Assignment 6
Map-Reduce

Computing Lab (II)
10th Feb 2021

This assignment is on map-reduce, which is a distributed and scalable way of extracting/mining required information from multiple datasets stored on multiple servers. Follow the tutorial to understand how you can design mapper and reducer for specific queries/operations.

Tutorial on map-reduce

You can start with a simple word count problem. Say, we have a text file and we want to count the frequency of occurrence of each word. The tutorial below explains how to solve this problem using a map-reduce algorithm.
Tutorial References :
http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
Next you can also look into the following tutorial for slightly harder query (tf-idf scores)
https://www.tutorialspoint.com/map_reduce/map_reduce_tutorial.pdf

Tasks

1. Study HDFS and MapReduce basics.
2. Write the necessary mapper and reducer routines in python to implement the queries mentioned in later sections.
3. Ensure that the reducer routine prints results of queries to a text file (as asked in the later sections).

Dataset

We will be using a dataset on a Facebook (FB) network with **3,892** users. Please download the data “network.txt”. You have to extract required information from this data. Below is a sample of data available in the file.

(**network.txt**)

```
0,1838
0,1744
0,14
0,2543
1,1009
1,1171
1,1465
```

where

- Each line represents an **undirected edge** between two users indicated by two node-ids separated by a comma. So, no pair of nodes are repeated in the file.

Sample code to read the files

```
import gzip
fp=open(“network.txt”)
for line in fp:
    l_arr=line.split(“,”)
    node1=l_arr[0]
    node2=l_arr[1]
```

Queries

The queries are to be implemented in the mapper and reducer phases. Some of them may give empty results. You need to implement the following queries in this assignment. The queries have an imaginary backstory to help you find a real world perspective in this assignment.

1. You are the CEO of a car company. You want to run an advertisement campaign on FB for your newly launched car. An FB official tells you that if you choose to send the ads to all users each with at least 20 friends, then they will charge you INR 100 per user. You want to calculate how much you will have to spend for your ad campaign if you choose to float ads only for users with at least 20 friends. Write mapper and reducer routines for this. (Hint: Count all the users (node-ids) with at least 20 friends using mapper and reducer, and then multiply 100 with that before printing the output)

   [20]

2. The FB official also gives you a free welcome offer valid only for once. This offer is for you to judge the impact of advertising on FB. They say they will send your ad to any 10 users as per your selection. You are a very intelligent business person. So, you think that if you send your ads to users who have a large number of friends, then your ad will easily spread through FB shares, and you won’t have to pay anything for the paid ad campaign. So, you want to find the top-10 users (node-ids) with the highest number of friends (order doesn’t matter). Write mapper and reducer routines for this.

   [20]

3. After you exploit the free offer, immediately FB realizes the flaw in the offer, and discontinues it. However, your ad has already gone to the top-10 users. Now, FB being a very manipulative company, decides to hide your ad from the newsfeed of the users who are connected to the top-10 users, so that even if the top-10 users share your ad, it won’t be visible to others and it won’t spread. So, FB wants to find out the users who have friendship with at least one out of the top-10 users. Write mapper and reducer routines for this. (Hint: the mapper can use the list obtained in the last query).

   [20]

4. Even though FB makes all the effort to stop your ad from spreading, by the time FB is able to change the newsfeeds of the users, your ad has already spread to a part of the FB network. As your trick has worked, you are now very happy, and you spend most of your advertising budget in organizing parties. However, you suddenly realize that, in the US, there is no demand for your car. You find out that even if your ad has spread to a part of the FB network, it has not gone to any user in the US which is the biggest market for your company. You, then, run back to the FB official. Looking at your situation and your previous endeavour, FB wants to teach you a lesson. The FB official tells you that they will

now charge you **INR 10*X** to show your ad to **a user with X number of friends**. After spending most of the advertising budget, you decide to send your ad only to top-10 US users with the highest number of friends in the US. In this regard, your company's board of directors want to know how much will be required for this task. You know that all the users with node-id between 0 and 999 are the only users from the US. Write mapper and reducer routines to find out how much you will have to spend to send your ad to the top-10 US users with the highest number of friends in the US.

[30]

How to run and test your code:

Since we do not have access to a Hadoop cluster, we will be testing our codes on a Linux system as follows:

```
cat network.txt | python mapper.py | sort | python reducer.py  > result.txt
```

Or just

```
python mapper.py | sort | python reducer.py > result.txt
```

Explanation on the above commands:

1. "cat" is a linux command to print the contents of a file on the console.
2. The pipe operator (|) directs the output of the previous command to the next command.
3. "sort" is a linux command to sort the input lexicographically.
4. ">" can be used to save the standard output in a file.

Deliverables: 4 sets of python codes (mapper and reducer), 4 Makefiles, 4 result.txt, 1 readme

Evaluation Scheme: Results: 90 marks, Coding Style: 10 marks

Important Instructions

1. **The mapper routine can pass over the network.txt file only once. Python dictionaries should not be used in the mapper routine. However, you can use arrays or dictionaries of size up to 10. These must be limited to one dimension only. Any violation will attract a penalty upto 100% of the marks assigned.**
2. **Submission Rule:** Python code for above functionalities must be compressed as **.tar.gz** (gzip) and named "**A6_<RollNo>.tar.gz**". For each query, make a directory named "**Query<no.>**". Your files must be inside the respective directories. Strictly adhere to this naming convention. Submissions not following the above guidelines will attract penalties.
3. **Makefile:** Each query folder **must** have it's own Makefile to execute the routines. You can find a relevant tutorial here (https://opensource.com/article/18/8/what-how-makefile). **Note: "sort" behaves differently in different environments. Ensure that you use sort in a way which works properly on a linux machine.**
4. **Code error:** If your code doesn't run or gives error while running, you will be awarded with zero mark. Your code must run correctly on **a linux machine**.
5. **Plagiarism Rule:** If your code matches (more than 50%) with another student's code, all those students whose codes match will be awarded with zero marks (may be with -ve marks too depending on the situation) without any evaluation. Therefore, it is your responsibility to ensure that neither you copy anyone's code nor anyone is able to copy yours.