PARALLEL MATRIX MULTIPLICATION

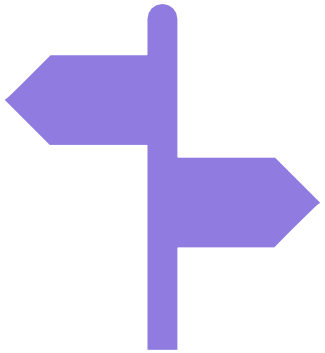
Student name - Nimduli Demin Achchi A D

Student number - 219078029

SEQUENTIAL MATRIX MULTIPLICATION

- Git hub link - https://github.com/NimduliAthukorala/SIT315/blob/main/Module%202/SMM.cpp

ROADMAP TO PARALLELISE



TASK DECOMPOSITION

Create matrix for A and B - Sequence

Print A and B - Sequence

Multiply A and B - Parallel

Print C - Sequence

Write C to text file - Sequence

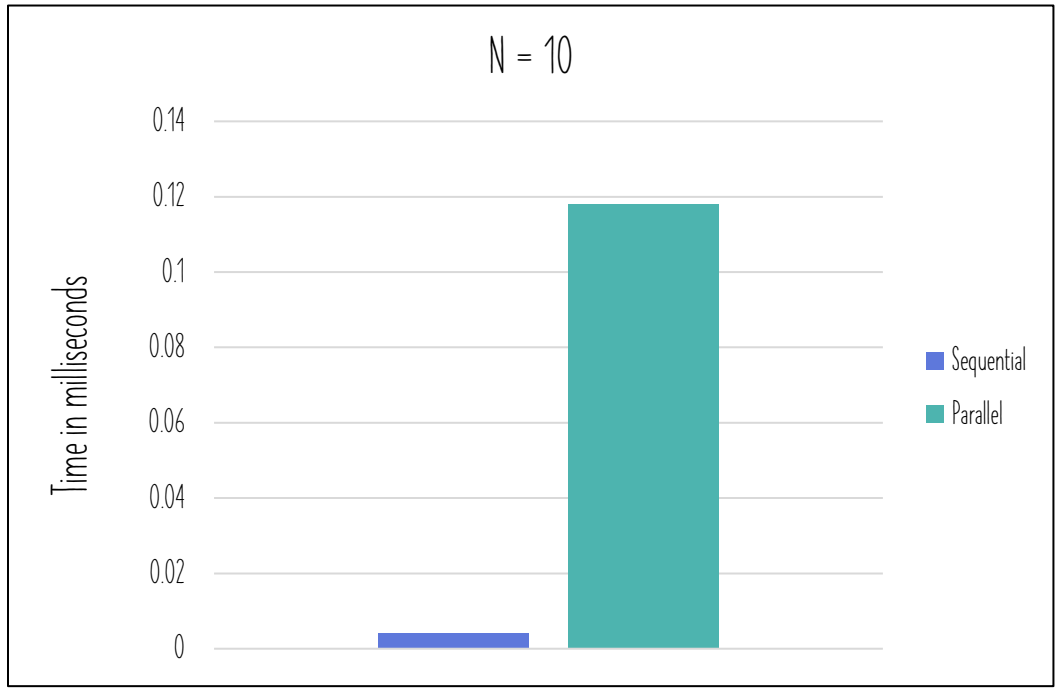
Calculate time - Sequence

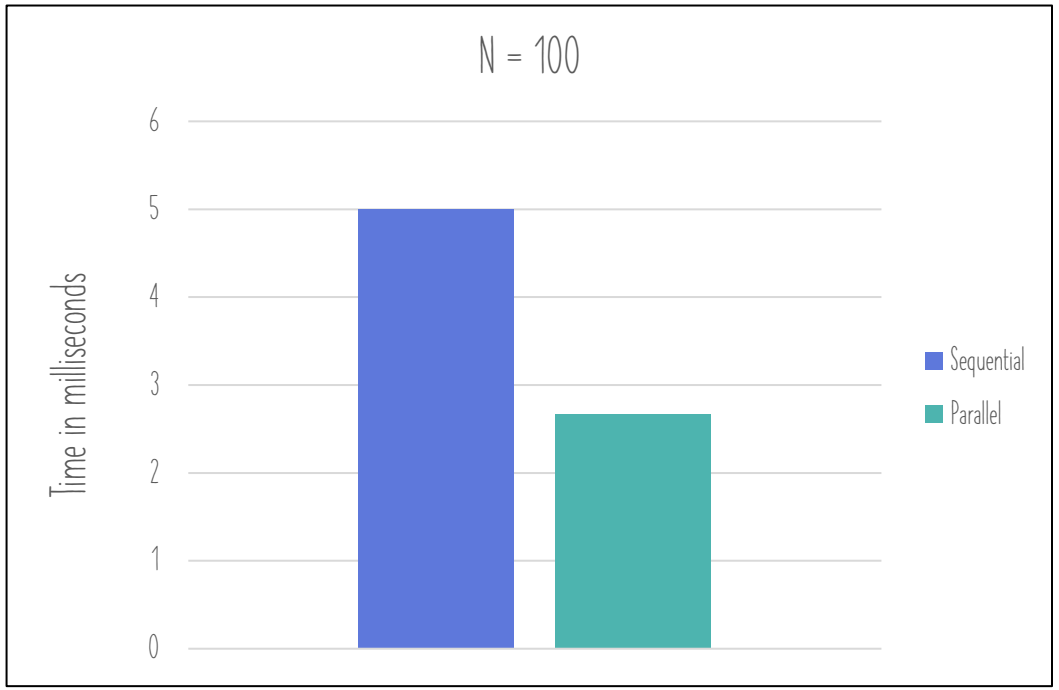
For several of these tasks it is possible to make them parallel for example the Create matrix. However, since we only need to compare multiplication times, this will not be implemented.

PARALLEL MATRIX MULTIPLICATION

- Git hub link - https://github.com/NimduliAthukorala/SIT315/blob/main/Module%202/PMM.cpp

