

VOLATILITY INDEX

1) IMPLEMENTATION

The project is implemented using two mapper and reducer chaining. First mapper reads the input line by line and gives outputs key as the company name and date with adjusted close price as value. The task of first reducer is to collect outputs from mapper and calculate Monthly rate of return of the company and after calculation Rate of return of each month, finally we are calculating Volatility index for that share. First reducer then write into a temporary file with company name as Key and Volatility index as value.

Now second mapper will read the output file of first mapper reducer pair it will emit all the keys to be same and value to be volatility concatenated with the stock name. In the reducer these iterable values will be added to a array which we will sort and get Top and Bottom 10 stocks according to volatility index.

2) PERFORMANCE AND EXECUTION LABELS

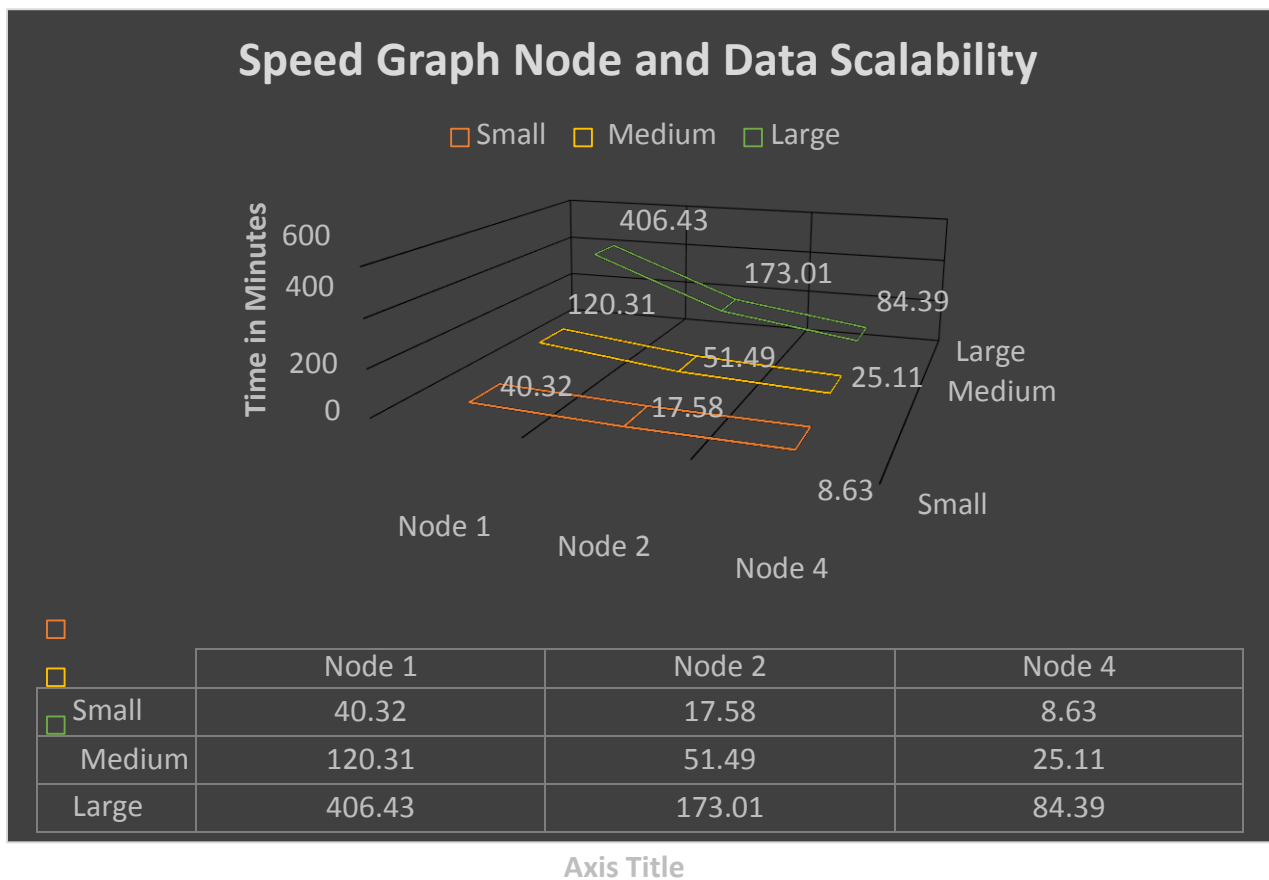
Problem Size	Execution Time: 1 node (12 cores)(minutes)	Execution Time: 2 nodes (24 cores)(minutes)	Execution Time: 4 nodes (48 cores)(minutes)
Small	40.32	17.58	8.63
Medium	120.31	51.49	25.11
Large	406.43	173.01	82.50

By measuring the values we can conclude that by increasing the data set size i.e. from small to medium to large the execution times on all node columns increases. Secondly we can say that as number of nodes increases for same data set size the performance increases. The execution

