Assignment 1

1. Outline

This is the first assignment of this course. Basically, the assignment is designed to facilitate you putting together most of the materials you have learned from the lectures and labs.

All the data prepared for the assignment is the following folder in ROGER:

/gpfs_scratch/geog479/assignment1

Twitter data: 2014_03_01_tweets.txt

New York taxi data: ny_taxi_march.csv

Task 1: Using Pig script to filter the New York Taxi data

Requirement:

1. As you can see from the Lab 6, there are invalid records in the provided New York taxi dataset. Specifically, the pick-up and drop-off coordinates are not always valid.

Let's confine all the records within this bounding box as approximation of New York City, which is -74.256090, 40.496111 (lower left); -73.700273, 40.917585 (upper right). Note it is longitude, latitude.

2. A case scenario is that we want to find all the taxi records with passenger_count **greater than** 1 **between** the dates of 2013-03-10 and 2013-03-20.

3. How many data entries are left in your output?

4. Description of the fields in the taxi data

medallion, hack_license, vendor_id, rate_code, store_and_fwd_flag, pickup_datetime, dropoff_datetime, passenger_count, trip_time_in_secs, trip_distance, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, payment_type, fare_amount,surcharge, mta_tax, tip_amount, tolls_amount, total_amount

Please give short description of your experiment producers or better your thinking process.

Answer:

Task 2: We have practiced the word count program with Hadoop Streaming API. In this case, we will perform similar tasks on the Twitter messages filtered by Pig script. The Twitter data covers the United States of March 1st, 2014.

Requirements:

1. Please remove the duplicated records first

2. Let’s focus on the Chicago area with the following coordinates: -88.707599, 41.201577 (lower left);-87.524535, 42.495775 (upper right);

3. To enable future spatial analysis, we will only deal with tweets with geo-locations and timestamps.

4. Keep the message contains the following word: “happy”, “sad”, “Chicago”, “traffic”, “weather”, “wind”

2. Once you have the output, please summarize the frequency of unique words (i.e., word counts) from the filtered data and sort the output based on the number of occurrences in descending order and pick up the top 10 words.

The sorting command in Linux can be issued as:

Say, your result is stored as “result.txt” shown as

X,10

Y,11

Z,16

```
sort -t',' -k2 -r result.txt
```

where -t',’ says the separator in the line is “,” -k means which column to sort, -r means reverse order.

If it does not work, copy the result to your desktop and try to use excel or other program to sort the list.

Please give short description of your experiment producers or better your thinking process.

Answer:

Task 3 (**Optional, TBD**):