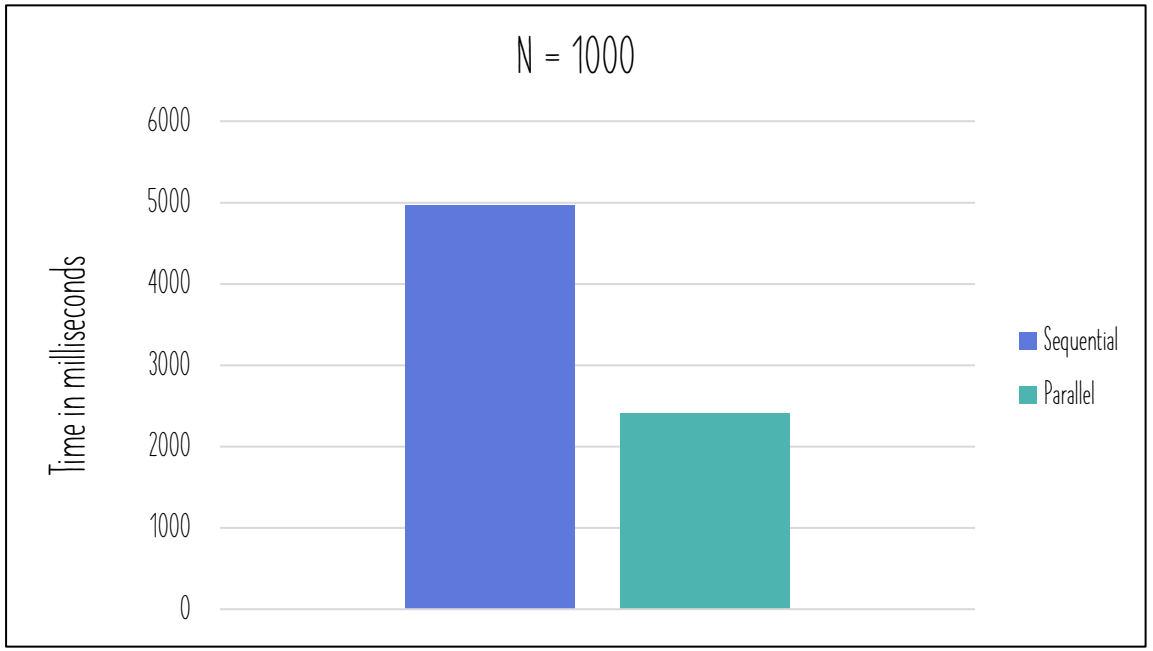
COMPARE THE PERFORMANCE OF SEQUENTIAL AND PARALLEL

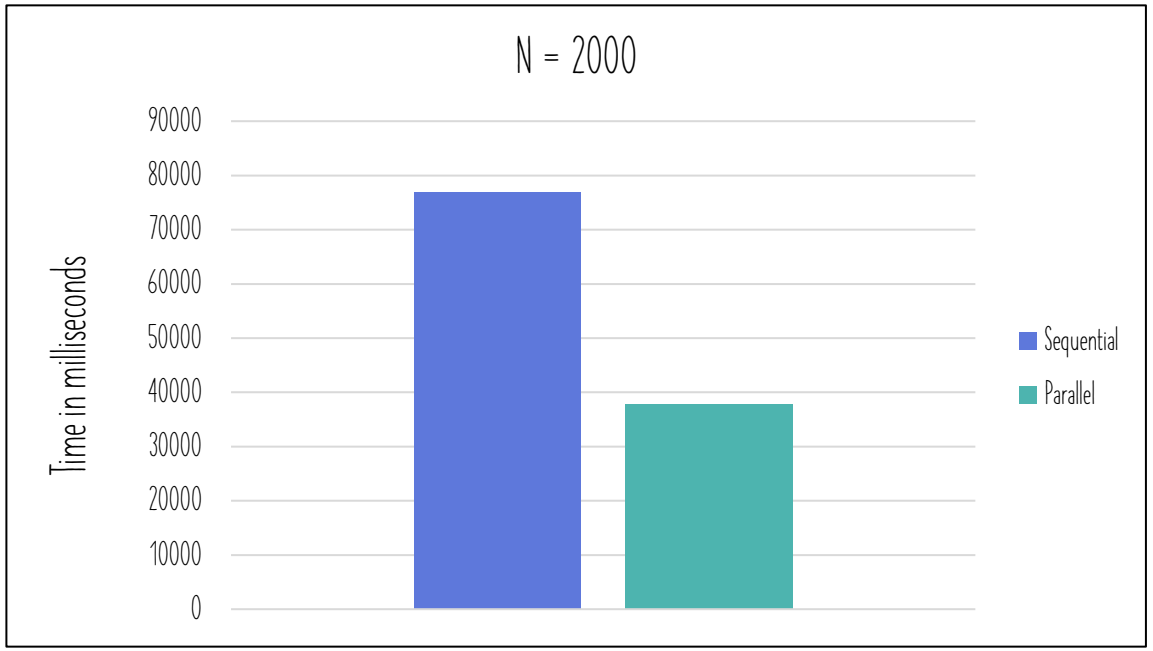




As shown above the two graphs compare the performance of Sequential and Parallel programs for different values of N. Here N is the size of the matrix, with N X N size. As we can see the performance of Sequential is better for very small matrices compared to a larger matrix where Parallel is up to 2 times faster.

