ARCHITECTURE

Directory

Main folder: **Exercise_2/**

```
tweetwordcount/
        src/
                bolts/
                        __init__.py
                        parse.py
                        wordcount.py
                spouts/
                        __init__.py
                        tweets.py
        topologies/
                tweetwordcount.clj
        virtualenvs/
                tweetwordcount.txt
        README.md
        config.json
        crdbta.py
        fabfile.py
        project.clj
        tasks.py
screenshots/
        Screenshot-final-results-hello.png
        Screenshot-final-results.png
        Screenshot-histogram.png
        Screenshot-wordcount.png
        Screenshots-stream-twitter-test.png
serve_scripts/
        barchart.py
        finalresults.py
        histogram.py
Plot.png
Twittercredentials.py
exercise_2_code.sh
hello-stream-twitter.py
psycopg-sample.py
```

File Structure

Exercise_2 folder
Contains three main folders:

1) tweetwordcount
2) screenshots
3) serve_scripts

Contains four important scripts and a bar chart plot:

1) Plot.png
2) Twittercredentials.py
3) exercise_2_code.sh
4) hello-stream-twitter.py
5) psycopg-sample.py

Description of Key Files

tweetwordcount/
Folder containing all the files needed to run the twitter application program.

tweetwordcount/crdbta.py
Creates the "tcount" database and a table within the database called "tweetswordcount."

tweetwordcount/src/
Folder containing programs for the spouts and bolts.

tweetwordcount/src/spouts
tweets.py: Creates the listener for the twitter stream and reads in the tweets. Listens out for English tweets and then emits the tweets.

tweetwordcount/src/bolts
parse.py: Collects the tweets emitted by the spout and splits the tweets into words. Then filters out the hash tags, RT, @ and urls, strips leading and lagging punctuations and checks if the words only have ASCII characters. Once complete, emits all the valid words.
wordcount.py: Collects the words emitted by parse-tweet bolts and counts the number of times each word appears. Inserts new words into the database that are not already in the database. Updates the word count for words that are already in the database. Logs the word count.

tweetwordcount/topologies/
tweetwordcount.clj: Implements the topology shown in the graph below. Three spouts which listen for and reads incoming tweets, and emits the tweets to three bolts which parse the tweets by splitting them into words and cleaning them up, then sends them to two bolts which counts the number of occurrence of the words in the twitter stream.

serve_scripts/
finalresults.py: Script that lists the number of occurrences of a specified word or all of the words in the twitter stream.

histogram.py: Lists the words that have counts which fall within a specified range.

barchart.py: Shows the top 20 words in my twitter stream.

screenshots/
Folder containing screenshots displaying the results of various aspects of the application.

screenshots/
Screenshot-final-results-hello.png: Results from running “finalresults.py hello” that counts the number of occurrences of “hello.”

Screenshot-final-results.png: Results from running the “finalresults.py” script that counts the number of occurrences of all the words in the stream.

Screenshot-histogram.png: Results from running the “histogram.py” script that returns all the words that occur a certain number of times in the stream.

Screenshot-wordcount.png: Shows the results of running the tweetwordcount project in the form of a log containing the words and their respective counts.

*Screenshots-stream-twitter-test.png: Shows the results of running the “*hello-stream-twitter.py” script. In this case it returned all tweets that contained “elclasico.”

