Task 1

Chart, line chart

Description automatically generated

Here is the plot shows the difference between CPU sequential computing (one thread) verse GPU parallel computing (1024 threads) in terms of matrix multiplication. The major difference between CPU and GPU is how the threads work on the calculation. The CPU I wrote only utilize one thread all the time, which means the tasks are organized one by one for all the elements in the matrix. One plot, we can observe that the relationship between matrix size N and elapsed time are linear. As for the GPU, there are multiple threads work simultaneously (1024 threads). Since jobs are executed simultaneously, the time are tremendously saved.

Task 2

Here is the plot shows the difference between CPU sequential computing (one thread) verse GPU parallel computing (1024 threads) in term of stencil calculation. Like task1, GPU saves a lot of time than CPU due to parallelism. CPU has a clear linear trend between the array size and elapsed time while GPU has a flat curve at beginning then grows up. GPU parallelism can accelerate even more if we make use of share memory. In my implementation, the share memory was applied to store the mask array and output arrays. To conclude, for the jobs in task2, GPU is better than CPU.

Chart, line chart

Description automatically generated