|  |  |  |
| --- | --- | --- |
| Number of OpenMP threads for each MPI process | Execution time | Explain the results |
| 2 | 1.716534 | Using two threads lightens the workload by parallelizing the process with threads, thus giving a short calculation time |
| 4 | 1.653041 | Using four threads hastens the time required to calculate, as the workload is now spread out even more, where each thread gets to use 1 core. |
| 8 | 1.701626 | When using 8 threads, while it is a greater number of threads, it seems that the calculation being done by different processes and their threads is not equal in calculation complexity, in addition, the number of cores still stays the same, 4, thus the workload is not spread out in an equal manner, in addition threads fight for resource control and it increases the calculation time |
| 16 | 1.663083 | When using 16 threads, it seems that the workload and the allocation of resources still stays in a problematic position, some threads will have to do harder work than others , and threads still fight for resource control ,hence , the amount of time to calculate the required sum stays about the same without a lot of change , meaning that we have reached a certain amount of saturation on the last step with 8 threads , and adding more threads wont actually reduce the calculation time. |