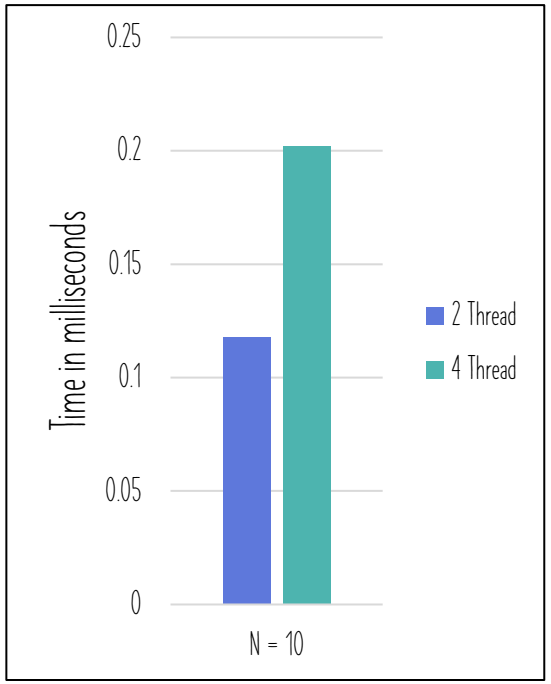
COMPARE THE PERFORMANCE OF SEQUENTIAL AND PARALLEL

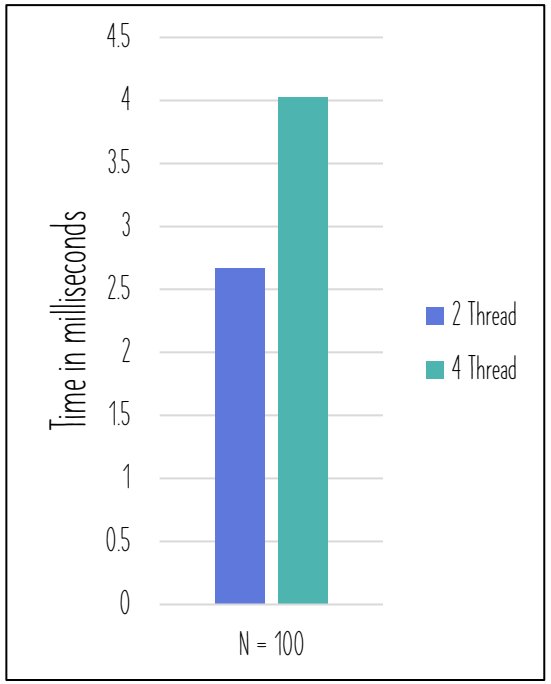


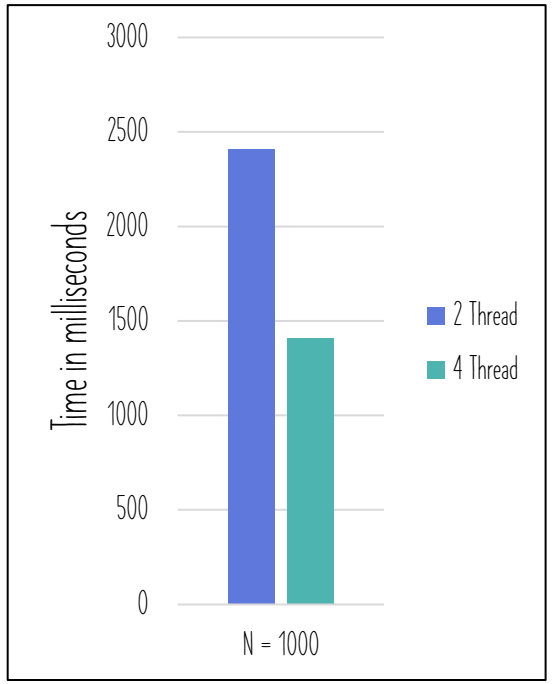


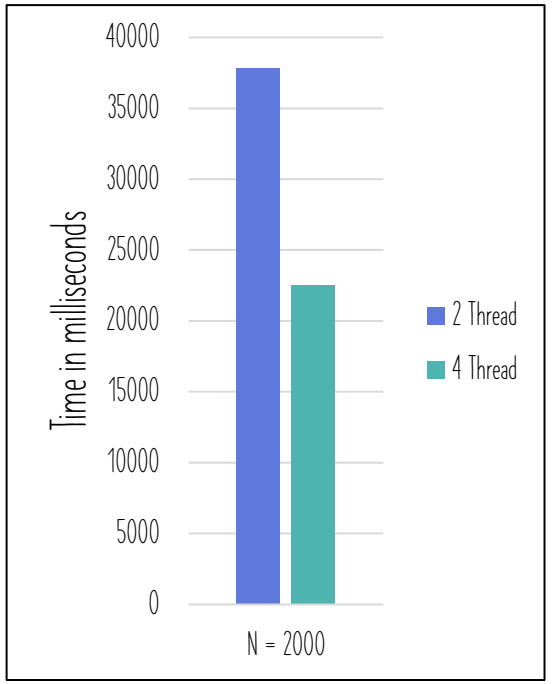
As shown above the two graphs compare the performance of Sequential and Parallel programs for different values of N. Here N is the size of the matrix, with N X N size. For a large matrix Parallel is up to 2 times faster.