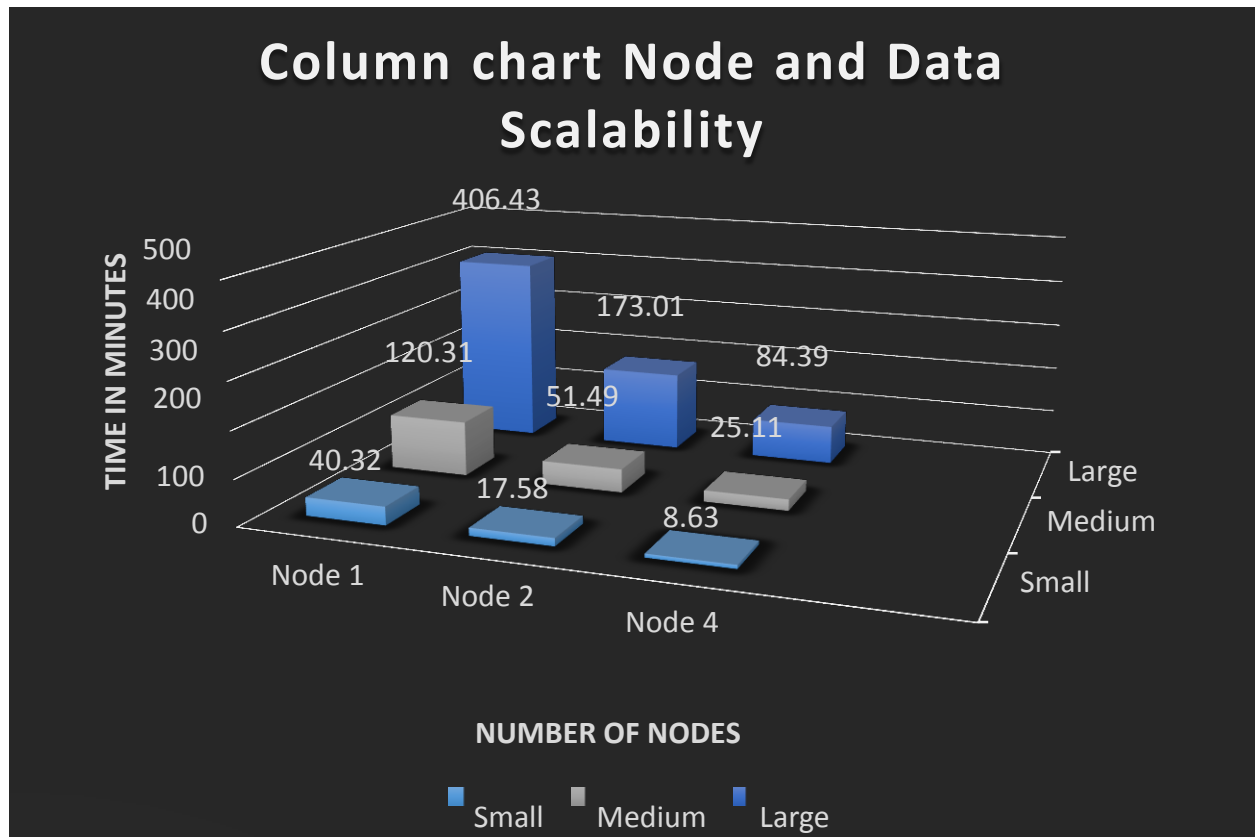
time by more than half if number of nodes are multiplied by 2. Hence as the number of nodes increase the performance also increase. The implementation is also scalable as it runs when data set is increased.

3) GRAPHS AND SCALABILITY

- a) Composite Line Graph Representation- x-axis->Number of nodes, y-axis-> Time in minutes.



b) Column chart representation.



4) OUTPUT VALUES

a) Small data set

```
ankitgoy@k07n14:~  
Bottom Stocks 10.0  
LDRI 5.149337E-4  
GAINO 5.650074E-4  
VGSH 0.0013014906  
MBSD 0.0025000459  
TRTLU 0.0034781051  
AGZD 0.0039385939  
SKOR 0.0039487402  
CADT 0.0041566362  
AXPWW 0.0044388373  
VCSH 0.0046377602  
Top Stocks 10.0  
ACST 9.2715897619  
NETE 5.3962539615  
XGTI 4.5423443115  
TNXP 3.2483321968  
EGLE 3.0222065379  
PTCT 1.8462537016  
GOGO 1.7793421752  
MEILW 1.7188134066  
ROIQW 1.396532084  
CFRXZ 1.079268245  
~  
"myoutput-3465091/part-r-00000" 22L, 397C
```

b) Medium data set

```
ankitgoy@k07n14:~  
Bottom Stocks 10.0  
LDRI-1 5.149337E-4  
LDRI-2 5.149337E-4  
LDRI-3 5.149337E-4  
GAINO-1 5.650074E-4  
GAINO-2 5.650074E-4  
GAINO-3 5.650074E-4  
VGSH-1 0.0013014906  
VGSH-2 0.0013014906  
VGSH-3 0.0013014906  
MBSD-1 0.0025000459  
Top Stocks 10.0  
ACST-3 9.2715897619  
ACST-2 9.2715897619  
ACST-1 9.2715897619  
NETE-3 5.3962539615  
NETE-2 5.3962539615  
NETE-1 5.3962539615  
XGTI-3 4.5423443115  
XGTI-2 4.5423443115  
XGTI-1 4.5423443115  
TNXP-3 3.2483321968  
~  
"myoutput-3465827//part-r-00000" 22L, 432C
```

c) Large data set

```
ankitgoy@k07n14:~  
Bottom Stocks    10.0  
LDRI-1  5.149337E-4  
LDRI-10 5.149337E-4  
LDRI-2   5.149337E-4  
LDRI-3   5.149337E-4  
LDRI-4   5.149337E-4  
LDRI-5   5.149337E-4  
LDRI-6   5.149337E-4  
LDRI-7   5.149337E-4  
LDRI-8   5.149337E-4  
LDRI-9   5.149337E-4  
Top Stocks      10.0  
ACST-9  9.2715897619  
ACST-8  9.2715897619  
ACST-7  9.2715897619  
ACST-6  9.2715897619  
ACST-5  9.2715897619  
ACST-4  9.2715897619  
ACST-3  9.2715897619  
ACST-2  9.2715897619  
ACST-10 9.2715897619  
ACST-1  9.2715897619  
~  
"myoutput-3465830/part-r-00000" 22L, 427C
```