CSE 587 HW # 3

Name: Daniel Nazareth
UbitName: dnazaret
Email: dnazaret@buffalo.edu

SUMMARY: The project uses the Apache PIG and HIVE frameworks each layered on Hadoop 2.6.0 to quickly and efficiently compute the most and least volatile stocks from small,medium and large sized NASDAQ stock datasets.

RATIONALE FOR USING PIG/HIVE: Pig and Hive are both frameworks sitting on top of Hadoop. The biggest reason for using either framework over traditional Hadoop MapReduce code (in say Java or Python) is that Pig Lating and HiveQL statements encapsulate huge chunks of MapReduce functionality into very simple syntax and readable code. In general Pig Lating and HiveQL scripts will use approximately 5% of the lines of code taken by traditional Java mapReduce to accomplish the same task. This means that development and QA time is reduced tremendously and data scientists are free to program short scripts at a high level without worrying too much about what's going on "under the hood". The only downside to this is that Pig/Hive scripts can run marginally slower than the corresponding Java mapreduce code.

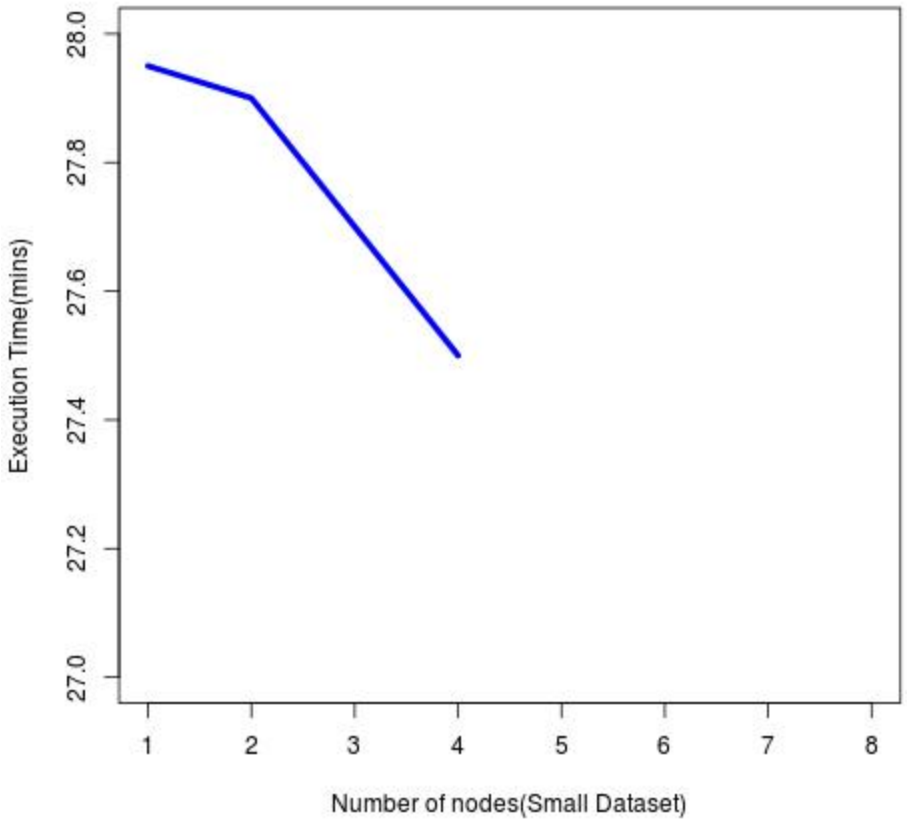
To summarise, Pig/Hive offer the following benefits:

- Shorter,simpler and much more readable code.
- Shorter development and QA cycles leading to quicker releases to production.
- Data scientist is free to program and think at a high level without worrying about low-level,specific details of the mapreduce.

