Twitter Data Analysis

11.20.2016

Amr Rashid
UCB - W205
EXERCISE 2

Overview

This is a streaming application that analyzes the Twitter data. Streamparse is used with a given topology. The application reads the stream of tweets from the Twitter streaming API, parses them, counts the number of each word in the stream of tweets, and writes the final results back to a Postgres database.

Goals

1. Capture live Twitter Data setup a stream processing pipeline
2. Get insights from Twitter live data by means of simple queries

Specifications

Using **Tweepy** library, the application pulls the live stream of tweets out of Twitter API in the **Tweet-spout** component.

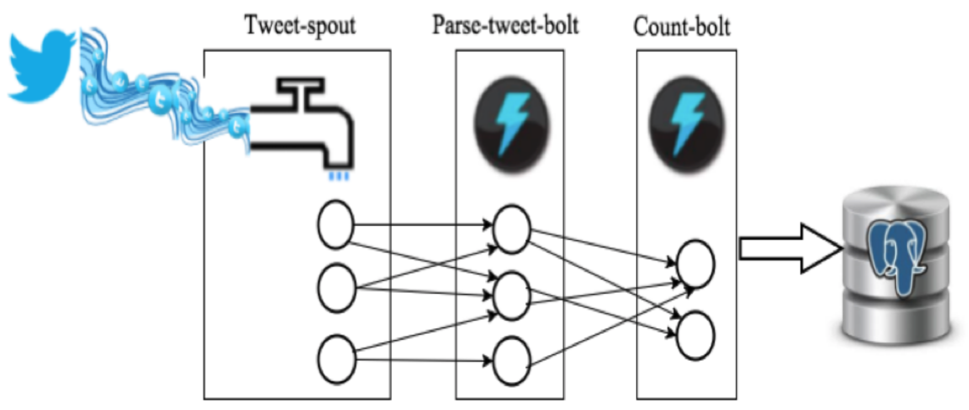
The **Parse-tweet-bolt** parses the tweets, extracts the words from each parsed tweet and emits the words to the next bolt component (i.e Count- bolt) in the topology.

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** postgres database.

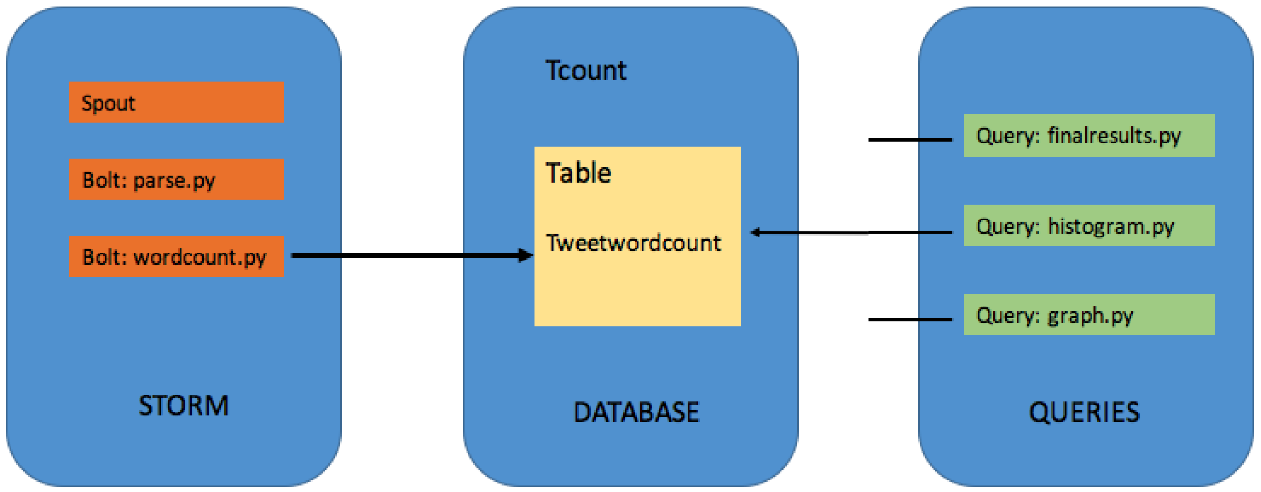
Count-bolt connects to the Tcount postgres database through **psycopg2**

