

# ME766- Assignment2 Report

Dikshant

180040033

Q1.) Create two matrices, A and B, each of size  $(N \times N)$ . Initialise the matrices to random floating point numbers. Write an OpenMP code for computing  $C = AB$  and then transforming C into an upper triangular matrix. Report the times taken for the codes. Vary the size of the problem from  $N = 100 \dots 10000$

Sol:

N	Time(1)	Time(2)	Time(3)	Time(Avg)(s)
100	0.109s	0.074s	0.102s	0.095
500	0.629s	0.671s	0.652s	0.651
1000	5.244s	4.982s	5.506s	5.244
2000	55.536s	55.335s	57.038s	55.969
5000	19m 8.28s	20m 21.22s	19m 53.45s	1187.65
7000	55m 53.25s	57m 14.36s	56m 25.62s	3391.07
10000	/tmp/ccXBzJlo.o: In function `__static_initialization_and_destruction_0(int, int)': q1.cpp:(.text+0x202): relocation truncated to fit: R_X86_64_PC32 against `.bss' q1.cpp:(.text+0x215): relocation truncated to fit: R_X86_64_PC32 against `.bss' collect2: error: ld returned 1 exit status			

When the size was increased to 10000, my PC was not able to decompose the array further. I searched on google for the possible errors and they have mentioned these points: The recommended flags to use when you have >2GB of static data are:

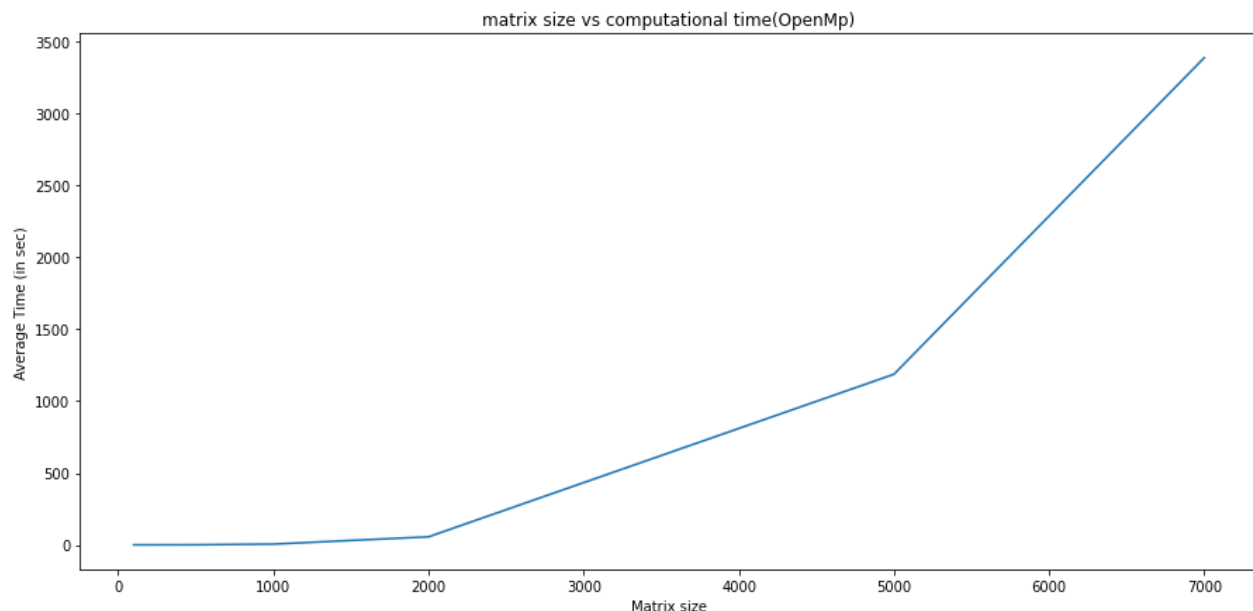
-mcmmodel medium -shared-intel (-mcmmodel large -shared-intel should also work, the only difference is that it also allows your code size to be >2GB, which is pretty rare).

I think the confusion arises because libraries that have not themselves been built with -mcmmodel medium, (because they do not contain > 2 GB of static data), must instead be built with -fpic so that they use full addressing and can be linked with objects that were built with -mcmmodel medium. Otherwise, you get the dreaded "relocation error" from the linker.

The Intel compiler runtime library is not built with -mcmmodel medium. The shared version of the runtime library is built with -fpic, but the static version is not. This is why you need to use the flag -shared-intel when compiling your code with -mcmmodel medium.

Here's just a snippet from my pc which shows that all four cores are on work:

```
dikshant@dikshant-pc: ~/projects/assignment2
1: [|||||100.0%]
2: [|||||100.0%]
3: [|||||100.0%]
4: [|||||100.0%]
Mem [|||||5.37G/7.66G]
Swp [|||||39.5M/10.2G]
Tasks: 219, 1147 thr; 4 running
Load average: 5.58 6.13 5.54
Uptime: 03:19:16
q1.cpp
q2_test.cpp #define N 5000
```



Q2.)

Now for the same problem, write an MPI code for computing  $C = AB$  and then transforming  $C$  into an upper triangular matrix. Report the times taken for the codes. Vary the size of the problem from  $N = 100 \dots 10000$ . Vary the number for the MPI process from 2 to 8.

Sol:

N=100

MPI Process	Time(1)	Time(2)	Time(3)	Time(Avg)(s)
2	1.452s	1.474s	1.495s	1.474
4	0.963s	0.979s	0.852s	0.931
6	0.561s	0.523s	0.550s	0.545
8	0.457s	0.502s	0.490s	0.483

N=500

MPI Process	Time(1)	Time(2)	Time(3)	Time(Avg)(s)
2	2.007s	2.055s	2.476s	2.179
4	1.218s	1.277s	1.363s	1.286
6	0.802s	0.986s	0.884s	0.891
8	0.720s	0.757s	0.766s	0.747

N=1000

8sMPI Process	Time(1)	Time(2)	Time(3)	Time(Avg)(s)
2	5.624s	5.460s	6.015s	5.579
4	4.099s	3.904s	4.244s	4.082
6	3.838s	3.583s	3.725s	3.715
8	3.589s	3.615s	3.805s	3.669

N=2000

MPI Process	Time(1)	Time(2)	Time(3)	Time(Avg)(s)
2	1m 22.276s	1m 23.531s	1m 21.952s	82.586
4	1m 8.868s	1m 8.904s	1m 11.432s	69.734
6	1m 1.254s	59.842s	1m 0.144s	60.413
8	56.532s	59.043s	58.614s	58.063

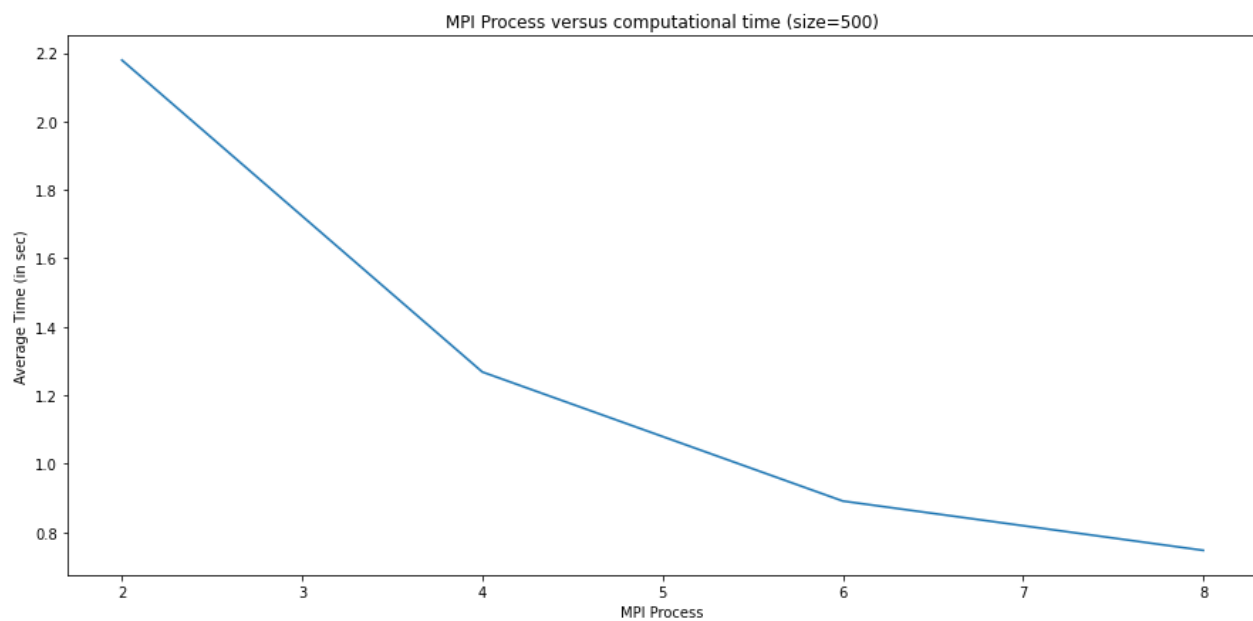
N=5000

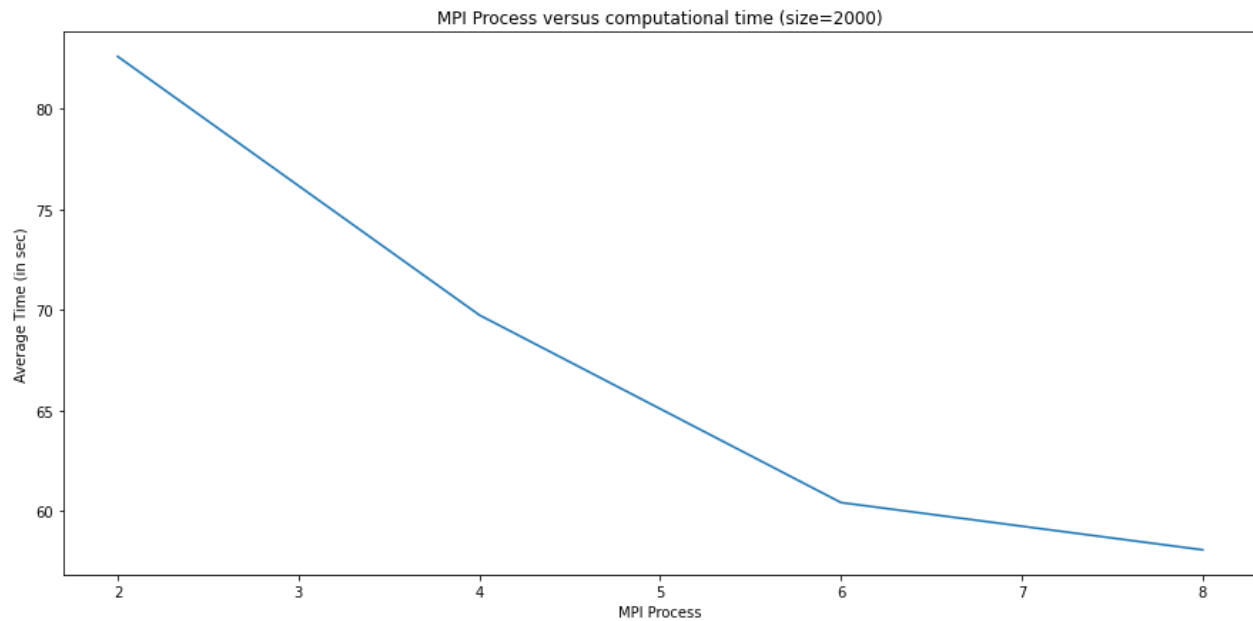
MPI Process	Time(1)	Time(2)	Time(3)	Time(Avg)(s)
2	18m 35.687s	19m 21.154s	18m 55.141s	1137.327
4	16m 28.443s	17m 0.021s	16m 51.232s	1006.565
6	15m 42.143s	15m 25.231s	15m 44.265s	937.213
8	15m 11.568s	14m 58.232s	15m 5.884s	905.228

N=10000

```
q2.cpp:(.text+0x2ff): additional relocation overflows omitted from the output
collect2: error: ld returned 1 exit status
```

As an illustration, I have shown how on increasing the number of processes effects the computation time for 2 cases (matrix size = 500 & 2000)





Since the matrix size is too high so I have shown my output for a small matrix size.

```

q1.cpp q2.cpp x
q2.cpp > mpi_decomposition(double(&)) [N][N], double(&)) [N][N], double(&)) [N][N]
1 #include <mpi.h>
2 #include <bits/stdc++.h>
3
4 using namespace std;
5
6 #define N 3
7 #define MASTER 0
8 #define MASTER_TASK 1
9 #define WORKER_TASK 2
10
11 double A[N][N], B[N][N], C[N][N], a[N][N], l[N][N], u[N][N];
12 int tasks, p, id, id2, workers, source, dest, mtype,
13 MPI_Status status;
14
15 void forw_elim(double **origin, double *master_row, s
16 {
17     if (**origin == 0)
18         return;
19     double k = **origin / master_row[0];
20
21     int i;
22     for (i = 1; i < dim; i++) {
23         (*origin)[i] = (*origin)[i] - k * master_row[i];
24     }
25     **origin = k;
26 }
27
28 void mpi_decomposition(double (&a)[N][N], double (&l)[N][N], double (&u)[N][N], int size)
29 {
30     MPI_Comm_rank(MPI_COMM_WORLD, &id2);
31     MPI_Comm_size(MPI_COMM_WORLD, &p);
32     int mx_size = N;
33     int tmp_size = mx_size - 1;
34     int diag_ref = 0;
35     std::cout << "\n***** decomposition going on *****\n";
36     if (id2 == MASTER) {
37         for (int i = 0; i < tmp_size; i++, diag_ref++) {
38             double *diag_row = a[diag_ref * mx_size + diag_ref];
39
40             // Matrix A
41             1 0 2
42             1 0 2
43             2 2 0
44
45             // Matrix B
46             1 1 1
47             0 1 1
48             1 1 0
49
50             // Message sent to worker for multiplication
51             ***** computation going on *****
52             // Message received by master after multiplication
53
54             // Matrix obtained after multiplication:
55             3 3 1
56             3 3 1
57             2 4 4
58
59             // Let's start matrix decomposition:
60             ***** decomposition going on *****
61
62             // Matrix obtained after decomposition
63             1 1 2
64             0 1 0
65             0 0 3
66
67             task done!!
68             dikshant@dikshant-pc:~/projects/assignment2$

```

## Machine Configuration:

```
dikshant@dikshant-pc:~/projects/assignment2$ sudo lshw
[sudo] password for dikshant:
dikshant-pc  int tasks, p, id, id2, workers, source, dest, mtype, rows, averow, extra, offset, rc;
description: Notebook
product: X556UQK
vendor: ASUSTeK COMPUTER INC.
version: 1.0
serial: H5N0CV109922218
width: 64 bits
capabilities: smbios-3.0 dmi-3.0 smp vsyscall32
configuration: boot=normal chassis=notebook family=VivoBook uuid=F57CF216-DD7E-4C85-861E-34848E961421
*-core:21
description: Motherboard
product: X556UQK
vendor: ASUSTeK COMPUTER INC.
physical id: 0
version: 1.0
serial: N0CV1720MB0103998
slot: MIDDLE
```

```
*-cpu0
description: CPU
product: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
vendor: Intel Corp.
physical id: 11
bus info: cpu0
version: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
serial: To Be Filled By O.E.M.
slot: U3E1
size: 800MHz
capacity: 4005MHz
```