# Homework 1

Release date: Mar 21, 2020
Due date: Apr 11, 2020 23:59
***No late homework will be accepted!***

**Every Student MUST include the following statement, together with his/her signature in the submitted homework.**

*I declare that the assignment submitted is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations.*

Signed (Student_____) Date:_____

Name_____ SID_____

**Submission notice:**

- Submit your homework to Email xuhl@dgut.edu.cn

**General homework policies:**

A student may discuss the problems with others. However, the work a student turns in must be created COMPLETELY by oneself ALONE. A student may not share ANY written work or pictures, nor may one copy answers from any source other than one's own brain.

Each student **MUST LIST** on the homework paper the **name of every person he/she has discussed or worked with**. If the answer includes content from any other source, the student **MUST STATE THE SOURCE**. Failure to do so is cheating and will result in sanctions. Copying answers from someone else is cheating even if one lists their name(s) on the homework.

If there is information you need to solve a problem but the information is not stated in the problem, try to find the data somewhere. If you cannot find it, state what data you need, make a reasonable estimate of its value, and justify any assumptions you make. You will be graded not only on whether your answer is correct, but also on whether you have done an intelligent analysis.

## Q1 [50 marks]: Multi-node Hadoop Cluster Setup

In this problem, you are required to setup a Hadoop cluster using Amazon EC2, or Google Compute Engine, or Windows Azure. Ref [1] - [3] provide the tutorial for each platform.

In order to setup a Hadoop cluster with multiple VMs, you need to setup a single-node Hadoop for each VM. Then change the configurations of Hadoop master and slaves. Ref [4] - [5] are the official documentation of Hadoop installation. You can refer to Tutorial 2 (https://xuhappy.github.io/courses/BigData/tutorial/tutorial2/) to configure a single-node Hadoop.

Grading Scheme:

(a) [10 marks] Install a Single-Node Hadoop (Hadoop version: 2.7.1 or above).

(b) [20 marks] Install and setup a Multi-Node Hadoop cluster **with 4 VMs (1 master and 3 slaves)**. Run a word counting example as in Tutorial 2. Use Shakespeare work (Ref [11]) as the dataset. There are two different settings:
   a) one directory consisting of many small files;
   b) merge these small files into one large file.
   Compare the running time between different settings and explain the reason.

(c) [20 marks] Run the word counting example for the **large** dataset (Ref [12]) multiple times while modifying the number of mappers and reducers for your MapReduce job(s) each time. You need to examine and report the performance of your programme for at least 4 different runs. Each run should use a different combination of number of mappers and reducers. For each run, performance statistics to be reported should include: (i) the time consumed by the entire MapReduce job(s) and the maximum, minimum and average time consumed by (ii) mapper tasks and (iii) reducer tasks. Tabulate the time consumption for each MapReduce job and its tasks. One example is given in the following table. Explain your observations.

| Maximum mapper time | Minimum mapper time | Average mapper time | Maximum reducer time | Minimum reducer time | Average reducer time | Total job |
|---|---|---|---|---|---|---|
| 60s | 40s | 50s | 60s | 40s | 50s | 2.5 min |

Hints:

1. Launching instances with Ubuntu 14.04 LTS is recommended.
2. To support Hadoop 2.0, the minimum requirement may be 2GB memory and 1 core CPU, i.e., *t2.small* for AWS.
3. After installing a single-node Hadoop, you can save the system image and launch VM with that image, so that you can save the effort of installing the single-node Hadoop on each VM. Another suggestion is to use *pssh* or *ansible* for parallel setup (refer to Tutorial 1).
4. After finishing step (a), you are suggested to run word counting example to verify if the installation is successful.

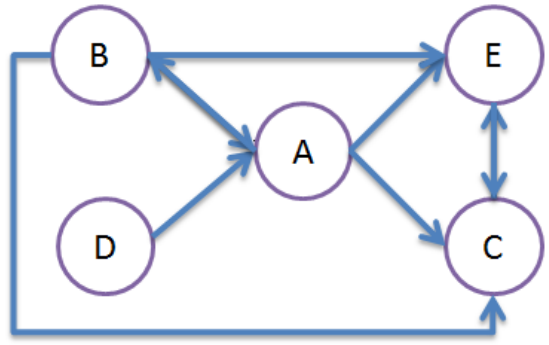
Submission Requirements:

1. Include all the key steps, together with screenshots, into the report.

# Q2 [50 marks + 15 bonus marks]: Community Detection in Online Social Networks

Community detection has drawn lots of attention these years in machine learning field. With the popularity of online social networks, such as Facebook, Twitter and Sina Weibo, we can get many valuable datasets to develop algorithms. In this homework, you will implement a community detection algorithm for the online social networks. Basically, people belong to the same community when they have high similarity (i.e. sharing many common interests). There are two types of relationship in online social networks. One is symmetric, which means when Alice is Bob's friend, Bob will also be Alice's friend; the other is asymmetric, which means Bob may not be Alice's friend even Alice is Bob's friend. In the second case, there are two roles in the relationship: follower and followee (like the case when using Twitter and Weibo). When Alice follows Bob, Alice is the follower and Bob is the followee.

