Assignment 4

Inderpreet Singh Chera - 160101035

Vector Matrix Multiplication

Three different implementations were implemented which are

- Row division matrix multiplication
- Column division matrix multiplication
- Block division matrix multiplication

The experiments were done 1, 4, and 16 processes so that we can square root of no of processes as an integer.

System Settings

All tests were done on **Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz** processor. This computer is **dual core** where each core has **2 threads**. Also during each experiment it was insured that **no other applications** were running in the background, so that we don't have any biased readings.

Implementations:

Let's say we want to do following, matrix vector multiplication

$$\mathbf{A^k_{n \times n} x_{n \times 1} = b_{n \times 1}}$$

Here, we want to multiply x vector to A vector k times. In the implementations of code, different values of n were used and k was taken 50. Moreover, A was taken as an identity matrix but this 2-D matrix was converted to 1-D array for easy calculations. X vector was initialised to all 1s.

Row division matrix multiplication

In row division matrix multiplications we divide given matrix A such that each process gets n/p rows of matrix A (p are the total number of processes) and n/p rows of x vector. Hence at the start of each iteration, we first gather full x at each process and then do the required multiplication. Finally, the root process gathers the final vector from all the processes to give the final output.

Column division matrix multiplication

In column division matrix multiplications we divide given matrix A such that each process gets n/p columns of matrix A (p are the total number of processes). Here to ease the implementation we took the transpose of the matrix A, so that we can easily divide columns of matrix A and hence implemented assuming that we have taken the transpose of matrix A.