COMPARE THE PERFORMANCE FOR DIFFERENT THREADS IN PARALLEL







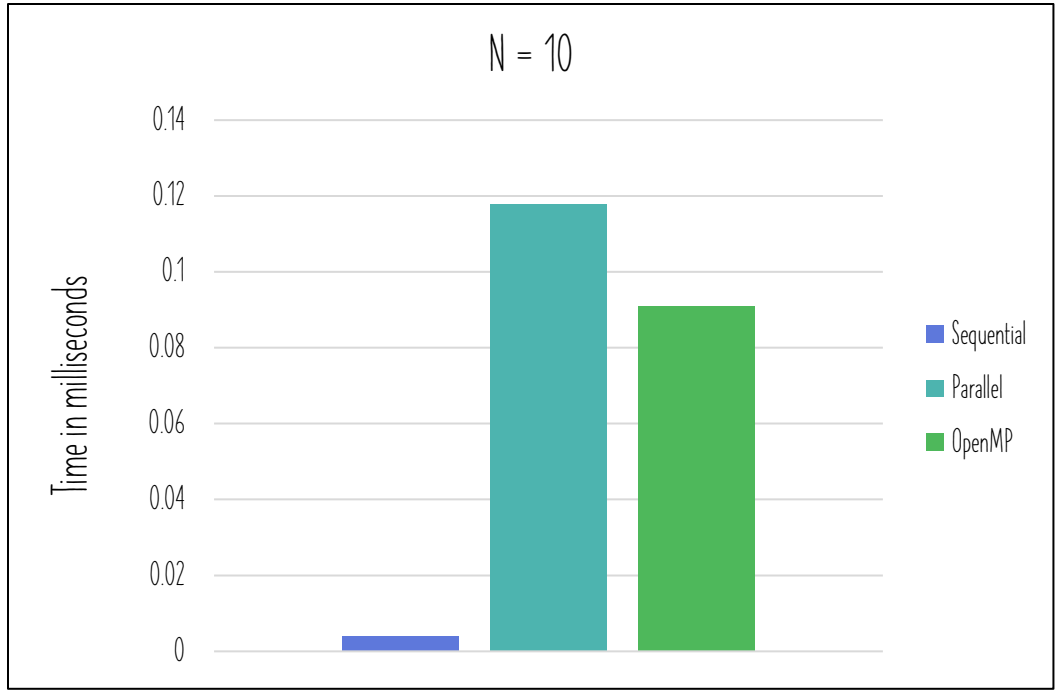


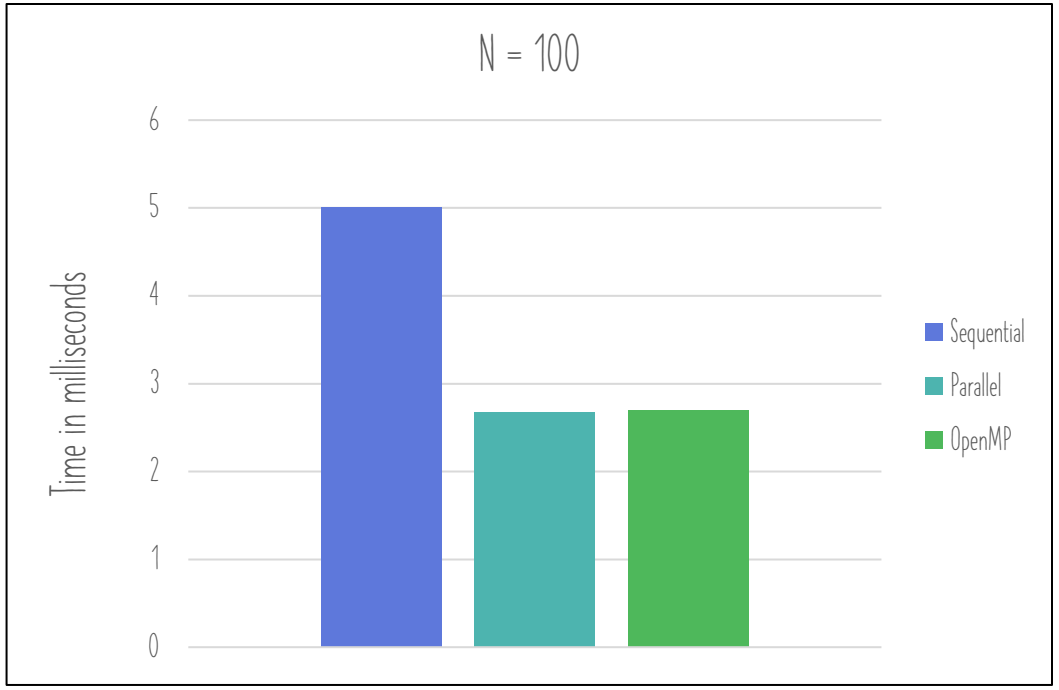
As shown above the graphs compare the performance of Parallel programs for different values of N. Here N is the size of the matrix, with N X N size. This also looks deeper into the performance and thread number. Up till 1000 using 2 threads is better compared to 4. Thus, indicating that it is faster to use 4 threads for large matrices.

OPENMP MATRIX MULTIPLICATION

- Git hub link - https://github.com/NimduliAthukorala/SIT315/blob/main/Module%202/Open.cpp