In order to detect communities, we need to calculate the similarity between any pair of users. In this homework, similarity is measured by the number of common followees for the given pair of users. The following figure illustrates the process.




A → B means A follows B

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | / | 2 | 1 | 0 | 1 |
| B | 2 | / | 1 | 1 | 1 |
| C | 1 | 1 | / | 0 | 0 |
| D | 0 | 1 | 0 | / | 0 |
| E | 1 | 1 | 0 | 0 | / |

Pair-wise Similarity

The set of followees of A is {B, C, E} and set of followees of B is {A, C, E}. There are 2 common followees between A and B (i.e. C and E). The similarity between A and B is therefore 2.

We will provide three datasets with different sizes, generated from Sina Weibo users' relationship. The smaller dataset contains 1000 users, and the medium one contains around 2.5 million users, and the large one contains 4.8 million users. Each user is represented by its unique ID number. The download links of three datasets are listed in the Ref [6] - [8]. The small dataset is provided to facilitate your initial debugging and testing. The design of your programme should be scalable enough to handle all the datasets.

The format of the data file is as follows (different followers separated by space):

*followee_1_id:follower_1_id follower_2_id follower_3_id ….*
*followee_2_id:follower_1_id follower_6_id follower_7_id ….*

...

For example, in the above figure, the data is represented as:

*A:B D*
*B:A*
*C:A B E*
*E:A B C*

The output of the community detection is that for *EVERY* user, we want to know the TOP K most similar persons. The output format should be (different similar persons separated by space):

*User_1:Similiar_Person_1 Similiar_Person_2 … Similiar_Person_K*
*User_2:Similiar_Person_1 Similiar_Person_2 … Similiar_Person_K*
*…*

Solve the above community detection problem using MapReduce. You can either use the IE DIC or the Hadoop cluster you built for Question to 1 run your MapReduce programme(s). You are free to use any programming languages to implement the required MapReduce components, mapper(s), reducer(s), partitioner(s), etc (e.g by leveraging the "Hadoop Streaming" capability).

Again, the design of your programme should be scalable enough to handle all three of the datasets. In other words, you are expected to use the same programme to solve the following problems:

a. [50 marks] Get the TOP 10 (=K) most similar people of EVERY user in the medium-sized dataset in Ref[7].

b. [Bonus 15 marks] Get the TOP 10 (=K) most similar people of EVERY user in the large dataset in Ref [8]. (Hints: use composite key and secondary sort as in Ref [9] and [10])
Warning: Please closely monitor the resource consumption, including memory and hard disk. The Amazon free tier may not provide enough hard disk. You need to apply for extra storage or use other platforms.

Hints:

1. Since the medium-sized dataset is also quite large already, to avoid excessive AWS charges, you are advised to you are advised to be careful when dealing with AWS.
2. As for the large dataset, you may want to set mapreduce.map.output.compress=true to compress the intermediate results, in case you don't have enough local hard disk space (to hold the intermediate tuples).
3. The large dataset may take a long time (around 10 hours) to process even if everything is correct. **You should start doing this homework as early as possible**!

Submission Requirements:

1. Submit the MapReduce code
2. As for (a) and (b), submit the community detection results of all those users whose IDs share the same last 5 digits with your DGUT student ID. For example, if your student ID is 115500054321, then you need to submit the community members for users with ID = 54321, 154321, 254321, 354321, …, 1154321,...

## *References:*


[1] Microsoft Azure Tutorial
https://azure.microsoft.com/en-us/get-started/
[2] Google Compute Engine Tutorial:
https://cloud.google.com/compute/docs/quickstart
[3] AWS Tutorial:
https://aws.amazon.com/getting-started
[4] Single-Node Hadoop setup:
https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html
[5] Hadoop cluster setup:
https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html
[6] Small scale dataset
https://www.dropbox.com/s/ntzk80l5iiiuh50/Small%20Dataset.txt?dl=0
[7] Medium scale dataset
https://www.dropbox.com/s/6sxnnadhxbyk7ho/Medium%20Dataset.txt?dl=0
[8] Large scale dataset
https://www.dropbox.com/s/lrlgz50m88j6fpc/Large%20Dataset.txt?dl=0
[9] Composite Key
http://tutorials.techmytalk.com/2014/11/14/mapreduce-composite-key-operation-part2/
[10] Secondary Sort
http://codingjunkie.net/secondary-sort/
[11] Shakespeare works
https://github.com/hupili/agile-ir/raw/master/data/Shakespeare.tar.gz
[12] Large dataset for word counting
https://www.yelp.com/dataset