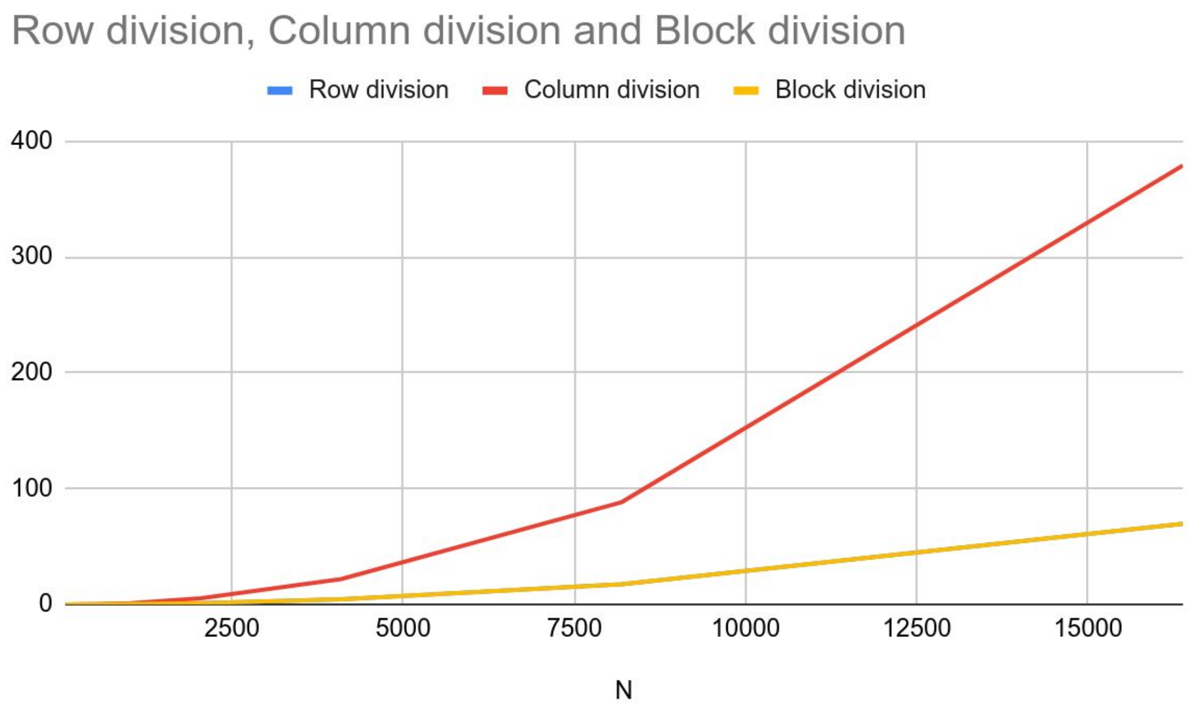
Block division matrix multiplication

In block division matrix multiplication, we divide matrix into blocks with size n / $\sqrt{p}$ X n / $\sqrt{p}$ where p is the number of processes. The processes are arranged in the grid with size $\sqrt{p}$ X $\sqrt{p}$, and each process is given one block. Now vector x is divided in n / $\sqrt{p}$ rows and given to the last column of the processes grid. Now at the start of each iteration, the first last column gives the vector x to the corresponding diagonal process in the grid. Then this diagonal process sends the part of the x that it got in their own column. Now each process has got enough info to do matrix multiplication and then finally the values got after matrix multiplication are send to the last column for final addition.

Observations:

Comparison using 1 process:

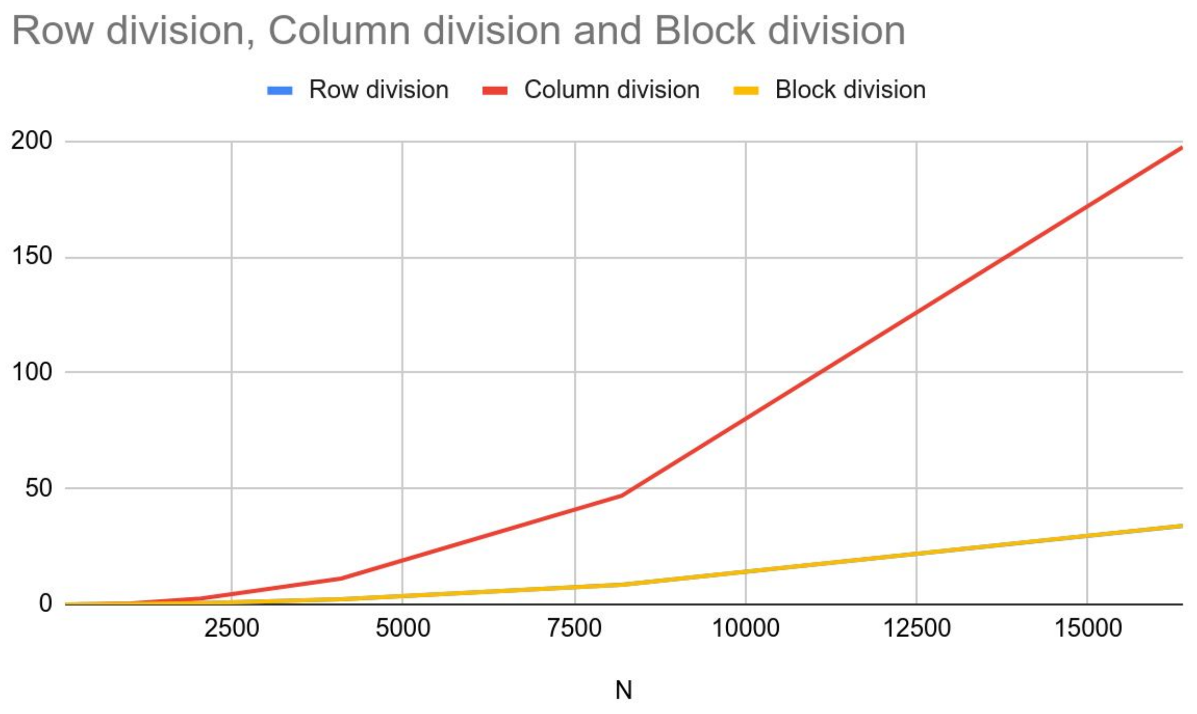
N	Row division	Column division	Block division
64	0.002113	0.002624	0.001148
128	0.006431	0.007324	0.007424
256	0.017479	0.025888	0.020773
512	0.070173	0.107524	0.067705
1024	0.272381	1.018256	0.271589
2048	1.08859	5.319183	1.091273
4096	4.340923	21.865534	4.346137
8192	17.403694	88.336097	17.366999
16384	69.566916	379.078725	69.490681



We observe almost exactly the same time between block and row vector matrix multiplication, so much so that the two lines in above graph overlaps, whereas time taken by column vector matrix multiplication is very large.