Architecture

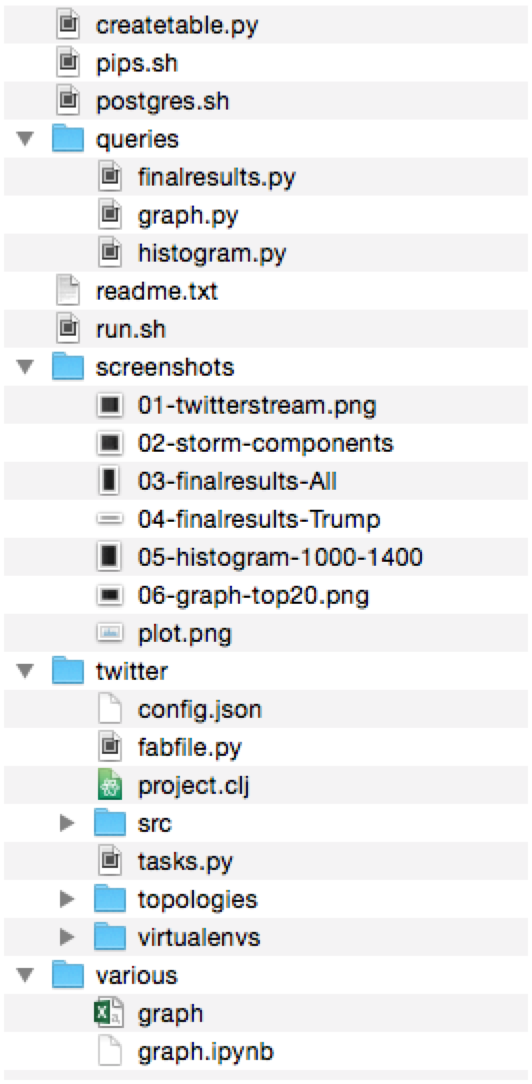
Storm Topology



Dependencies



File Structure



```
createtable.py
pips.sh
postgres.sh
▼ queries
    finalresults.py
    graph.py
    histogram.py
readme.txt
run.sh
▼ screenshots
    01-twitterstream.png
    02-storm-components
    03-finalresults-All
    04-finalresults-Trump
    05-histogram-1000-1400
    06-graph-top20.png
    plot.png
▼ twitter
    config.json
    fabfile.py
    project.clj
  ▶ src
    tasks.py
  ▶ topologies
  ▶ virtualenvs
▼ various
    graph
    graph.ipynb
```

Files on root folder:

Pips.sh:	Installs tweepy & psycopg2
Postgres.sh:	Runs Postgres
Run.sh:	Runs the Application

createtable.sh: Used only to drop Tcount Database and create a new one

Directories:

twitter:	Contains all Storm Files (topology, spouts, bolts etc.)
queries:	Contains Python files to query the database
screenshots:	Contains all screenshots requested in this Exercise
various:	Contain python files to draw the graph with matplotlib (local)

Serving Scripts:

finalresults.py: gets a word as an argument and returns the total number of word occurrences in the stream

finalresults.py: without an argument returns all the words in the stream and their total count of occurrences, sorted alphabetically in an ascending order

Histogram.py: gets two integers k1,k2 and returns all the words that their total number of occurrences in the stream is more or equal than k1 and less or equal than k2

graph.py: returns top 20 words in the stream in terms of number of occurrence. ***To make it meaningful, it excludes words from a list of most common words e.g: I, me, you, we, on, in, that, the, this etc..***

Notes:

- Screenshots for end-to-end execution of the application are all under screenshots folder
- step-by-step instructions to run the application are in readme.txt file under root directory