# CS3523: OPERATING SYSTEMS 2

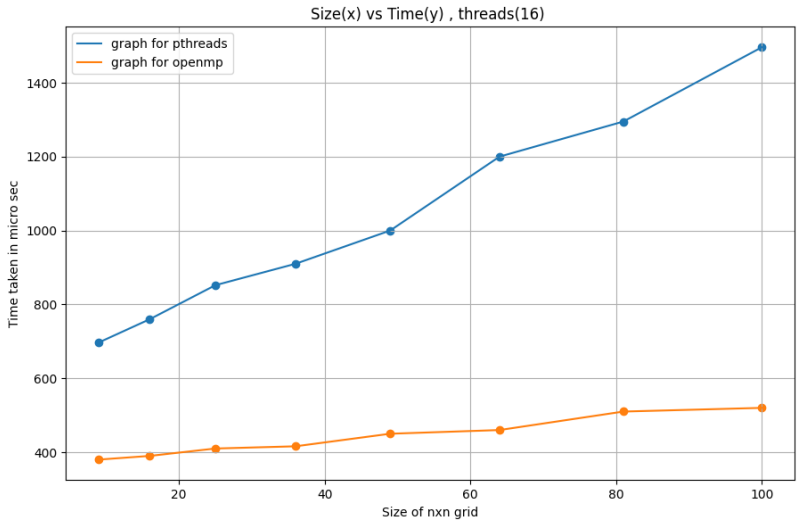
# Programming Assignment 2

-CS21BTECH11011
-B. Asli Nitej Reddy

## Comparing the performance of Algorithm:

### Plot 1 - size of sudoku vs time taken:

Taking Total time taken in Y axis () in microseconds and Size of sudoku (n x n) in X axis ()
Fixing no of threads to 16

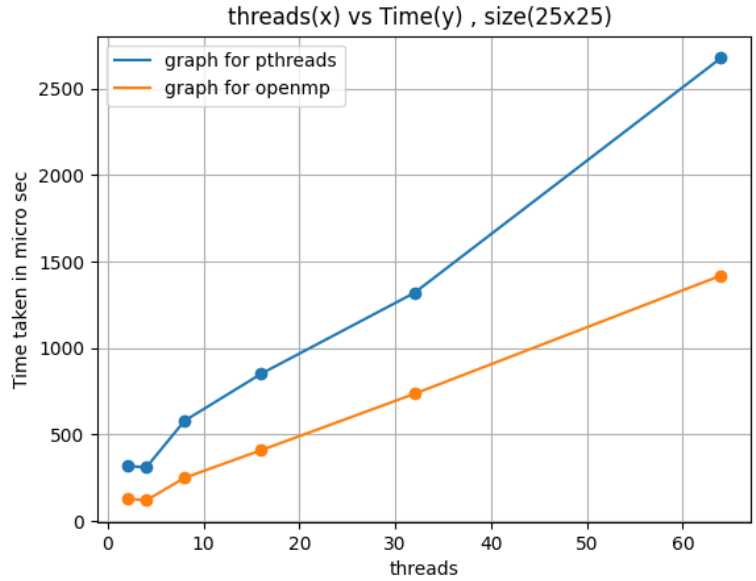



### Observations from plot 1:

- From the above plot we can clearly see that OpenMP is performing better than the Pthread
- In the above plot we can clearly see as the size of sudoku increases the time taken also increases which is expected because of more computation required if sudoku is larger size
- The main reason I think is because OpenMP performs better in the matrix related things as here sudoku is like matrix

**Plot 2 - Number of threads vs time taken:**

Taking Total time taken in Y axis () and Number of threads taken (powers of 2) in X axis ()
Sudoku size is fixed at (25 x 25)




**Observations from plot2:**

- From the above plot we can clearly see that OpenMP is performing better than the Pthreads and the next observation is
- Here what's happening is as threads increases time taken first decreases then again increases drastically but expected is as total threads increases time taken should decrease and explanation is below
- In generally threads work efficiently when we have a large amount of data or a big task to be performed because
- We reduce workload on a core, so total time taken must be reduced
- But if the data is less in our case (it is just 25 x 25 matrix) the time taken for creating the threads and handling that many threads is more complex, so time taken to complete the task is less compared to the handling of that many threads
- This is the main reason why we got plot like above

**What will Output.txt file contains:**

Thread 1 checked column no 1 and is found to be valid
Thread 1 checked column no 1 + total_threads and is found to be valid.
.
.
Thread 2 checked column no 1 and is found to be valid
.
.
Thread 5 checked row no 1 and is found to be valid
.
.
.
Thread j checked grid no m and is found to be valid
.
Thread i checked column no k and is found to be valid invalid
.
.

Sudoku is invalid
Total time taken is 0.000509 in seconds

{
1.thread related details
2.sudoku is valid or invalid
3.total time taken
}

## Conclusion:

For my algorithm openMP is performing better than the Pthread based on the above graphs