Table 3 - Matrix multiplication compute time time for increasing matrix size

(Serial & Hybrid MPI-POSIX/OMP). **Note: You may reduce the size of the matrices to suit your local computing resources**

Number of nodes (Specify 1 if only using your local computer): 1 Number of CPU cores or logical processes per node: 2 Number of MPI processes: 1 Number of threads per MPI process: 2					
Matrix size	500x500	1000x1000	2000x2000	3000x3000	4000x4000
Serial time, Ts (s)	0.49	3.91	33.14	137.18	333.25
Parallel time, Tp - POSIX/OMP (s)	0.02	0.04	47.79	119.41	206.06
MPI communication time (s)	0.00	0.00	0.01	0.02	0.22
Speed Up (Ts/Tp)	24.50	97.75	0.69	1.15	1.62

The theoretical speed up is approximately 1.818.

To calculate the theoretical speedup when using 2 processes compared to the serial version of your matrix multiplication code, we can use Amdahl's Law. This law states that the speedup of a task using multiple processors is limited by the sequential portion of the task.

Amdahl's Law Formula:

S = 1 / ((1 - P) + P / N)

Assume that about 90% of the work can be parallelized, then P = 0.9.

S= 1 / ((1 - 0.9) + 0.9 / 2) ≈ 1.818

The actual speedup for larger matrices is somewhat close to the theoretical speedup predicted by Amdahl's Law but small matrix sizes show unusually high speedup likely due to low overhead. For larger matrices, where overhead and communication start to play a role, the speedup aligns more closely with the theoretical value of 1.818. The speedup for 500x500 and 1000x1000 is much higher than the theoretical value (24.50 and 97.75 respectively). This is likely because the problem size is small and the parallel overhead is negligible. However, the speedup for 2000x2000 is significantly lower than the theoretical value of 1.818. This suggests that for this matrix size, the parallel overhead and MPI communication time start to become significant factors. For 3000x3000 and 4000x4000 matrices, the speed up value starts to decrease to approach the thereotical value showing the benefits of parallelization for larger matrix sizes.

```
fit3143-student@fit3143:~/Desktop$ mpirun -np 2 ./task3
Reading Matrix A and B - Start
Reading Matrix A and B - Done
Writing Matrix C to File - Start
Writing Matrix C to File - Done
Total execution time: 0.017793 seconds
Broadcast time: 0.000542 seconds
Scatter time: 0.000007 seconds
Gather time: 0.000000 seconds
```

```
fit3143-student@fit3143:~/Desktop$ mpirun -np 2 ./task3
Reading Matrix A and B - Start
Reading Matrix A and B - Done
Writing Matrix C to File - Start
Writing Matrix C to File - Done
Total execution time: 0.040442 seconds
Broadcast time: 0.001963 seconds
Scatter time: 0.000001 seconds
Gather time: 0.000000 seconds
```

```
fit3143-student@fit3143:~/Desktop$ mpirun -np 2 ./task3
Reading Matrix A and B - Start
Reading Matrix A and B - Done
Writing Matrix C to File - Start
Writing Matrix C to File - Done
Total execution time: 47.786936 seconds
Broadcast time: 0.010337 seconds
Scatter time: 0.012437 seconds
Gather time: 0.000000 seconds
```

```
fit3143-student@fit3143:~/Desktop$ mpirun -np 2 ./task3
Reading Matrix A and B - Start
Reading Matrix A and B - Done
Writing Matrix C to File - Start
Writing Matrix C to File - Done
Total execution time: 119.410520 seconds
Broadcast time: 0.028025 seconds
Scatter time: 0.026137 seconds
Gather time: 0.000000 seconds
```

```
fit3143-student@fit3143:~/Desktop$ mpirun -np 2 ./task3
Reading Matrix A and B - Start
Reading Matrix A and B - Done
Writing Matrix C to File - Start
Writing Matrix C to File - Done
Total execution time: 206.056898 seconds
Broadcast time: 0.224183 seconds
Scatter time: 0.200954 seconds
Gather time: 0.000000 seconds
```