CSE 487/587 Programming Assignment #1

Due Date: Feb 28, 2015
Weight: 12% of your grade

Problem: (a) Use Mapreduce to compute the volatility of stocks in NASDAQ
(b) Evaluate the scalability of your implementation on different data sizes and number of nodes.

Description

In this assignment, you will use Mapreduce on a Hadoop environment at CCR to compute the monthly volatility of stocks. You are given daily data of 2970 stocks on NASDAQ market for 3 years from 01/01/2012 to 12/31/2014 (except holidays, otherwise called trading days). Imagine that you are a data analyst working for an investment company, your daily job is to analyze stock price data, and find out which stocks in a certain period have higher earnings potential, etc. One characteristic that is widely used by traders is the volatility index. You can get more details at http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:standard_deviation_volatility

Your data is 2970 CSV format files (Comma Separated). Each file contains the data for one stock using its symbol as the file name. A stock list file is also provided.

Data Format

For example, in the file **AAPL.csv**(Apple inc. stock), there are 7 columns, each row represents one day. The 5th column Close can be neglected(no use).
Date represents the date of the stock AAPL;
Open represents the open price in that day of stock AAPL;
High represents the highest price in that day of stock AAPL;
Low represents the lowest price in that day of stock AAPL;
Adj Close represents the close price in that day of stock AAPL;
Volume represents the volume in that day of stock AAPL;

Date	Open	High	Low	Close	Volume	Adj Close
12/31/2014	112.82	113.13	110.21	110.38	41403400	110.38
12/30/2014	113.64	113.92	112.11	112.52	29881500	112.52
12/29/2014	113.79	114.77	113.7	113.91	27598900	113.91
12/26/2014	112.1	114.52	112.01	113.99	33721000	113.99
12/24/2014	112.58	112.71	112.01	112.01	14479600	112.01

Calculate Stock Volatility of Monthly Rate of Return:

x_i = Monthly Rate of Return = (Month end adjusted close price – Month beginning adjusted close price) / (Monthly beginning adjusted close price)

volatility = $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$, $\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N}$, where N is set to 36 (months), ending close price is the close price of the last trading day in each month, beginning close price is the close price of the first trading day in each month.

Find

- the top 10 stocks with the lowest (min) volatility
- the top 10 stocks with the highest (max) volatility

Specific Submission Guidelines: Assignment 1

1. Files should be strictly organized as following structure in your own directory (/gpfs/classes/cse587/spring2015/students/username/hw1) and the naming of the directory should be followed exactly (case sensitive):

   NOTE THAT YOU SHOULD NOT MAKE ANY CHANGES TO THE DIRECTORY AFTER THE SUBMISSION DEADLINE AS THE TIME STAMP OF THE FILES WILL BE USED FOR TIMELY SUBMISSION.

   hw1/src/
   (include the source code (.java) of your mapreduce job)
   hw1/SLURMmyHadoop
   (the sample slurm script of your mapreduce job)
   hw1/mapreduce.jar
   (your runnable mapreduce jar file)
   hw1/*username*.pdf
   (your assignment report)
   hw1/lib *(optional)*
   (include the library you may have used in this assignment)
   hw1/bin *(optional)*
   (include compiled class files)
   hw1/misc/ (optional)
   (include any other files you may want to submit)

2. Your code will be evaluated using automated script in following fashion.

   [username@rush:~] sbatch SLURMmyHadoop <input file directory path> <output file directory path>

   for example,

   sbatch SLURMmyHadoop /gpfs/courses/cse587/spring2015/data/hw1/small ./output

So in your slurm script, you should use $1 as your input data path, $2 as your output data path. When testing your code, we will change the data (however, the structure of the data will be the same as given data set)

3. For your output, you should have only one ***part-r-00000*** file, in which the result (stock names) is listed, *i.e.,* the number of reducer in your last mapreduce job should be fixed as 1.

4. In your report, include execution time for the following cases.

Problem Size	Execution Time: 1 node (12 cores)	Execution Time: 2 nodes (24 cores)	Execution Time: 4 nodes (48 cores)
Small			
Medium			
Large			

- Execution time is the time taken for your entire ***main*** function. (Including any preprocessing you might have used)
- You should also submit a compressed tar file of your entire hw1 directory to UBLearns.

Grading Criteria

- Program correctness (working program): 50%
- Data Scaling: 12%
- Node Scaling: 12%
- Performance: 10%
- Report: 16%

The report should include:

1) Speedup graphs (time on one core/time on multiple cores) from 1, 2, and 4 nodes, each with 12 cores.
2) The above speedup graphs for each data set. You could put all of them in one graph.
3) Discussion on the performance of your program and its scalability.