

# REPORT

*Sreevatsav B*

*Chaitanya Sreenivas*

*Siddharth Reddy*

*P L Sravanthi*

*Lohit Garge*

**AIM:** To develop a parallel code for the following problem using OpenMP and to report the speedup of your implementations by varying the number of threads from 1 to 16 (i.e., 1, 2, 4, 6, 8, 10, 12, 14, and 16). Use gettimeofday() for calculating runtime and consider the average of 5 runs. Finally, draw appropriate plots using the GNU plot

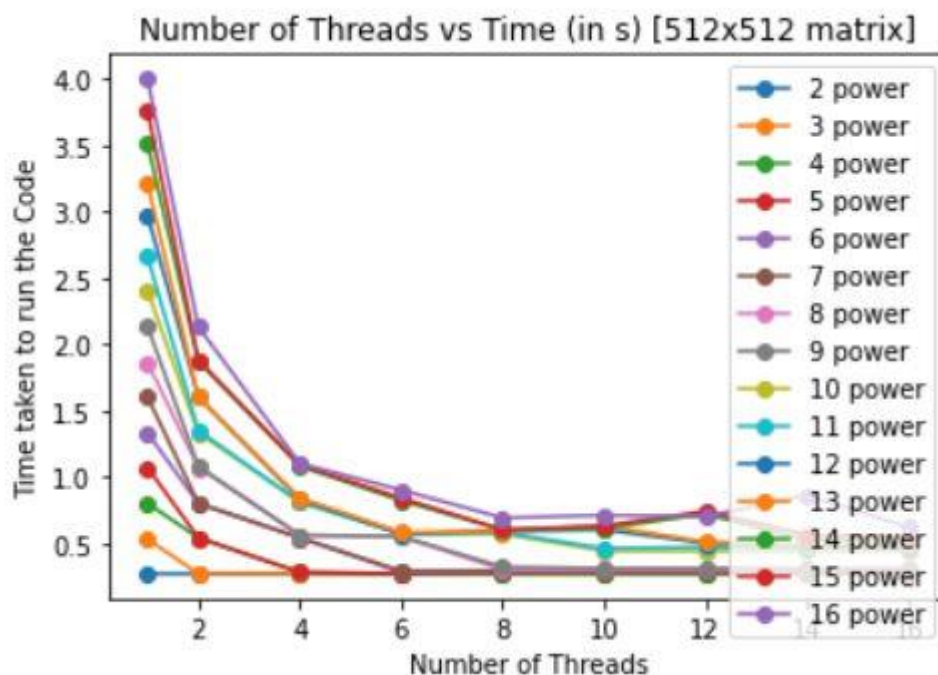
## **OBSERVATIONS:**

For each power, matrix size vs time all graphs are present in the ipynb file.

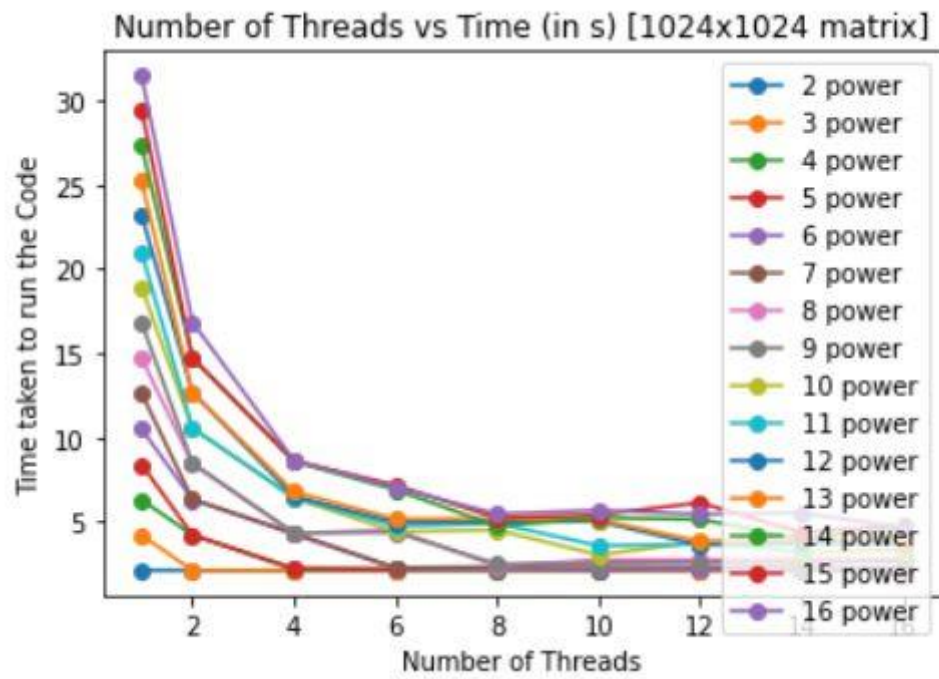
Average of 5 runs were considered.

