CMSC5702 – Advanced Topics in Parallel / Distributed Systems
Assignment 2: Parallel Programming
(Due Wednesday, 4 April 2023 at 18:30)

Objectives:

- To strengthen students' understanding of parallel computing concepts introduced in the lectures.
- To help the students gain hands-on experiences in designing and implementing a parallel distributed application.

Description:

For this assignment, you can choose to work on your own or in a group of 2 to 3 members to complete one of the topics below. A bonus mark of 5% will be given to the groups who have attempted Topic 2.

Topic 1:

You are required to write a MapReduce program to calculate the frequency of every link in all input files from the attached input directory, and sort the result by their frequency in descending order. You need to:

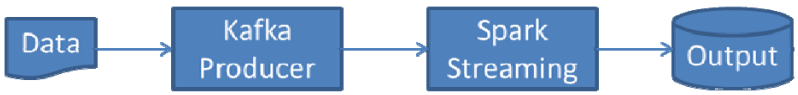
1. Use MapReduce to calculate the frequency of every links in all files, just like the "WordCount" of the links. (To simplify this work, you can use getAllStartTags("a") and then use getAttributeValue("href") in the htmlparser package to get the content of the links.)
2. After frequency calculation, sort the result by their frequency in descending order.
3. The links occurred less than six (≤ 5) times should be ignored.
4. Output each links followed by their frequency in each line (see the sample output).

You can start writing your program based on the WordCount example together with Jericho HTML parser to build a Java-based MapReduce program for running on Hadoop. Alternatively, you can also use any methods that you learnt from the class, e.g. OpenMP, MPI, etc, or use Scala or other programming language to write your program for running on Spark.

Topic 2:

The demand for stream processing is increasing a lot these days. The reason is that often processing big volumes of data is not enough. Data has to be processed fast, so that a firm can react to changing business conditions in real time. This is required for trading, fraud detection, system monitoring, and many other examples. In meeting these requirements, a combined architecture of Kafka + Spark Streaming has often been used.

You are required to set up a Kafka + Spark Streaming architecture as shown below where a Kafka producer reads a day worth of Apple share transaction history and simulates a real-time feeds into Spark Streaming process. The Spark Streaming process generates a statistics of high, low and volume of the day every 1 minute and stores the output in a file in 5 tuples, i.e. current time, current price, low, high, total volume. The Apple share transaction history can be downloaded from: https://www.nasdaq.com/market-activity/stocks/aapl/latest-real-time-trades



You can use the sample code like in https://gist.github.com/fancellu/f78e11b1808db2727d76 to create your Kafka producer and then use the DirectKafkaWordCount example as the base to build your Spark Streaming application.

Submission Guidelines:

On the due date, you or your group should submit the followings:

- JAR file and all source codes and any other necessary files.
- A readme file, which tells us how to install and test your program.

You should email your submission in a ZIP file to hartanto@cse.cuhk.edu.hk with the subject line "CMSC5702 Assignment 2". If the files are too large, you can put them in a public FTP and send the file location to us to download.

Grading:

This assignment will contribute 20% to the final marks. The markings will be based on:

- The correctness of your application (17%), e.g. the application making use of MapReduce pattern, can be compiled and run, won't crash due to any exceptions and deliver the right results.
- The completeness of the documentation (3%), e.g. the design of your program, the environment setup, the readme file on how to compile and run the application in order to get the results.

-- END --