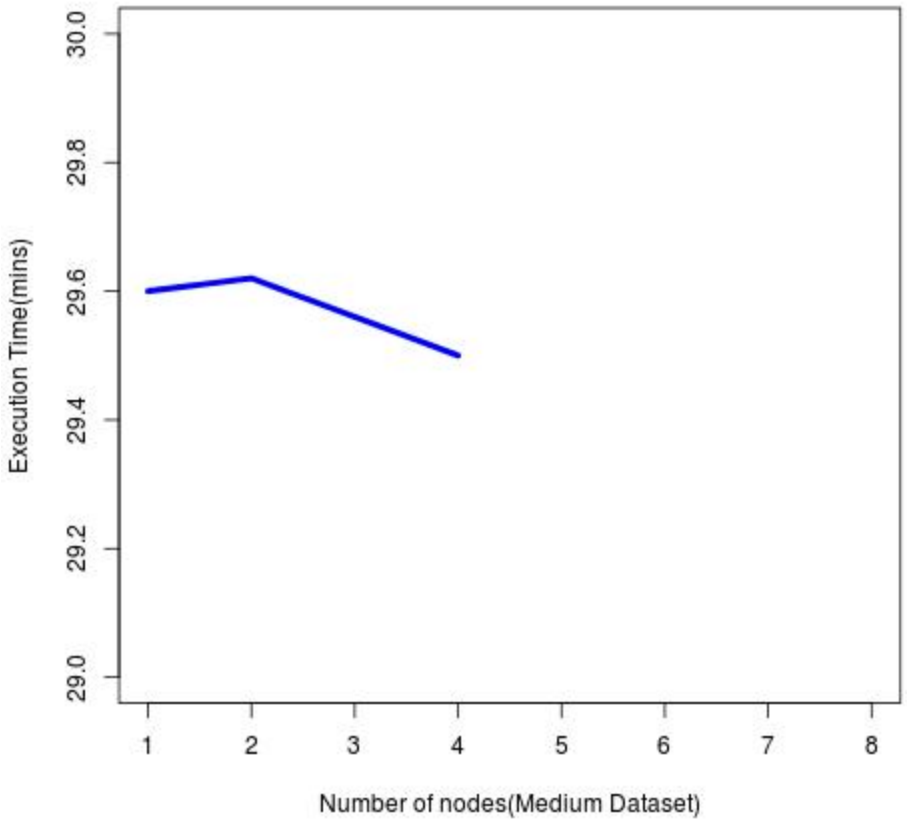
RESULTS: The results are summarised in tabular and graphical form below for each of the small,medium and large datasets. This can be verified by running the attached code at the command line as well.

PIG RESULTS

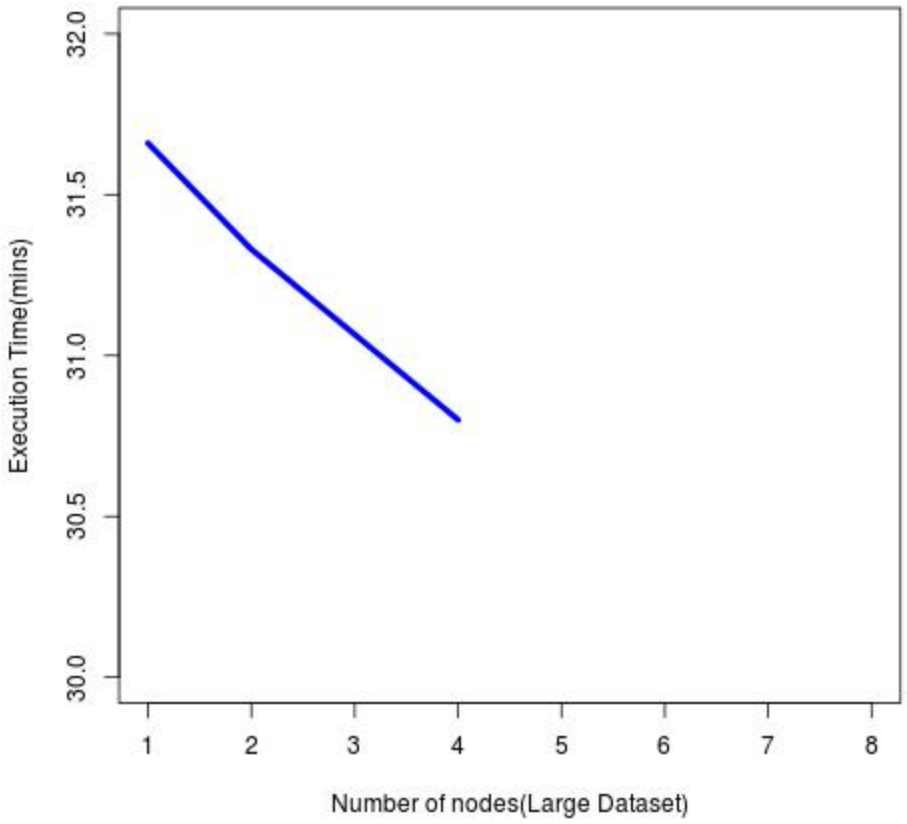
PROBLEM SIZE	EXECUTION TIME(1 NODE,12 CORES)	EXECUTION TIME(2 NODES,12 CORES)	EXECUTION TIME(4 NODES,12 CORES)
Small	27 mins,59 seconds	27 mins, 54 seconds	27 mins,37 seconds
Medium	29 mins,44 seconds	29 mins,45 seconds	29 mins,31 seconds
Large	31 mins, 46 seconds	31 mins,21 seconds	30 mins,49 seconds



