

High Performance Computing

Lab-1 Report by Hrishikesh Vedantam

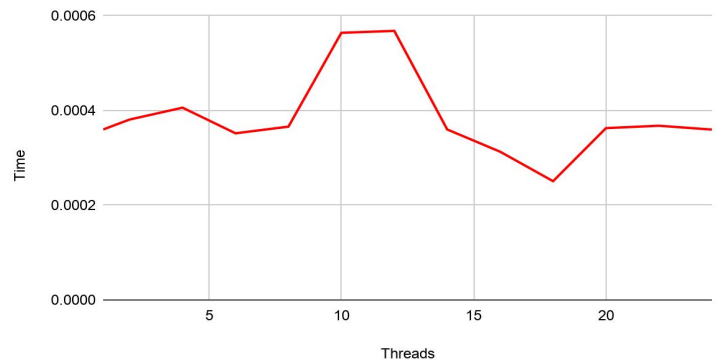
- OpenMP

- Matrix Addition

- This problem focuses on adding 2 large matrices with double precision values and calculates time taken for the entire process after parallelising it by multiple thread assignment.
 - It takes the least amount of time with **18** threads
 - The parallel fraction of this operation is **0.6966878882**

Threads	Time
1	0.000360012
2	0.000380993
4	0.000406027
6	0.000351906
8	0.000365973
10	0.000564098
12	0.000568151
14	0.000360012
16	0.000313044
18	0.000250816
20	0.000362873
22	0.00036788
24	0.000359774

Matrix Addition



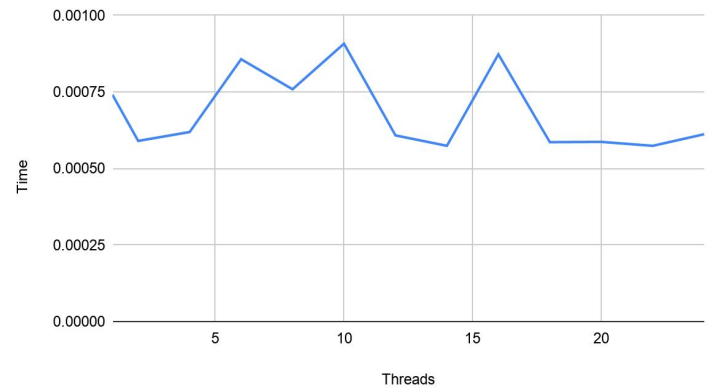
- Matrix multiplication column major order

- This problem focuses on calculating time taken by different thread limits in parallelising the multiplication of a large matrix of double precision values using Column major order.
 - It takes the least amount of time with **14** threads.

- The parallel fraction of this operation is **0.6966878882**

Threads	Time
1	0.000741005
2	0.000590086
4	0.000619173
6	0.000857115
8	0.000758886
10	0.000907898
12	0.000608206
14	0.000574112
16	0.000873089
18	0.000586033
20	0.000586987
22	0.000574112
24	0.00061202

Matrix Multiplication Column Major



○ Matrix Multiplication Block based approach

- This problem focuses on calculating time taken by different thread limits in parallelising the multiplication of a large matrix of double precision values using block based multiplication method.
- I have used a block size of **10**.
- It takes the least amount of time with **10** threads.
- The parallel fraction of this operation is **0.5276536912**

Threads	Time
1	0.0237961
2	0.024451
4	0.019536
6	0.020968
8	0.0246129
10	0.0125561
12	0.020215
14	0.0216451
16	0.026293
18	0.0222981
20	0.0205131
22	0.0355959
24	0.0169032

Matrix Multiplication Block method(block =10)