## **Q3**

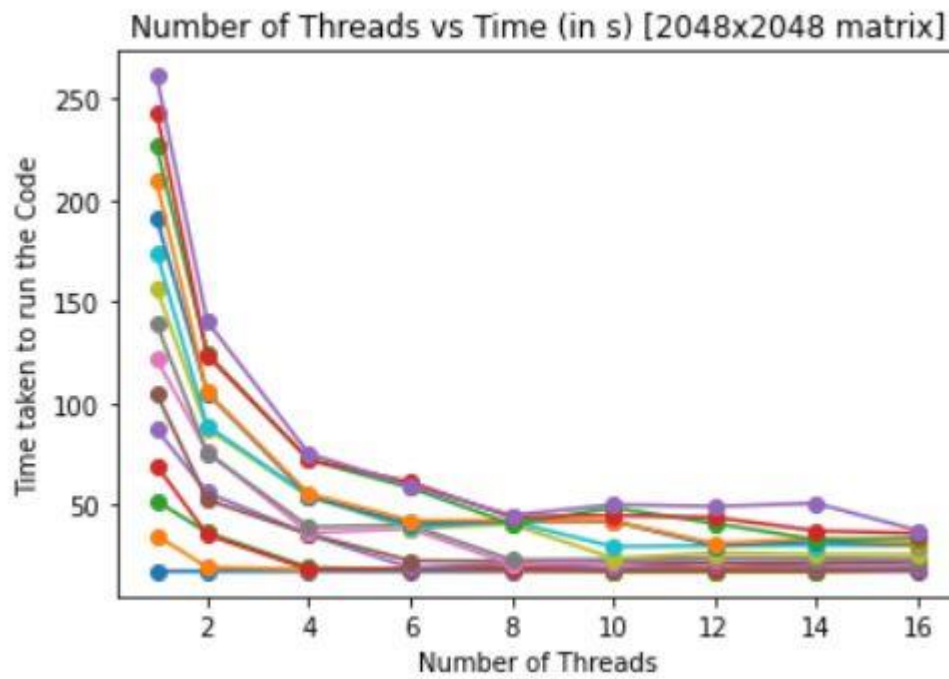
### GRAPH FOR 512\*512 MATRIX SIZE



GRAPH FOR 1024\*1024 MATRIX SIZE