Comparison using 4 processes:

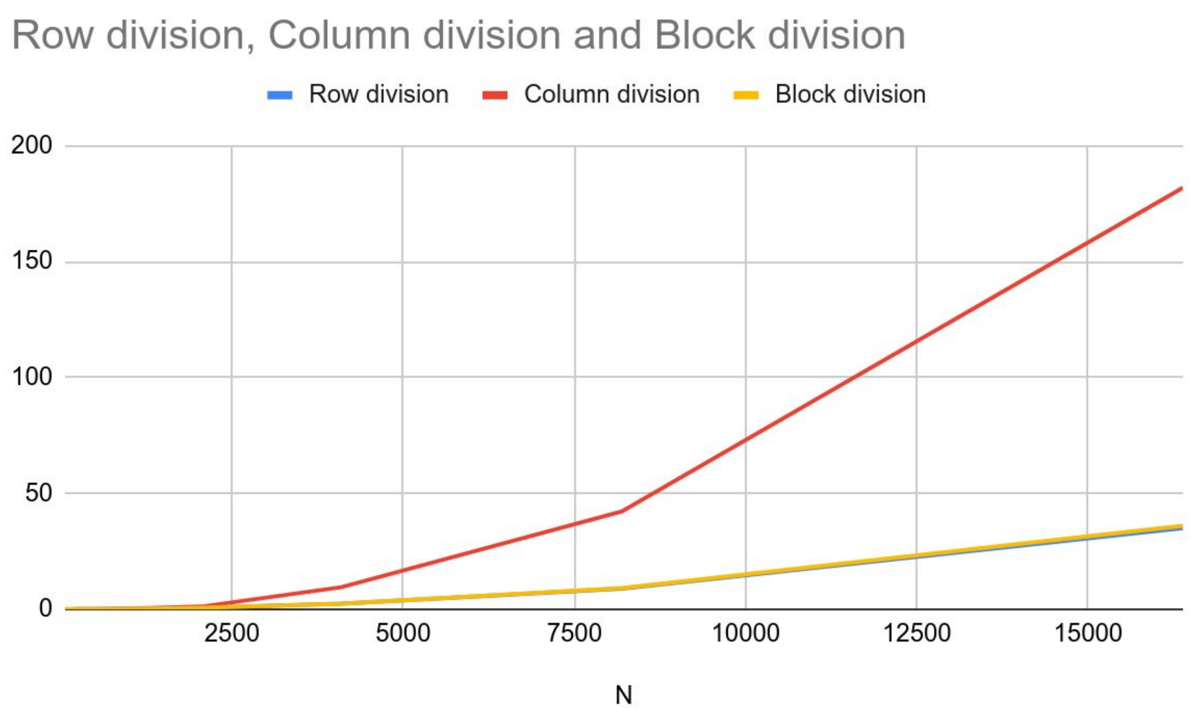
N	Row division	Column division	Block division
64	0.001259	0.001893	0.00114
128	0.002479	0.002662	0.002585
256	0.017902	0.009585	0.00892
512	0.03367	0.053789	0.033521
1024	0.138899	0.423423	0.133221
2048	0.533224	2.519843	0.538908
4096	2.174926	11.166487	2.142339
8192	8.523229	46.899947	8.509205
16384	33.895759	197.432966	33.957784



We observe almost exactly the same time between block and row vector matrix multiplication, so much so that the two lines in above graph overlaps, whereas time taken by column vector matrix multiplication is very large.

Comparison using 16 processes:

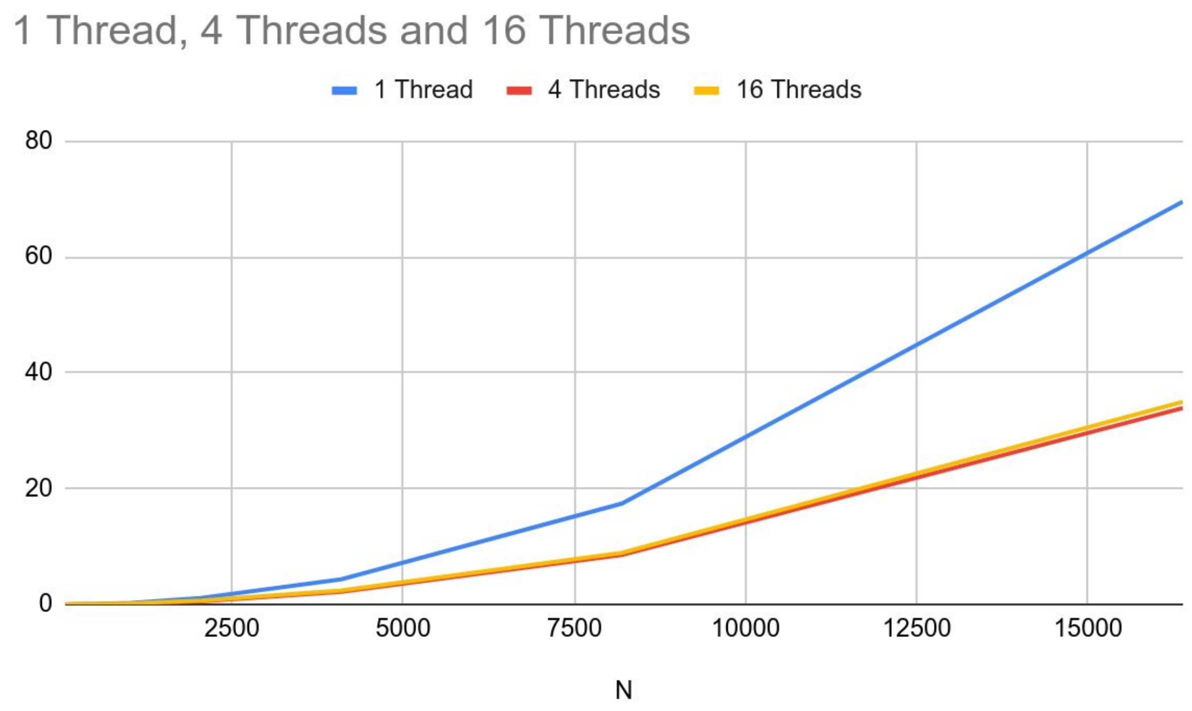
N	Row division	Column division	Block division
64	0.006181	0.010729	0.021587
128	0.008542	0.011322	0.039482
256	0.021461	0.027487	0.046834
512	0.056403	0.058836	0.088842
1024	0.169532	0.311958	0.217912
2048	0.675465	1.135899	0.630953
4096	2.418267	9.531073	2.424544
8192	8.902495	42.228468	9.135908
16384	35.008411	181.936508	36.062202



As seen in previous results, column division vector matrix multiplication takes a lot larger time than the other two. Block division vector matrix multiplication takes a little more time than the row division vector matrix multiplication.

Comparison between number of processes taking row division matrix multiplication:

N	1 Thread	4 Threads	16 Threads
64	0.002113	0.001259	0.006181
128	0.006431	0.002479	0.008542
256	0.017479	0.017902	0.021461
512	0.070173	0.03367	0.056403
1024	0.272381	0.138899	0.169532
2048	1.08859	0.533224	0.675465
4096	4.340923	2.174926	2.418267
8192	17.403694	8.523229	8.902495
16384	69.566916	33.895759	35.008411



Using 1 process takes a lot more time than 4 and 16 processes. And using 4 processes is a bit better than using 16 processes.