HW3

PART 1. Read the following paper (attached), and write a short summary/report.

Google has, over the past few years, built out a massively scalable infrastructure for its search engine and other applications, including Google Maps, Google Earth, GMail, Google Finance, and Google Apps. Google's approach was to solve the problem at every level of the application stack. The goal was to build a scalable infrastructure for parallel processing of large amounts of data. Google therefore created a full mechanism that included a distributed file system, a column-family-oriented data store, a distributed coordination system, and a MapReduce-based parallel algorithm execution environment. Graciously enough, Google published and presented a series of papers explaining some of the key pieces of its infrastructure. The most important of these publications are as follows. Read the following papers, and write a short summary/report for each paper.

http://static.googleusercontent.com/media/research.google.com/en/us/archive/gfs-sosp2003.pdf
http://static.googleusercontent.com/media/research.google.com/en/us/archive/mapreduce-osdi04.pdf
http://static.googleusercontent.com/media/research.google.com/en/us/archive/bigtable-osdi06.pdf
http://static.googleusercontent.com/media/research.google.com/en/us/archive/chubby-osdi06.pdf

PART 2 – Programming Assignment

All hadoop commands are invoked by the bin/hadoop script. Running the hadoop script without any arguments prints the description for all commands.

Usage: hadoop [--config confdir] [--loglevel loglevel] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]

Execute each hadoop command once, and place the screenshots into a word file. If a command cannot be executed for any reason (such as, a distributed environment is needed), you may write the definition of the command, and skip execution.

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

PART 3 – Programming Assignment

Copy the attached 'access.log' file into HDFS under /logs directory.

Using the access.log file stored in HDFS, implement MapReduce in Hadoop to find the number of times each IP accessed the website.

PART 4 – Programming Assignment

Download and Copy all the files (http://msis.neu.edu/nyse/) (DailyPrices_A to DailyPrices_Z) to a folder in HDFS.

PART 4.1 – Write a MapReduce to find the Max price of stock_price_high for each stock. Capture the running time programmatically (or manually using a wristwatch or smartphone).

PART 4.2 – Write a Java Program to implement PutMerge as discussed in the class to merge the NYSE files in a single file on HDFS. Now, repeat 4.1 on the single merged-file. Capture the running time.

Did MapReduce on a single file run faster than running MapReduce on a bunch of files?

PART 5 – Programming Assignment

Write one MapReduce program using each of the classes that extend FileInputFormat<k,v> (CombineFileInputFormat, FixedLengthInputFormat, KeyValueTextInputFormat, NLineInputFormat, SequenceFileInputFormat, TextInputFormat)

http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/input/FileInputFormat.html

You could use any input file of your choice. The size of the input files is not important. The MR programs could simply do counting, or any other analysis you choose.