MEDIUM DATASET:



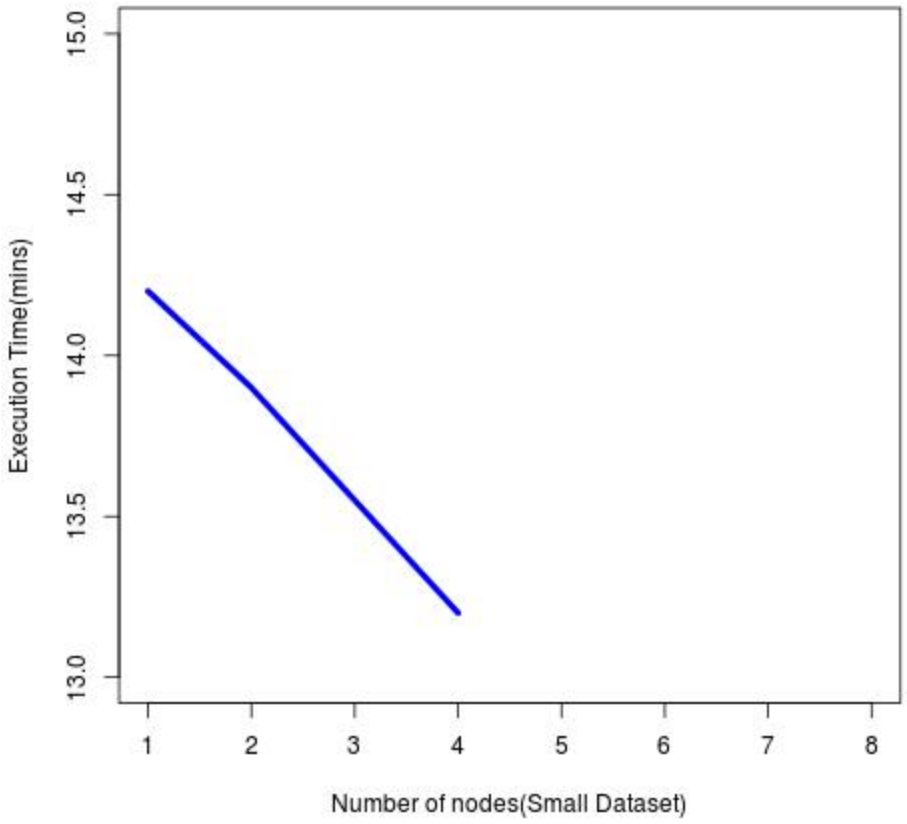
LARGE DATASET:



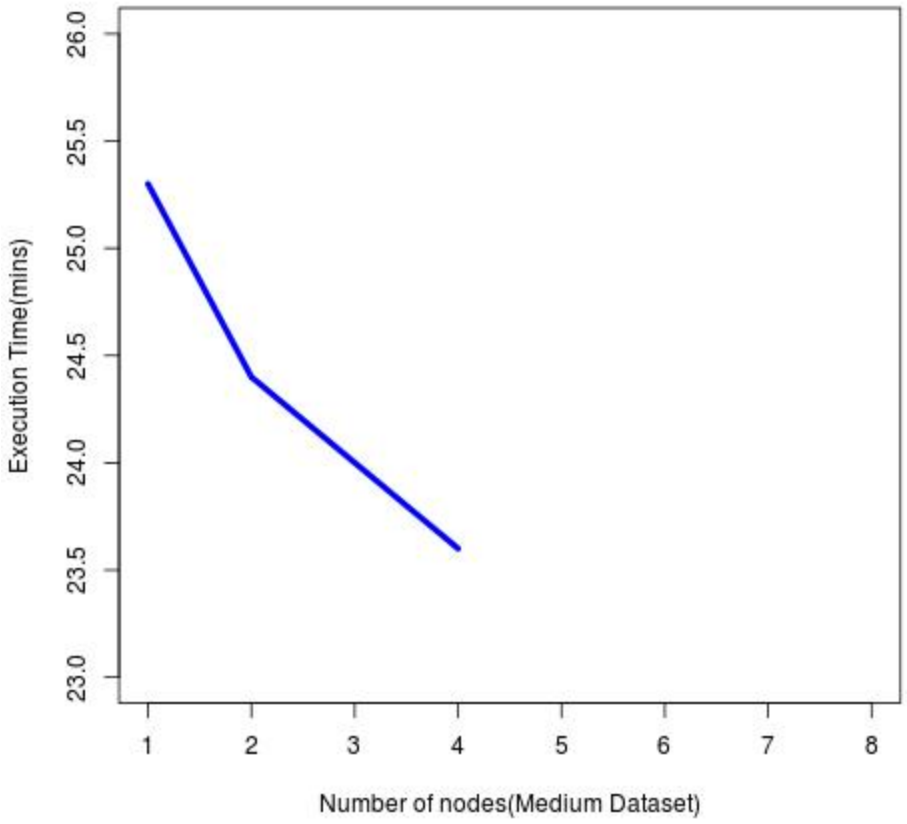
HIVE RESULTS

PROBLEM SIZE	**EXECUTION TIME(1 NODE,12 CORES)**	**EXECUTION TIME(2 NODES,12 CORES)**	**EXECUTION TIME(4 NODES,12 CORES)**
Small	14 mins,14 seconds	13 mins, 56 seconds	13 mins,15 seconds
Medium	25 mins,17 seconds	24 mins,22 seconds	23 mins,36 seconds
Large	Memory Related Exception: Garbage Collection Overhead Exceeded.		

SMALL DATASET GRAPH/SCALING:



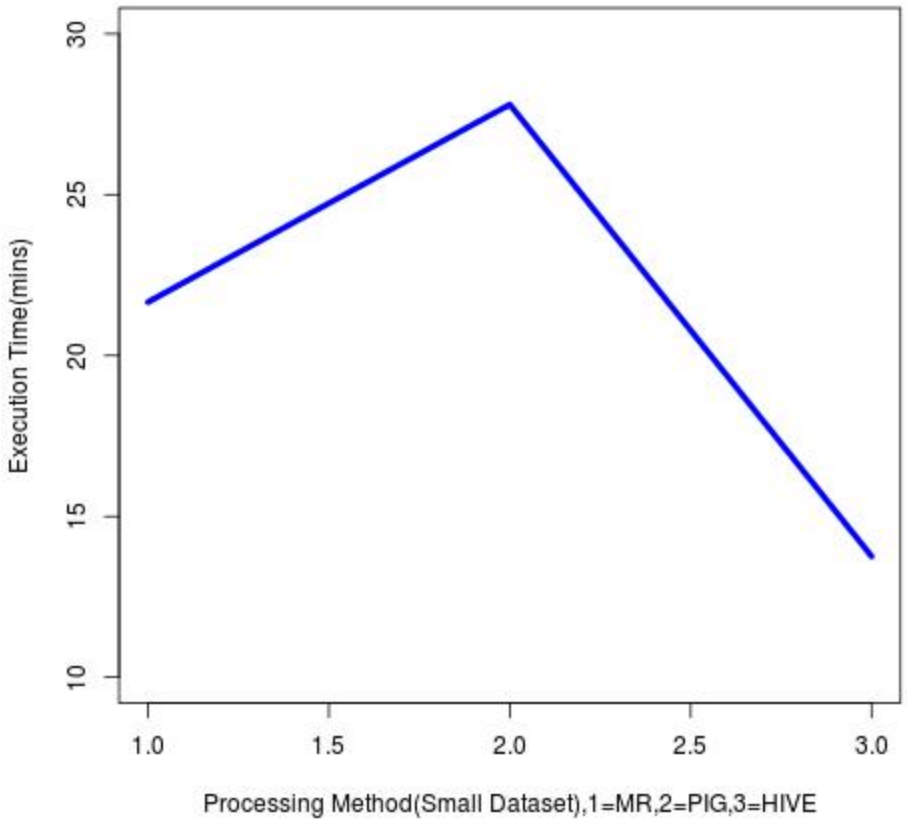
MEDIUM DATASET GRAPH/SCALING:



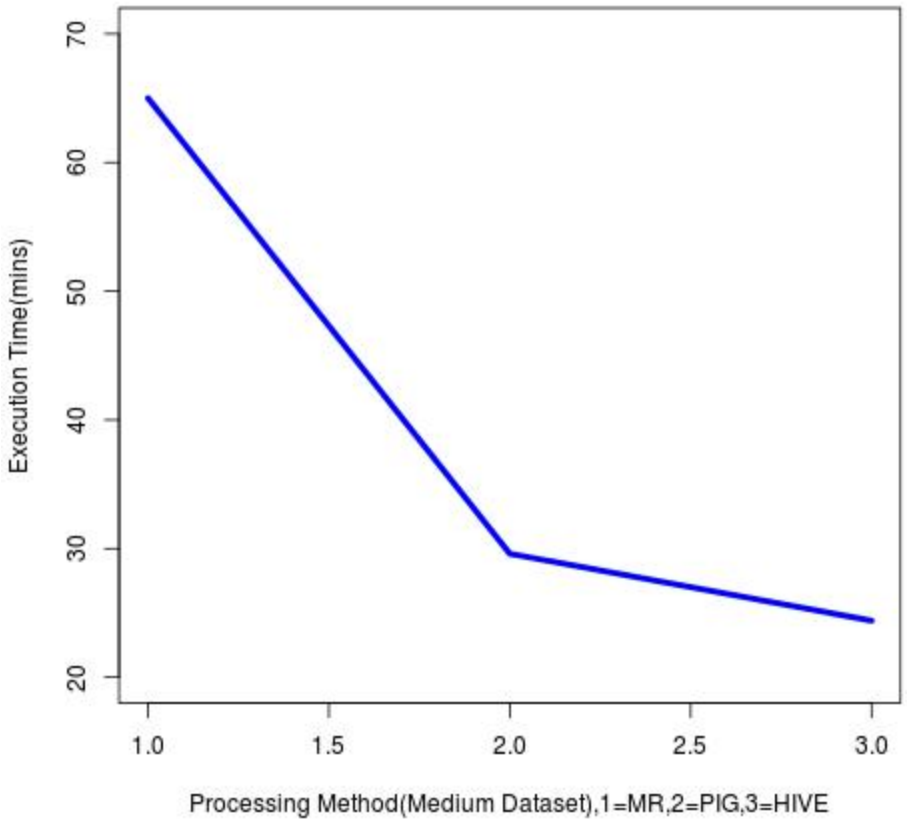
COMPARISON OF MAP REDUCE/PIG/HIVE: We see from the below tabular and graphical comparison that for small datasets, all 3 methods are closely matched with Hive being the runaway winner. For medium and large datasets Pig and Hive in particular offer significantly superior performance. However Map Reduce performance can be significantly improved by assigning more computing nodes-this cannot be done for Pig/Hive.

