CS6675/ CS4675

Homework Assignment 2

(Programming Category)

Student Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Session: cs6675 or CS4675 (circle one)

You are given three types of programming problems in the second homework assignment. Problems 2 and 3 consists of multiple options. You only need to choose the first problem or one option from the second or third problem as your second homework. Feel free to choose any of your favorite programming language Java, C, Perl, Python, …

Post Date: Monday of Week 4 (Feb. 1)

Due Date: Midnight on Friday of Week 5 (Feb.12)

**Problem 1. Learning HDFS and Hadoop MapReduce.**

*Suitable for students who are the beginner of Hadoop MapReduce Platform*

Install HDFS and Hadoop MapReduce on your laptop, and run two example map-reduce programs/applications provided with the package, such as word count, sort, grep, and report the runtime performance for each example program using two different sizes of datasets. Note the package comes with both code and datasets.

You may use excel file to generate your runtime statistics plot or organize the performance measurement data in a tabular format.

You are encouraged to learn by observing the runtime performance of Hadoop MapReduce programs through running two different programs and show your analysis on the runtime performance of the 4 MapReduce jobs (2 programs on 2 different datasets, each job is composed with one program and one dataset).

Deliverable.

1. URL to the MapReduce codes and the datasets used
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.
4. Your analysis.

**Problem 2. Learning SPARK by Examples**

This problem has 3 options.

**Option 1.**

This option is designed for students who are beginner of Spark and have no hand-on experience with Hadoop MapReduce.

1. Download Spark on your laptop and run one example program of your choice on two example datasets provided in the Spark package.

2. Report the runtime performance for your chosen example program using two different sizes of datasets. You may use excel file to generate your runtime statistics plot or organize the performance measurement data in a tabular format.

3. Show your analysis on the runtime performance of the measurement results.

Deliverable.

1. URL to the Spark code and the datasets used
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.
4. Your analysis.

**Option 2.**

This option is designed for those students who are familiar with Hadoop MapReduce programming model and would like to learn Spark.

1. Download Spark on your laptop and run an example MapReduce program on Spark. You can choose the example program and the dataset provided in the Spark package or Hadoop package.

2. Download HDFS and Hadoop MapReduce and run the same program you chose to run on Spark but this time run on top of HDFS.

3. Report the runtime performance of MapReduce program on both Spark and HDFS. You may use excel file to generate your runtime statistics plot.

4. Compare the performance of the same program running on two different platform and provide your analysis.

Deliverable.

1. URL to the Spark code and the datasets used
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.
4. Your analysis.

**Option 3.**

This option is designed for students who are familiar with both Hadoop MapReduce and Spark and interested in hand-on comparison of them through example programming problems or big datasets and / or through configuration tuning.

Compare Hadoop MapReduce and SPARK using a common analytic problem: a simple one like sort or word count, and a complex one like Clustering or k nearest neighbor search. You are encouraged to write your own program.

Deliverable.

1. URL to the HDFS/Spark code, the MapReduce code and the datasets used
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.
4. Your analysis.

**Problem 3. Learning BigTable, NoSQL, Key-Value Systems**

This problem has 2 options.

**Option 1.**

This option is designed for students who are familiar with HDFS but are beginner of Big Table/NoQL/Key-Value Store Systems and have no hand-on experience with any existing Big Table systems.

1. Download one of your favorite NoSQL (key-value) store software package, such as HBase, Redis, LevelDB, Mongo, and so forth.
2. Populate the system with example dataset. For example, you can use YCSB to generate dataset or use real world datasets available in public domain. See an example set of datasets in the course resource tab.
3. Run 2-3 types of query workloads and measure the performance of the NoSQL system. Provide your analysis.

Deliverable.

1. URL to the NoSQL download and the datasets used
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.
4. Your analysis

**Option 2.**

This option is designed for students who are familiar with Big Table/NoQL/Key-Value Store Systems.

1. Download two of your favorite NoSQL (key-value) store software packages, such as HBase, Redis, LevelDB, Mongo, and so forth.

2. Populate the two system with the same example datasets. For example, you can use YCSB to generate dataset or use real world datasets available in public domain. See an example set of datasets in the course resource tab.

3. Run 2-3 types of query workloads and measure the performance of the two NoSQL systems. Provide your comparative analysis.

Deliverable.

1. URL to the NoSQL systems and the datasets used
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.
4. Your comparative analysis.