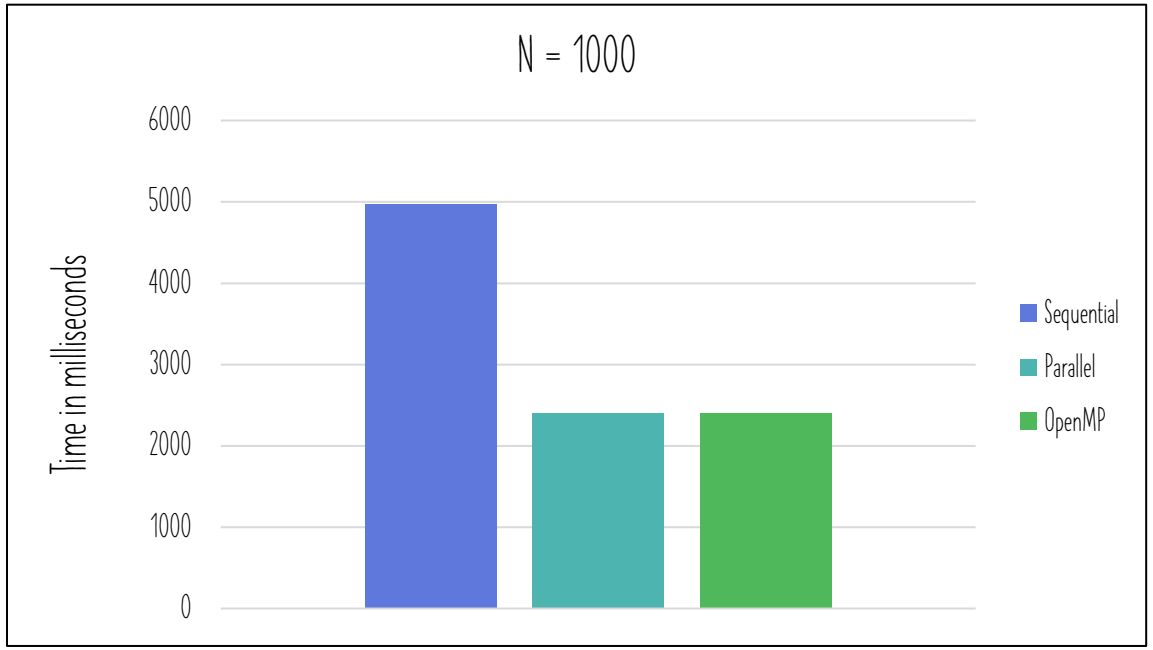
COMPARE THE PERFORMANCE

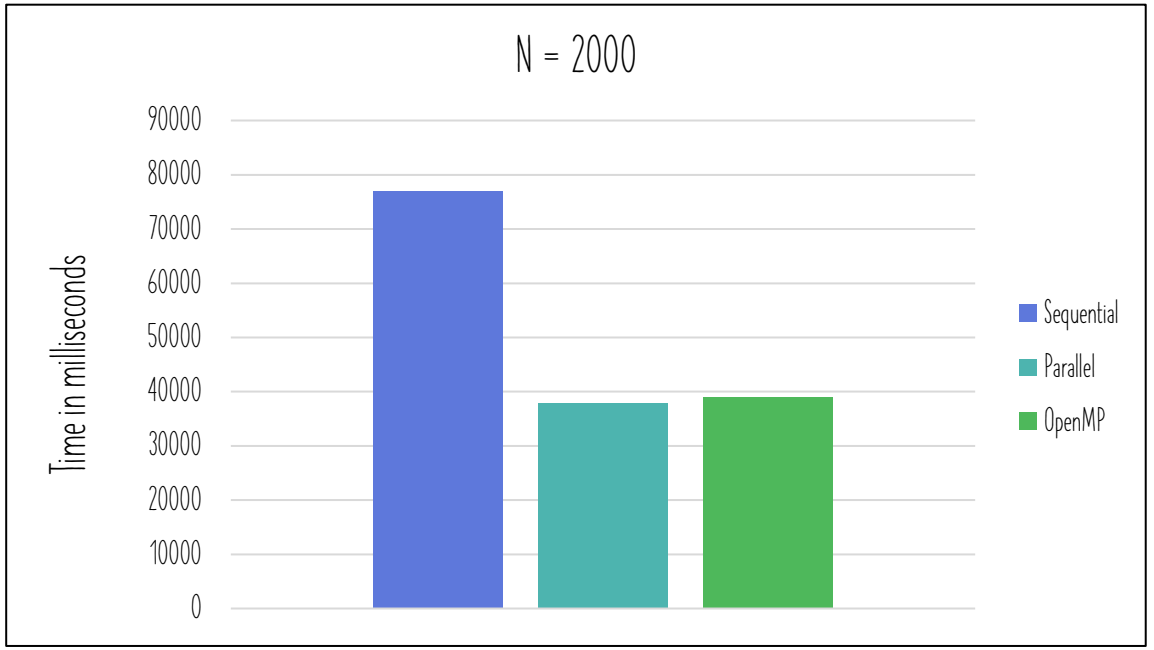




Here we compare Sequential and Parallel to using OpenMP. Similar to before Sequential is better for extremely small matrices.

