W205: Exercise 2 – Real-Time Tweet Word Count

Application Idea:

Tweetwordcount is an application that taps into Twitter's API with a streamparse (Apache Storm) backend that dynamically parses each tweet in real-time, counts the number of occurrences of unique words and saves these results into a Postgre SQL Database.

Technologies used in this exercise:

- Apache Storm
- Apache EC2
- Python
- Twitter API
- Streamparse
- Postgres SQL
- Psycopg
- Tableau

Database Info:

Database	tcount
Table	tweetwordcount

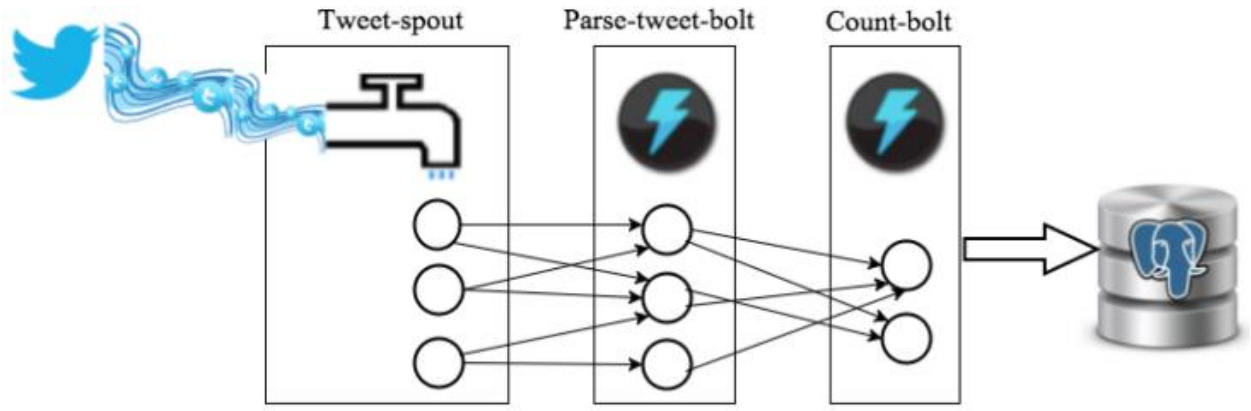
Database Schema:

Table	tweetwordcount
Column Name	**Type**
word	String(80)
count	int

Application Description:

The application is setup in a way where the user can SSH (Secure Shell) into an Amazon EC2 Instance that is running Storm and PostgreSQL on the server. Storm directly consumes real-time Tweets using the Spouts and Bolts topology mentioned above and saves the data in a PostgreSQL database. Finally, Tableau remotely accesses the saved data in the relational database and builds visualizations on top of this stored data.

Application Topology:



Runtime instructions and notes:

Login to Amazon EC2 Instance with an UCB MIDS W205 EX2-FULL AMI.

Ensure Python, psycopg2, and tweepy are installed and that you have Twitter application Credentials.

Ensure PostgreSQL is installed and fired up!

Clone this repository: git clone https://neman018@github.com/neman018/W205_Exercise2.git

Navigate to the Tweetwordcount directory: cd ex2/Tweetwordcount

Run the Application: streamparse run

Directory Structure and Component Details:

**inside /exercise_2/Tweetwordcount/ directory*

File Name	File Type	Description
src	Directory	Folder storing the Source Files
bolts	Directory	Folder storing the Bolts
__init__	python	Initialization file
parse	python	This is the code for the first Bolt that is responsible for parsing each individual word from the tweet message. It consumes the output data from the Spout. There are three of these parse-tweet bolts that send a word string object downstream to the count bolts using the shuffle methodology.
wordcount	python	This is the code for the second bolt that processes and maintains the frequency of each word and saves the data into a Postgre database. This second bolt consumes data from the first bolt and outputs a final tuple ["word","count"] that will be saved directly into the Postgre SQL Relational Database. There are two of these count bolts.
spouts	Directory	Folder storing the spouts
__init__	python	Initialization file
tweets	python	This is the spout that connects directly to the twitter API and reads all English tweets that contain the words: [a, the, I, you, u]. Finally, this spout publishes a single tweet string object to the downstream bolts.
topologies	Directory	Folder storing the topology of the architecture
tweetwordcount	clojure	This clojure file definses the application topology.
virtualenvs	Directory	Folder storing the virtual environment vars
tweetwordcount	txt	States required dependencies.
wordcount	txt	States required dependencies.
.gitignore	txt	Legacy File
README	md	Legacy File
config	json	JSON Config File
fabfile	python	Custom Actions prior to topology
finalresults	python	This is a script I created that spits out the number of occurances of a specific word. If no arguments are passed in, then it will return all the words in the stream and the total occurances, sorted alphabetically.
histogram	python	This script returns all words with occurances between the first and second user-inputted argument integers.
project	clojure	JVM Dependencies [storm-core 0.9.5 / 0.0.4-Snapshot]
tasks	python	Pre/Post Submit Functions