Report for Programming Assignment #3

# Stock Volatility Analysis Using PIG and HIVE

Harsh Harwani

CSE-587

## Method and Implementation

In this assignment we were given the data of 2970 stocks on NASDAQ market for 3 years from 01/01/2012 to 12/31/2014. The work happening in the specific jobs (Hive and Pig) is explained below.

- We have been provided with a data set of comma separated files with the monthly trading value of the stocks.
- The data rows are in descending order of the monthly dates. For ex. the data for a particular month starts from the last traded day in the month and ends with the first traded day of the month.
- Sample Input data:
  Date|Open|High|Low|Close|Volume|Adj Close
  - 2014-12-31,50.68,50.68,50.68,50.68,000,50.55
- I used the columns Date and Adj Close price for calculating the volatility of the stocks over the period of 3 years.

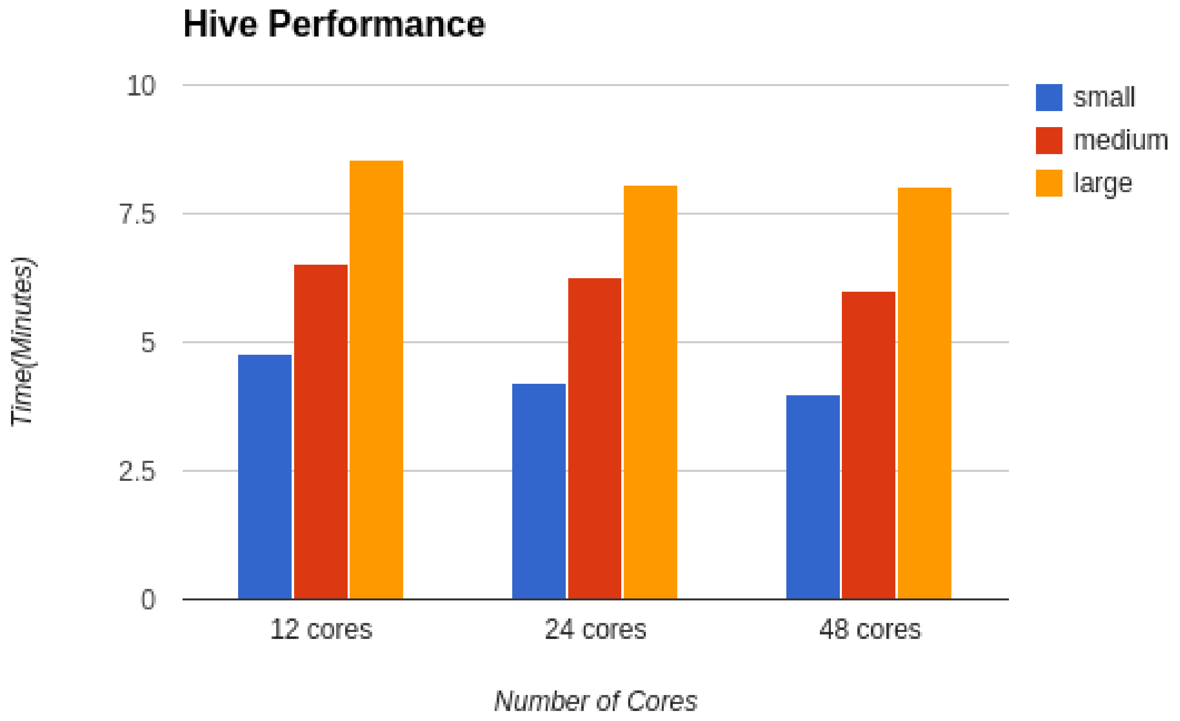
## HIVE

The method used in implementation of pig for calculating the volatility of stocks is as follows:

- Load data into external table from hdfs and remove the header using tableproperties.
- Filter the required data with file name into another table storing the date as substring of year and month also.
- Finding the min and max date using group by operation using the substring of year-month field.
- Applying the double join with the filtered all data table to find the adjusted close price for max and min date.
- Use the unbiased standard deviation inbuilt method of stddev_samp of the hive.
- Removed the 0 values and sort the data in both order and stored in a table
- Stored the sorted data into the  output files using the command in SLURM file.

## Performance Analysis:

| Problem Size | Execution Time(mins): 1 node (12 cores) | Execution Time(mins): 2 node (24 cores) | Execution Time(mins): 4 node (48 cores) |
|---|---|---|---|
| Small | 4.8 | 6.52 | 8.56 |
| Medium | 4.2 | 6.29 | 8.06 |
| Large | 4 | 6.01 | 8.04 |



As we can see in the graph performance of time taken in case of small dataset is approximately 4 minutes,6 minutes and 8 minutes for the small,medium and large dataset respectively. As the number of nodes increases there is not a significant reduction in execution time by which we can conclude when the data is not too large parallel execution does not decrease the time significantly.