PROBLEM SIZE	**AVERAGE EXECUTION TIME(MAP REDUCE)**	**AVERAGE EXECUTION TIME(PIG)**	**AVERAGE EXECUTION TIME(HIVE)**
Small	21.66 minutes	27.8 mins	13.76 minutes
Medium	65 mins	29.6 mins	24.4 minutes
Large	120 mins	31.2 mins	Not Available

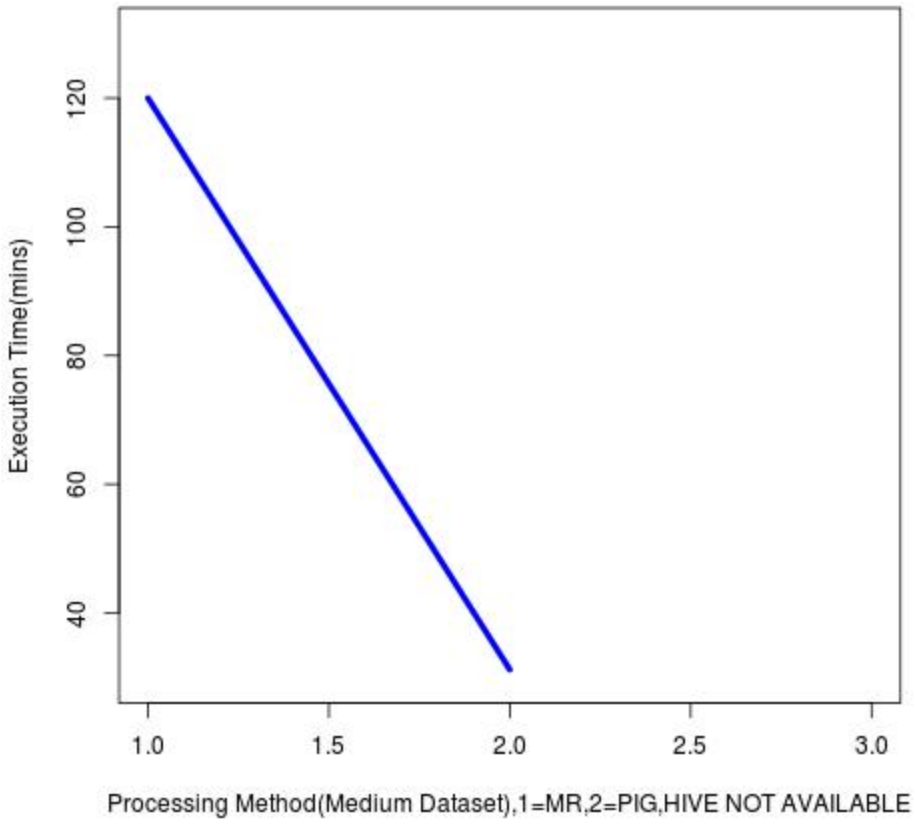
SMALL GRAPHICAL COMPARISON:



MEDIUM GRAPHICAL COMPARISON:



LARGE GRAPHICAL COMPARISON:



NOTES/ISSUES FACED:

- **We observe that additional nodes makes little or no difference to execution times for both Pig and Hive but a huge difference for Map Reduce. Times scale by small factors over dataset sizes but barely at all over number of nodes allocated. This would seem to indicate that for a given Hive or Pig script, computational efficiency cannot be reduced beyond a certain point, no matter how many resources are allocated.**
- Memory overhead seemed to be excessive for Hive large dataset and hence job could not complete on CCR. Possible improvements to code efficiency could possibly help remedy this problem.

REFERENCES

https://cwiki.apache.org/confluence/display/Hive/Tutorial

http://www.tutorialspoint.com/hive/hive_installation.htm

http://chimera.labs.oreilly.com/books/1234000001811/ch05.html#foreach_basic

https://pig.apache.org/docs/r0.7.0/piglatin_ref2.html