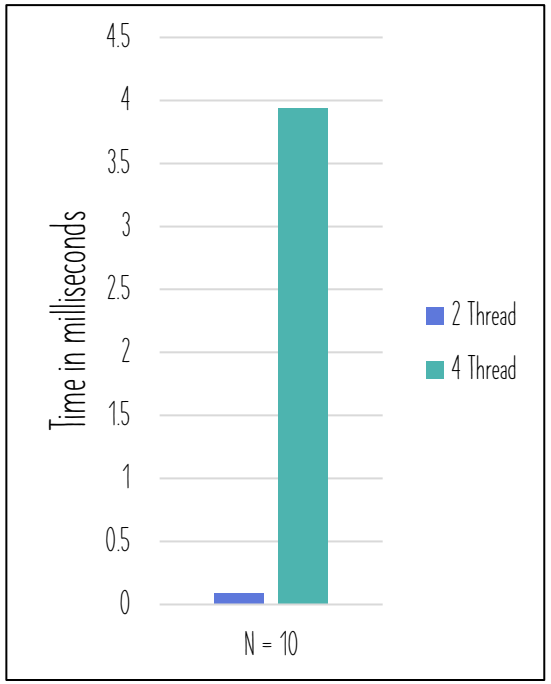
COMPARE THE PERFORMANCE

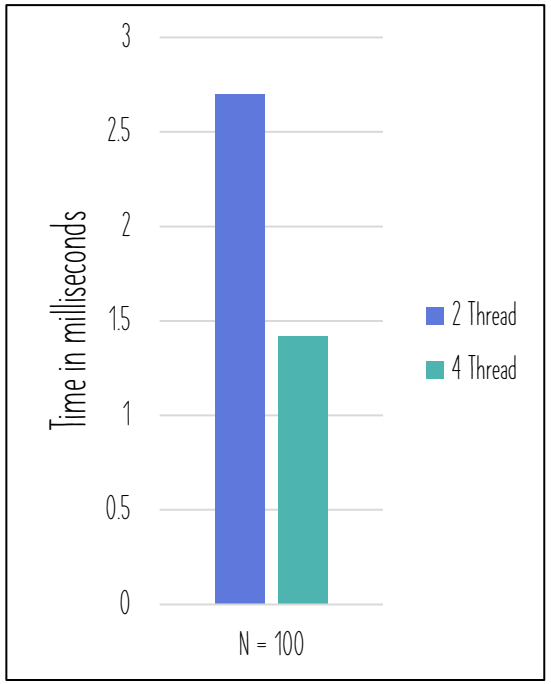


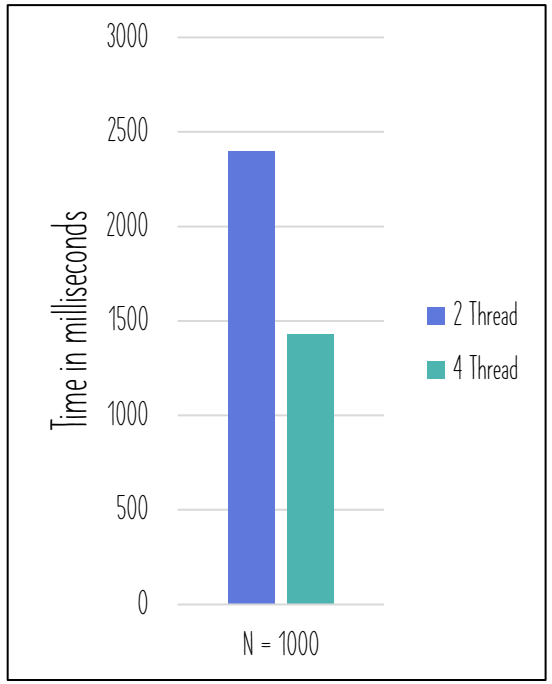


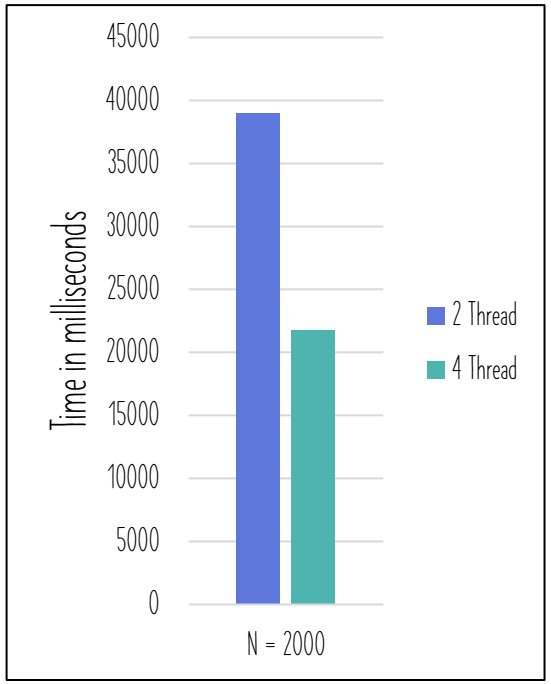
Here we compare Sequential and Parallel to using OpenMP. The performance of the latter two are similar and faster than Sequential.

COMPARE THE PERFORMANCE FOR DIFFERENT THREADS









As shown above the graphs compare the performance of OpenMP programs for different values of N. Here N is the size of the matrix, with NXN size. This also looks deeper into the performance and thread number. Up till 100 using 2 threads is better compared to 4. Thus, indicating that it is faster to use 4 threads for large matrices.