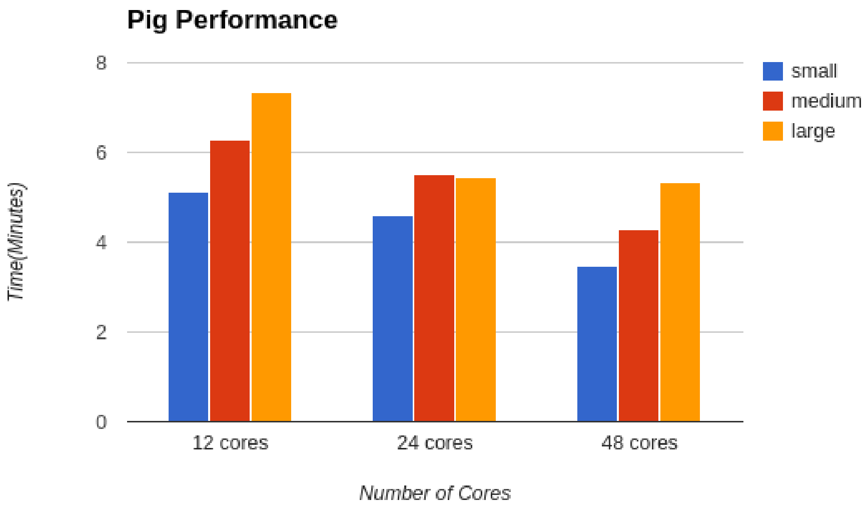
## PIG

The method used in implementation of pig for calculating the volatility of stocks is as follows:

- Load the data into the pig variable.
- Filtered the data, removing the header and collected the useful data.
- with the substring of date's year and month and day stored the data into bag of tuples.
- Using the values of tuple in the bag, sorted the data using the days ,grouped and flattened the output to extract the Adj close price on mindate and maxdate.
- Calculated value of xi AND xi square by grouping the last input by stock name
- Calculated the sum of xi square and the total months for that stock.
- Then there are multiple steps including Join, Foreach and Group statements to find the final volatility.
- Filtered the stocks with zero volatility and found the top 10 stocks with maximum volatility and minimum volatility.
- The results were then stored in the output file.

## Performance Analysis:

| Problem Size | Execution Time(mins): 1 node (12 cores) | Execution Time(mins): 2 node (24 cores) | Execution Time(mins): 4 node (48 cores) |
|---|---|---|---|
| Small | 5.11 | 6.29 | 7.34 |
| Medium | 4.58 | 5.52 | 5.45 |
| Large | 3.47 | 4.29 | 5.35 |



As we can see in the graph above,the performance of time taken in case of small dataset is approximately 4 minutes,6 minutes and 8 minutes for the small, medium and large dataset respectively.

## Map-Reduce

## Performance Analysis:

| Problem Size | Execution Time(mins): 1 node (12 cores) | Execution Time(mins): 2 node (24 cores) | Execution Time(mins): 4 node (48 cores) |
|---|---|---|---|
| Small | 40.73333333 | 17.9 | 9.016666667 |
| Medium | 120.8 | 52.18333333 | 25.61666667 |
| Large | 408.7666667 | 172.4166667 | 83.2333 |

## Comparison and Discussion

- As we can observe the processing time in case of pig and hive for all the data sets is significantly less than map-reduce.
- In case of hive and pig the performance is approximately 101% in case of large dataset.
- The time required to process the large dataset decreases by about 100%.
- Performance of pig and hive are both almost equal for the three datasets.
- Pig is more of a data flow language and has concepts like bags,tuples,maps,I think its pig is more of a step by step data flow language where one can go step by step and reach the final results.
- As a result of the step by step data flow approach provided by pig it is easy to checkpoint data in case of pig
- Hive is very much similar to Sql, so the person knowing SQL will be very much comfortable with HIVE.
- Using hive and pig simple tasks can be developed quickly and efficiently.Pig and hive also support user defined functions where the developer needs to develop certain things which are not possible by queries.
- HIVE and PIG are better than Map-Reduce for simpler tasks but when the tasks are complicated,one has to use Map-Reduce.
- In conclusion for simpler tasks hive and pig definitely reduce the development time and the effort required but in case of more complex tasks Map-Reduce has to be used and with efficiently manipulating various map-reduce parameters one can achieve better performance in map-reduce as well.