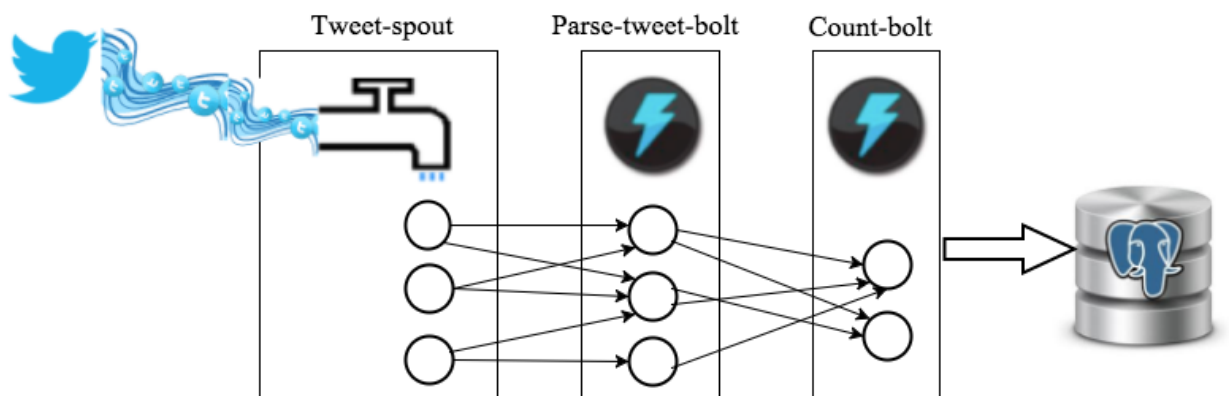


## Application Idea

Capturing and analyzing live twitter data around your business interest area can give you a deeper understanding of the current social trends and demands. Historical data can give you information on the mainstream trends over a certain period of time, but live data can provide immediate and real-time insight. For example, a person who manages the ads for the TV programs can capture the live Twitter trends at the time of the live show to engage more TV viewers during a popular TV program (i.e contextual advertising).

The main idea of this application is to collect real-time tweets through the Twitter API, parse the tweets into individual words, count the occurrences of each unique word, and storing the words and their counts in a Postgres database.

Figure 1 shows the overall architecture of the application. Figure 1 also shows the storm topology that you need to develop as part of the application. Using Tweepy library, the application reads the live stream of tweets from twitter in the **Tweet-spout** component. The **Parse-tweet-bolt** parses the tweets, extracts the words from each parsed tweet and emits the words to the next bolt component (i.e **Count-bolt**) in the topology. **Count-bolt** counts the number of each word in the received tuples and updates the counts associated with each words in the **Tweetwordcount** table inside the **Tcount** database. **Tcount** is a Postgres database.



**Figure 1: Application Topology**

## File Structure

Name of the program	Location	Description
<b>tweets.py</b>	Exercise_2/tweetwordcount/src/spouts/	Tweet-spout
<b>parse.py</b>	Exercise_2/tweetwordcount/src/bolts/	Parse-tweet-bolt
<b>wordcount.py</b>	Exercise_2/tweetwordcount/src/bolts/	Count-bolt
<b>weetwordcount.clj</b>	Exercise_2/tweetwordcount/topologies/	Topology for the program
<b>Twittercredentials.py</b>	Exercise_2/	Twitter App keys
<b>hello-stream-twitter.py</b>	Exercise_2/	Sample Twitter stream program
<b>psycopg-sample.py</b>	Exercise_2/	Sample psycopg code
<b>psycopg-init.py</b>	Exercise_2/	Code to initialize Postgres database and table; <b>must</b> run before running program
<b>finalresults.py</b>	Exercise_2/Serving-scripts/	Serving script 1 of 2
<b>histogram.py</b>	Exercise_2/Serving-scripts/	Serving script 2 of 2
<b>Screenshot-twitterStream.png</b>	Exercise_2/Screenshots/	Screenshot of hello-stream-twitter.py running
<b>screenshot-extractResults.png</b>	Exercise_2/Screenshots/	Screenshot of program running
<b>plot.png</b>	Exercise_2/	Bar chart showing the top 20 words in my Twitter stream

## Pre-requisites

**Environment:** Python 2.7

**Python libraries:**

- psycopg2
- tweepy
- virtualenv

**Other:**

- lein

## Step-by-step:

1. **After connecting to the pre-baked EC2 AMI, activate Python 2.7 environment.**

```
$source /opt/py27environment/bin/activate
```

2. **Mount EBS volume and start Postgres database. Go on Postgres and create a database called tcount. Exit Postgres.**

```
$mount -t ext4 /dev/xvdf /data
$/data/start_postgres.sh
$psql -U postgres
#CREATE database tcount;
#\q
```

3. **Clone the project GitHub repository, and then start a project called “tweetwordcount” in the streamparse directory.**

```
$git clone https://github.com/Jackanator/W205-Storing-and-Retrieving-Data.git
$cd streamparse
$sparse quickstart tweetwordcount
```

4. **Remove the existing topology files, spouts, and bolts.**

```
$rm /root/streamparse/tweetwordcount/topologies/wordcount.clj
$rm /root/streamparse/tweetwordcount/src/spouts/words.py
$rm /root/streamparse/tweetwordcount/src/bolts/wordcount.py
```

5. **Copy the new topology files, spouts, and bolts into the corresponding directories.**

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/tweetwordcount/topologies/tweetwordcount.clj
/root/streamparse/tweetwordcount/topologies
```

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/tweetwordcount/src/spouts/tweets.py
/root/streamparse/tweetwordcount/src/spouts
```

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/tweetwordcount/src/bolts/parse.py
/root/streamparse/tweetwordcount/src/bolts
```

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/tweetwordcount/src/bolts/wordcount.py  
/root/streamparse/tweetwordcount/src/bolts
```

**6. Copy the .py files.**

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/Twittercredentials.py  
/root/streamparse/tweetwordcount
```

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/hello-stream-twitter.py  
/root/streamparse/tweetwordcount
```

```
$cp -avr /root/W205-Storing-and-Retrieving-Data/Exercise_2/psycpg-init.py  
/root/streamparse/tweetwordcount
```

**7. Run program in tweetwordcount project directory. The “psycpg-init.py” script deletes existing “tweetwordcount” table and creates a new one.**

```
$cd tweetwordcount  
$python psycpg-init.py  
$python hello-stream-twitter.py  
$sparse run
```