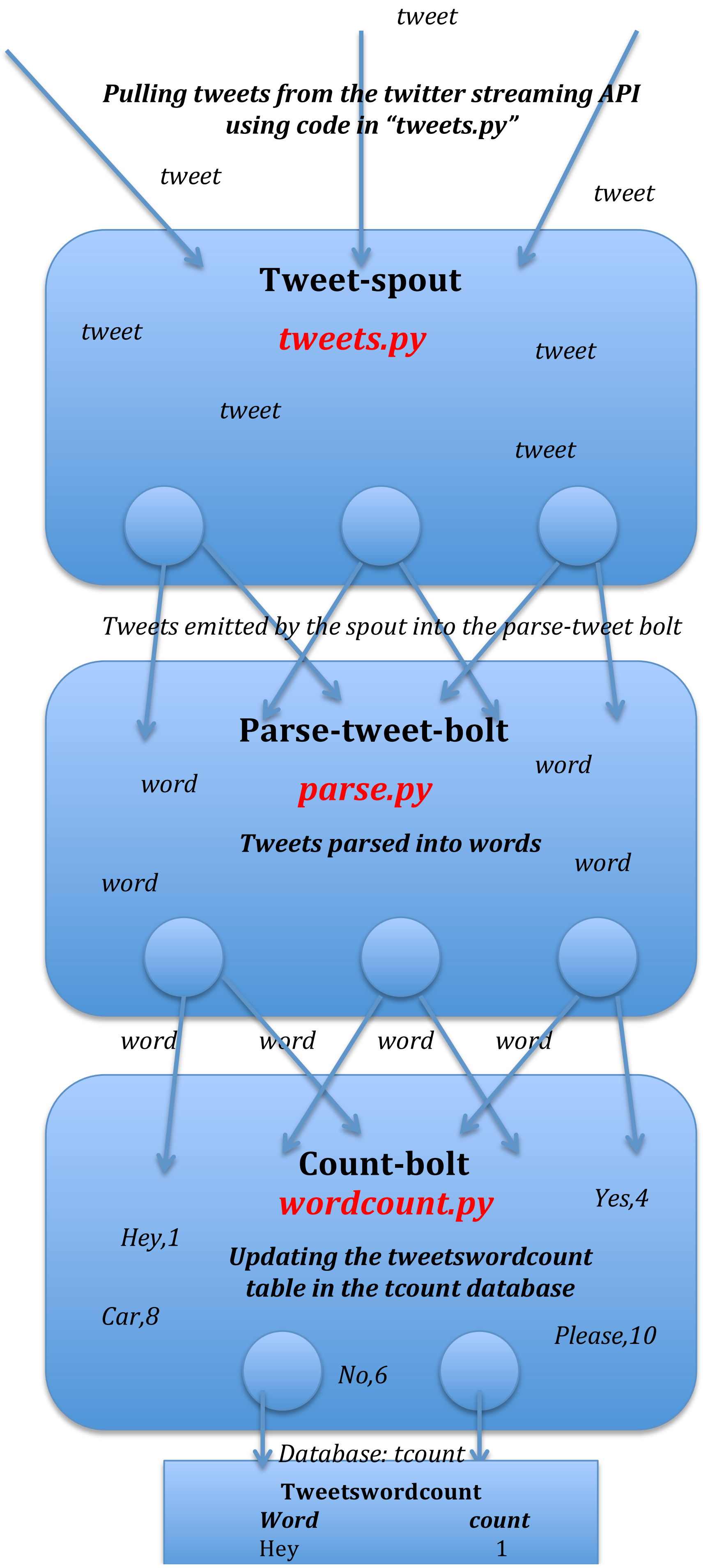
Plot.png: Bar chart showing the top 20 words in my twitter stream.

hello-stream-twitter.py: Using Twitter stream API to print all the tweets in the stream containing the term "elclasico" in a 1 min period.

Twittercredentials.py: Contains the consumer key, consumer secret, access token and access token secret needed to read the stream of tweets from Twitter.

exercise_2_code.sh: Contains all the code needed to run the program in terminal.

Please see diagram of topology below:



Pulling tweets from the twitter streaming API using code in “tweets.py”

Tweet-spout

tweets.py

Tweets emitted by the spout into the parse-tweet bolt

Parse-tweet-bolt

parse.py

Tweets parsed into words

Count-bolt

wordcount.py

Updating the tweetswordcount table in the tcount database

Database: tcount

Tweetswordcount

Word	count
Hey	1