and tableau to retrieve results as needed.

II. Directory and File Structure

```
W205_2017_spring/
        exercise_2/
                extweetwordcount/
                        src/
                                spouts/
                                        _init_.py
                                        tweets.py
                                bolts/
                                        _init_.py
                                        parse.py
                                        wordcount.py
                        topologies/
                                tweetwordcount.clj
                        vurtualenvs/
                                tweetwordcount.txt
                                wordcount.txt
                        .gitignore
                        README.md
                        config.json
                        fabfile.py
                        project.clj
                        tasks.py
                        finalresults.py
                        histogram.py
                screenshots/
                        exercise 2 twitter streaming.JPG
```

exercise 2 twitter update postgres.JPG
exercise 2 finalresult output.JPG
readme.txt
hello-stream-twitter.py
psycopg-sample.py
Plot.PNG
ODBC setup.PNG
Architecture.pdf
readme.txt

III. File Dependencies

The tweets.py file in spouts requires Twitter app credential that is created by the user. After create an app from Twitter, user should create a TwitterCredentials.py file within spouts folder with their credential for the tweets.py to use.

IV. Application Idea

The intent of this application is to create a process to process the tweets in real time and serve it in terms of word count to the user. To achieve this, the application is designed to have 4 major parts: Twitter API, data processing layer, data service layer, and analytic layer. For data processing layer, we choose Apache Storm which is designed to streaming data. There are three layers in its topology: 1 tweet spout with 3 process to take in data from twitter. It feeds the tweets into the first bolt layer—parse tweet bolt with 3 process and have it parsed into single words and remove hashtags, hyperlinks, user mentions, etc. Then the parsed word are feed into the second bolt—count bolt with 2 process to count the occurrence of each word. The stream is designed to shard by word so that those 2 process will not count the same word separately and overwrite the result.

For the database, we choose to store the word count in Postgres as table. In order to have it show the up to date count, database and table are created when the second bolt initiated and word count are continuously feed into the table while the count bolt is counting. It will search the table and update the count if the word exist and add new record if the word is not currently in the table.

For analytic layer, user could use psycopg2 in python to analyze the result and use ODBC to connect the table to Tableau to create visualizations.