Architecture of

Twitter Word Counts Application

Guangzhi (Frank) Xie

---

1. Overview

The application uses Apache Storm to ingest live tweets from Twitter Stream API, and stores word count for each word in a Postgres database for further analysis. It tracks people's live interests regarding two candidates for US 2016 election, Donald Trump and Hillary Clinton. The high level architecture is shown in *Figure 1*.

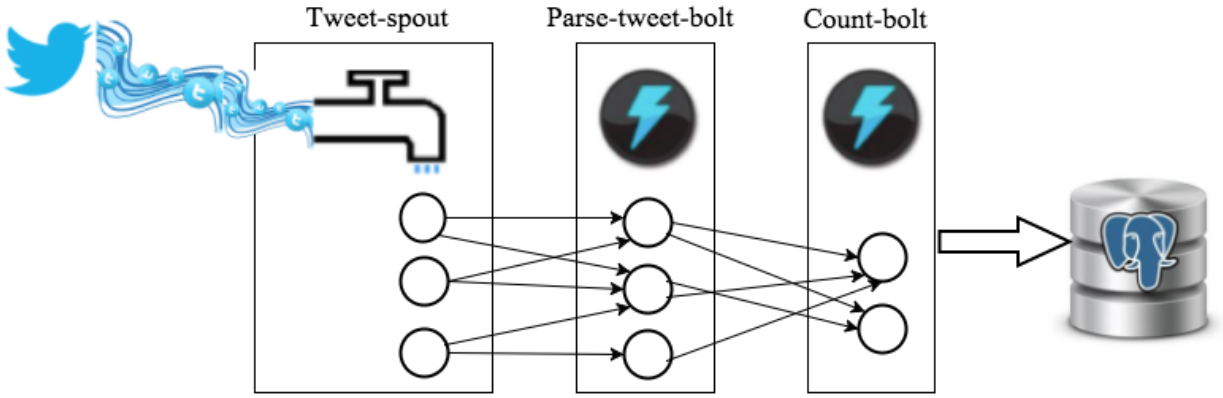

Figure 1 Application Topology

2. Architecture

The application contains two main components:

1. Apache Storm app to ingest, parse, count, and store word counts. The spouts ingest live tweets from Twitter Stream API. It tracks on key words "donanld", "trump", "hillary", "clinton", and "election" using Tweepy Stream's track functionality, and passes tweets down to parse bolts to filter out noises and break tweets into separate words. The word count bolts will then count each word by treating all words in their lower cases, and save the results into Postgres database.

2. Python scripts to interact with the stored data and serve the information. They are provided as examples for further data analysis.

File Structure

File/Folder Name	Location	Description
tweetwordcount		Folder for the Storm app
tweetwordcount.clj	tweetwordcount/topologies/	Topology for the app
tweets.py	tweetwordcount/src/spouts/	Spout to ingest live tweets
parse.py	tweetwordcount/src/bolts/	Bolt to parse tweets
wordcount.py	tweetwordcount/src/bolts/	Bolt to count words
serves		Folder for the python scripts
finalresults.py	serves	Script to show counts for words
histogram.py	serves	Script to show words and their counts in a range in ascendant order
dbsetup.py		Script to setup database and table in Postgres
docs		Documentation for the app
Architecture.pdf	docs	High level description
Plot.png	docs	Bar chart shows top 20 words
screenshots		Fold for the screenshots

File/Folder Name	Location	Description
screenshot-twitterStream.png	screenshots	Screenshots of the running application for the stream of tweet counts
screenshot-storm-components.png	screenshots	Screenshots of the storm topology
screenshot-extract-results.png	screenshots	Screenshots of the extracted results from database

3. Dependencies

- Amazon Web Services EC2 instance
- Twitter application with access key and token
- Python 2.7
- Postgres Database
- virtualenv
- lein
- streamparse
- psycopg2
- tweepy
- redis