Exercise 2 Architecture

Peter Zhou

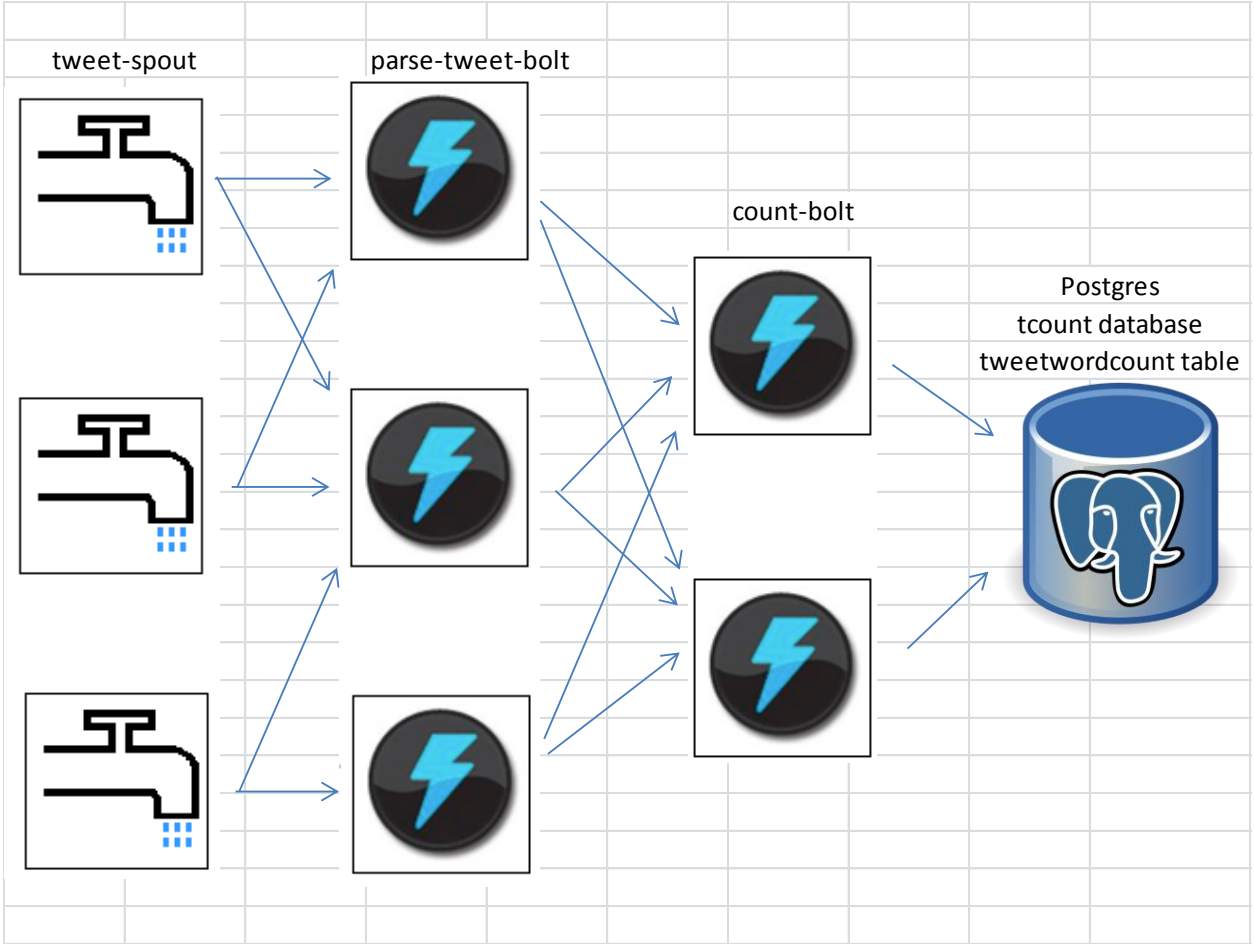
W205

Application Idea:

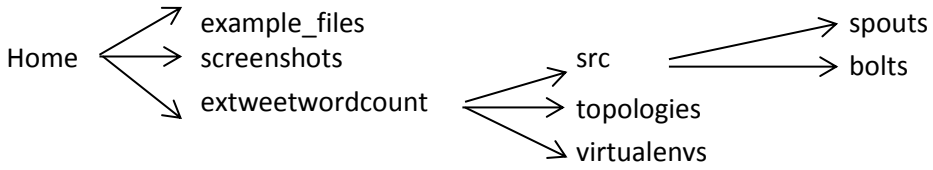
This application is meant to capture live tweets by using the Tweeps library, process them in real time using a Storm topology, and aggregate the results into a Postgres database. We begin by creating a database and table within Postgres by executing the file create_psql_table.py. Then using the Storm topology, we read a live stream of tweets from Twitter with the tweet-spout that then gets parsed with the parse-tweet-bolt which extracts the words of each parsed tweet which then passes this on to the count-bolt which then counts the number of each word and updates the counts into the Postgres database containing the tcount database and tweetwordcount table. After collecting the tweets and updating the table, we can then use the python files finalresults.py and histogram.py to evaluate the count of words based on the users input.

Description of Architecture:

For this application, we use a live Twitter feed that gets read by 3 tweet-spouts which then gets fed into 3 parse-tweet-bolts which then passes it on to 2 count-bolts that update a Postgres table.



Directory and File Structure:



Home Directory:	
Contents	File Type
README.txt	txt
plot.png	png
Architecture.pdf	pdf
example_files	**folder**
Twittercredentials.py	python
Twittercredentials.pyc	pyc
hello-stream-twitter.py	python
psycopg-sample.py	python
screenshots	**folder**
screenshot-finalresult.png	png
screenshot-histogram.png	png
screenshot-stormcomponents.png	png
screenshot-twitterstream.png	png
extweetwordcount	**folder**
src	**folder**
spouts	**folder**
tweets.py	python
bolts	**folder**
parse.py	python
wordcount.py	python
topologies	**folder**
tweetwordcount.clj	python
virtualenvs	**folder**
wordcount.txt	txt
config.json	json
create_psql_table.py	python
fabfile.py	python
finalresults.py	python
histogram.py	python
project.clj	clj
tasks.py	python

File Dependencies:

create_psql_table.py:

1) psycopg2
2) from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

tweets.py:

1) tweetwordcount.clj (from the topologies folder)
2) itertools, time
3) tweepy, copy
4) queue, threading
5) streamparse.spout import spout

parse.py:

1) tweetwordcount.clj (from the topologies folder)
2) re
3) streamparse.bolt import Bolt

wordecount.py

1) tweetwordcount.clj (from the topologies folder)
2) collections import Counter
3) streamparse.bolt import Bolt
4) psycopg2
5) from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

finalresults.py

1) sys
2) psycopg2
3) from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

histogram.py

1) sys
2) psycopg2
3) from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT