1.

```
sit315-001@sit315-001-VirtualBox:~$ time ./sequential_matrix_multiply
Result matrix (sample):
328350 323400 318450 313500 308550
333300 328250 323200 318150 313100
338250 333100 327950 322800 317650
343200 337950 332700 327450 322200
348150 342800 337450 332100 326750

real    0m0.007s
user    0m0.005s
sys     0m0.002s
```

```
sit315-001@sit315-001-VirtualBox:~$ time mpirun -np 4 --hostfile ~/hostfile ./mpi_matrix_multiply
Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Result matrix (sample):
328350 323400 318450 313500 308550
333300 328250 323200 318150 313100
338250 333100 327950 322800 317650
343200 337950 332700 327450 322200
348150 342800 337450 332100 326750

real    0m0.825s
user    0m0.098s
sys     0m0.141s
```

Time

1. Sequential Version (sequential_matrix_multiply):
real: 0m0.007s
user: 0m0.005s
sys: 0m0.002s

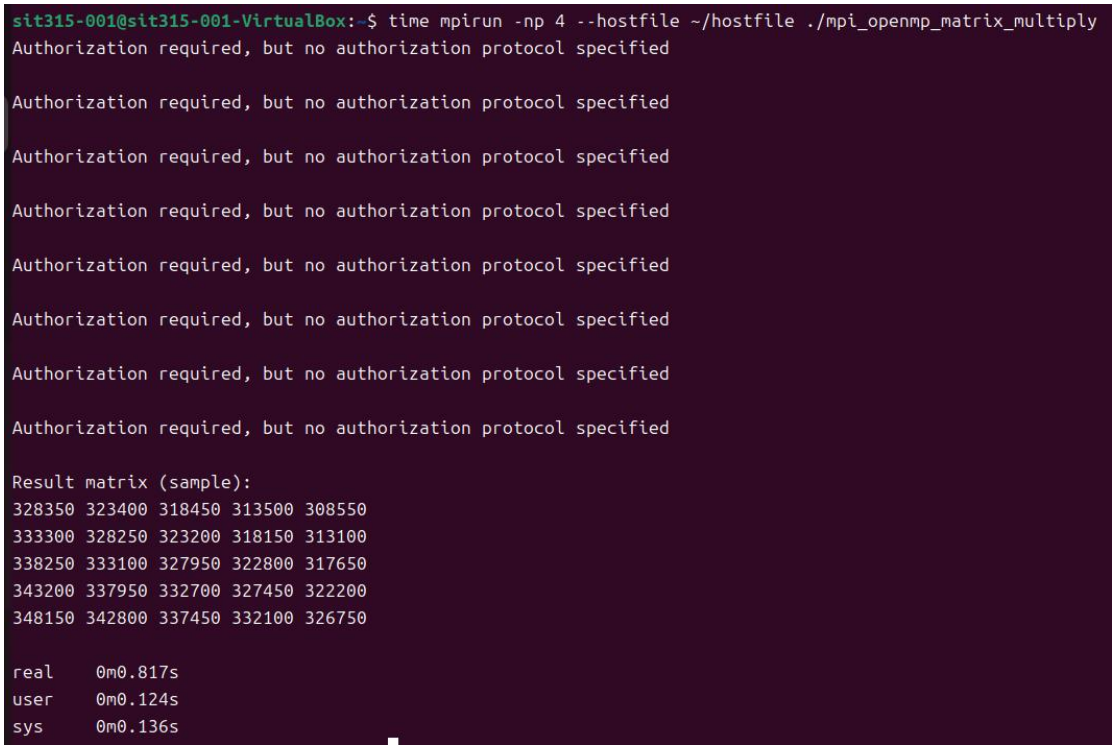
2. MPI Version (mpi_matrix_multiply):
real: 0m0.825s
user: 0m0.098s
sys: 0m0.141s

Conclusion

Sequential Program: The sequential method is fast for small-scale operations due to minimal overhead. However, this method only uses a single CPU core and will not scale efficiently for larger matrices or more complex operations. Its real-time execution is the shortest, but it's limited to small data sizes and lacks scalability.

MPI Programs: MPI introduces parallelization by distributing tasks across multiple processes, but this comes at the cost of inter-process communication overhead. While it makes sense for distributed systems with large data sizes, the observed real-time execution is much higher than the sequential approach due to this overhead.

2.



Time

3. MPI + OpenMP Version (mpi_openmp_matrix_multiply):
real: 0m0.817s
user: 0m0.124s
sys: 0m0.136s

Conclusion

MPI + OpenMP Program: Adds multi-threading but provides minimal additional benefit for small-scale tasks. The overhead from both MPI communication and thread management limits its effectiveness at small sizes, making it no better than MPI alone for smaller problems.

3.

```
sit315-001@sit315-001-VirtualBox:~$ time mpirun -np 4 -hostfile hostfile ./mpi_opencl_matrix_multiply
Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified
Authorization required, but no authorization protocol specified

Authorization required, but no authorization protocol specified


Authorization required, but no authorization protocol specified

Result matrix (sample):
328350 323400 318450 313500 308550
333300 328250 323200 318150 313100
338250 333100 327950 322800 317650
343200 337950 332700 327450 322200
348150 342800 337450 332100 326750

real    0m0.983s
user    0m0.174s
sys     0m0.203s
sit315-001@sit315-001-VirtualBox:~$
```

Time

real: 0m0.983s
user: 0m0.174s
sys: 0m0.203s

Conclusion

The MPI + OpenCL version exhibits a slightly longer execution time compared to the MPI and MPI+OpenMP versions. The extra overhead is likely due to OpenCL kernel initialization, context creation, and data transfer between host and device.

Github Link
https://github.com/Lonely-DM/SIT315/tree/main/M3.T1P

Video Link
https://youtu.be/Jfnb3oZKwIg