Sam Pfeiffer

MATH 9898 - Big Data

Professor Andrew Sheppard

2/20/15

SCRUB

Command line input is the input filename and the number of threads. There is an optional third input parameter.

* If the third parameter is “no”, all operations take place on the original input file (bad data lines get their first character replaced with an ‘x’.) At the end of the program, this file is renamed to “signal.txt”.
* If the third parameter is left out or is anything other than “no”, the program first copies the input file into a file “signal.txt”. The rest of the program treats the signal.txt file as the input file (x’s are placed in the signal.txt file, not the input file). This is done so that the program can be rerun on the same input file without having to redownload the file.

I used mmap for reading the file. This uses virtual memory to access the file. I grab the first 100 lines of the data and find the median value for the date, time, price, and units. This provides the program with a starting data with which to compare following values with. After that, the “comparison” data is updated when new good data is read.

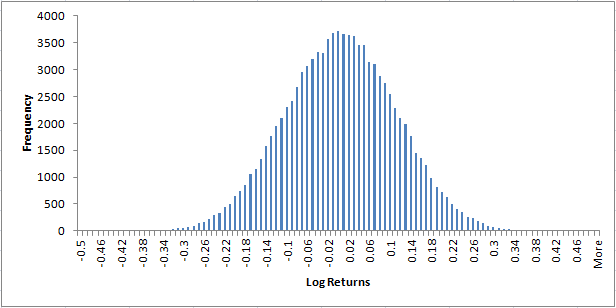
The file is split evenly between the threads. If a thread starts in middle of a line, that line is thrown out (negligible at this size).

Time is considered good if it’s less than 2 seconds away from the previous data’s time. Prices are good if it is positive and the percentage change from the previous data point is no more than 5 times the difference in time. Units is good if it is positive and no more than 5000 times the previous units.

The program tracks some important data and computing resource usage statistics in a logging file.

NORMAL

Command line input is the input filename and the number of threads. This program uses mmap in the same way that the scrub program does. The file is split evenly between the threads. If a thread starts in middle of a line, that line is thrown out, just like in the scrub program. I chose to use the Jarque-Bera normality test. It tests whether the skewness and kurtosis are the correct amounts to be normally distributed (0 and 3 respectively). Note that with large data sets, even small but consistent deviations from normality will lead to tests strongly indicating non-normality. This becomes obvious with this data set. Graphing some of the log returns of the data produces the following normal looking chart.



However, there are small deviations from normality which cannot be seen on such a scale. This results in the tests rejecting the hypothesis that the log returns are normally distributed. Note that in my normality program I chose not to sort the data by time. The reason is that according to the null hypothesis the log returns are normally distributed. Assuming that the reodering of the data is random, the distribution of the log returns should not be affected by the order of the prices.