W205 Exercise 2 Qing Zhang

## Introduction:

Streaming applications may seem complex, but understanding how they operate is critical for a data scientist. In this exercise, we will explore a streaming application that analyzes Twitter data. To explore a complex implementation in a short period, you will develop your application using an existing codebase. I will use Stream parse, as seen in Lab 6, with a given topology. The application reads the stream of tweets from the Twitter streaming API, parses them, counts the occurrences of each word in the stream of tweets, and writes the results back to a Postgres database.

## Application Architecture

The application architecture uses what we learnt so far to efficiently analysis tweets. The follow graph is the application process.

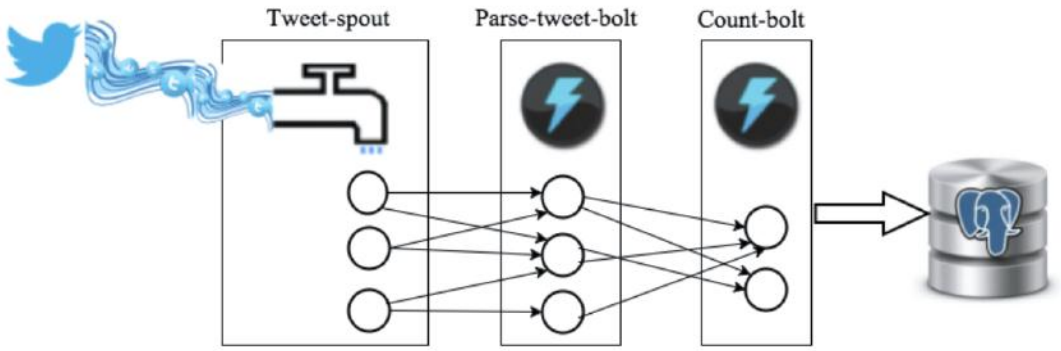


Figure 1: Application Topology

Following components are used as part of the architecture:

• Apache Strom: to help us store and unzip the stream data in real-time processing.

• Amazon EC2: help us to provides scalable computing capacity through Amazon EC2 Instance.

• Python: mainly use to run through the application.

• Twitter API: Tweepy is used for interfacing with Twitter to receive the feeds.

• Streamparse: help to execute python code in real-time.

• Postgres SQL: We uses Postgres for storing the words and count them.

• Psycopg: another language used in this application.

## Folder Structure:

Tweetwordcount

- Src
  - Bolts:
    - Parse.py
    - Wordcount.py
  - Spouts:
    - Tweets.py
  - Topologies:
    - tweetwordcount.clj
- Screenshots
  - Final result png
  - Histogram png
  - Successful Parse bolt png
  - Top 20 png
- Architecture.pdf
- Readme.txt

## File Dependencies

The application requires following software installed prior to execution.

- Apache Stream Strom

- Postgres

- Version: latest
- Install command for Psycopg; Install command- pip install psycopg2
- A table named tcount database is created, and a table tweetwordcount is used to record the counts. The table has two columns: word (text) and count (int), and a primary key is defined on word to avoid duplicate word entries.

Python 2.7

Tweepy API

- Install command: $pip install tweepy
- Twetter APP- Visit https://apps.twitter.com and click

- on "Create New App".

Package installation

To successfully run this application,

We have to import some extra packages.

Import psycopy2.

Import psycopy2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

from collections import Counter

from streamparse.bolt import Bolt

Execution process:

Introduction

Set up the environment. As the instruction, this whole exercise is running on Amazon EC2 instance.

The instance should be created under AMI Name: UCB MIDS W205 EX2-FULL

AMI ID: ami-d4dd4ec3.

Step1 start postgres.

- login to postgre database using command 'psql -U postgres'
- use command create database tcount;
- use command \c tcount, if it did not connect the database.

Step 2 data collection

- Update your credential from your own tweetapp. The location is /tweetwordcount/src/spouts/tweets.py
- At tweetwordcount directory, type sparse run and wait for one or two mins for data collection.
- Stop the program by typing ctrl c.

Step 3 Database interaction

To get the result, you must do the following step.

Step 1 Change direction to service code.

Step 2 run command “python finalresults.py (your own word)”for example, python finalresults.py happy.

Step 3 run command “python his.py number 1, number 2” for example, python his.py 10 100.