IEMS5730 Spring 2019 Homework 1

Release date: Jan 21, 2019

Due date: Feb 18, 2019 (Mon) 11:59am (i.e. noon-time)

The solution will be posted soon after the deadline. 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 on Elearning system 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, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/.

Signed (Student_____) Date:_____

Name_____ SID_____

Submission notice:

- Submit your homework via the blackboard system

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 [100 marks]: Multiplying Sparse Matrices with MapReduce

Consider a matrix M with element m_{ij} in row i and column j, and another matrix N with element n_{jk} in row j and column k. The the product P=MN is the matrix P with element p_{ik} in row i and column k where:

$$p_{ik} = \sum_j m_{ij} n_{jk}$$

Of course, the number of columns of M should be equal to the number of rows of N. In the applications of our interest. M and N are both large but sparse matrices, i.e. most of their elements are 0. Matrices are stored with the following format:

$$< i > \text{<TAB>} < j > \text{<TAB>} < m_{ij} >$$

a. [**40 marks**] You need to design one or more MapReduce job(s) to perform the matrix multiplication described above in the small dataset[1].

b. [**40 marks**] Perform the matrix multiplication in the median dataset[2] and large dataset[3].

c. **[20 marks]** Re-run your program in b) multiple times using different number of mappers and reducers for your MapReduce job each time. You need to examine and report the run-time performance statistics of your program for at least 4 different combinations of number of mappers and reducers. For each setting, performance statistics to be reported should include: (i) the time consumed by the entire MapReduce jobs and (ii) the maximum, minimum and average time consumed by the mapper tasks and reducer tasks. Try to explain your observation.
    i. If your program design requires a multiple-stage pipeline of MapReduce jobs (i.e. one different MapReduce job for each stage), you only need to modify the number of mappers and reducers for the MapReduce job in the last-stage of the pipeline and report its performance statistics accordingly.

    Tabulate the run-time statistics for the MapReduce job and its tasks. One example is given in the following table:

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

Important Hints:

1. You can use a single MapReduce job to get the final results of the above matrix multiplication problem. However, it is also okay if you need to use multiple (i.e. a series of different) MapReduce jobs. In either case, your programme should be scalable enough to handle the big dataset we provide without running into memory exhaustion problems.
2. You are allowed to leave the final step of sorting the row and columns to another script (e.g., via `Linux sort` command). Your MapReduce program only needs to calculate the necessary indices.
3. You can either run your MapReduce programme(s) using the multi-node Hadoop cluster you built for Homework#0 or directly use the Amazon Elastic MapReduce (EMR) or its counterpart in Google Compute Engine to run your MapReduce code. *If you do not have a credit card to setup the cluster, please contact the TAs and we will provide you an AWS account.*
4. If you use Java to write your MapReduce programme, you can specify the number of reducers with the following code: `job.setNumReduceTasks(20).` If you use Hadoop streaming with Python, you can specify it via the following command option: `-D mapred.reduce.tasks=20`. Refer to [6] for more information.
5. If you use Java, you can specify the number of mapper with the following code: `job.setNumMapTasks(20).` If you use Hadoop streaming with Python, you can specify it via the following command option: `-D mapred.map.tasks=20`. If this does not work, you may need to modify the split size in the `$hadoop/etc/hadoop/mapred-site.xml: mapred.min.split.size =268435456(256M)`
6. The large dataset may take a long time to process even if everything is correct. You should try the small dataset first and use it to help debug your initial programme. Depending on your programme design, you may need to alter the resource configuration of the nodes in your cluster in order to handle the large dataset.
7. Feel free to use any programming languages (Java [3], Python [4], etc) to implement the required MapReduce components, mapper(s), reducer(s), partitioner(s), etc.
8. Start working on this homework as soon as possible.

Submission Requirements:

1. Submit your MapReduce code
2. As for the small dataset, submit the matrix multiplication results of all those rows whose IDs share the same last 2 digits with your CUHK student ID. For example, if your student ID is 1155038634, you would need to submit the 34th, 134th, 234th, 334th rows …
3. As for the large dataset, submit the matrix multiplication results of all those rows whose IDs share the same last 3 digits with your CUHK student ID. For example, if your student ID is 115500054321, then you need to submit the 321th, 1321th, 2321th, 4321th rows …
4. Submit the performance comparison report

5. Include the result from Step 1 to Step 4 into a SINGLE PDF report.

References:

[1] Small dataset:
http://mobitec.ie.cuhk.edu.hk/iems5730Spring2019/homework/hw1_small_dataset.zip
[2] Large dataset:
http://mobitec.ie.cuhk.edu.hk/iems5730Spring2019/homework/hw1_median_dataset.zip
[3] Large dataset:
http://mobitec.ie.cuhk.edu.hk/iems5730Spring2019/homework/hw1_large_dataset.zip
[4] Write a Hadoop program in Java
https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html
[5] Write a Hadoop program in Python
http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/
[6] How many Mappers and Reducers?
https://wiki.apache.org/hadoop/HowManyMapsAndReduces