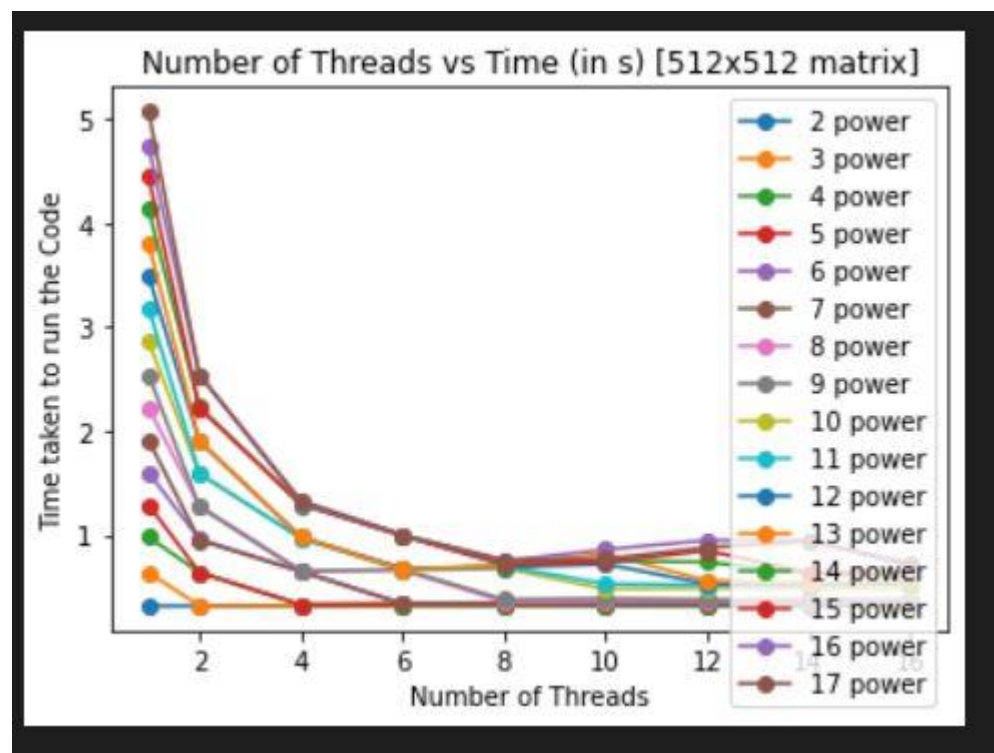


GRAPH FOR 2048\*2048 MATRIX SIZE

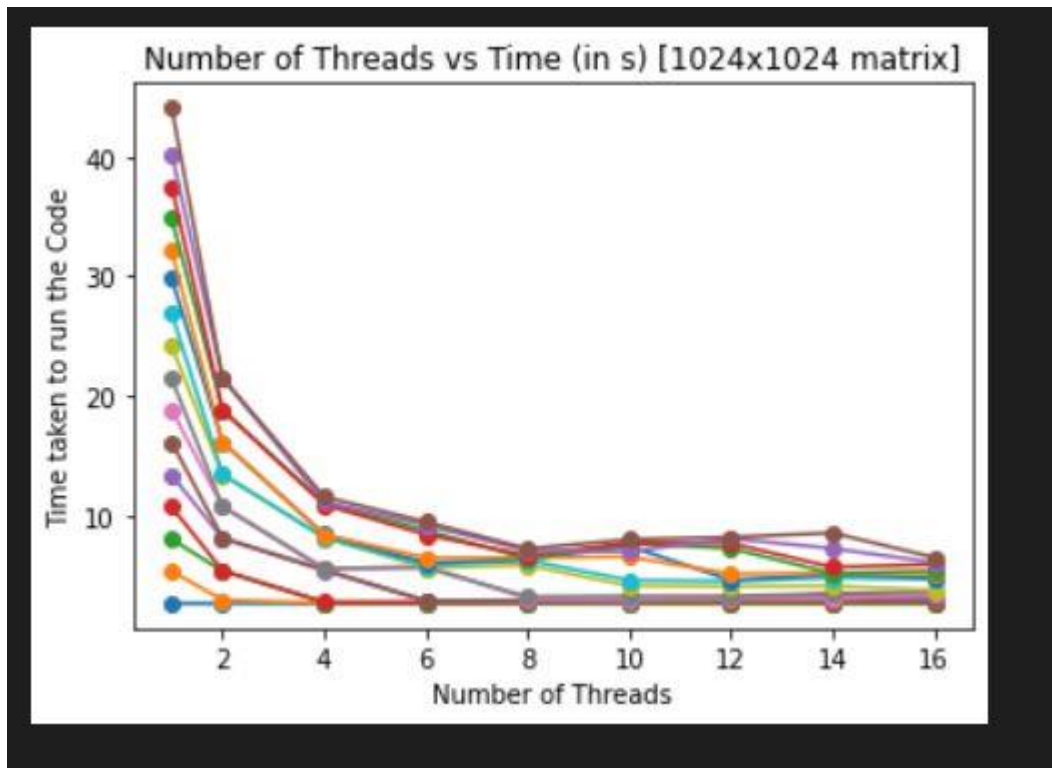


Q1

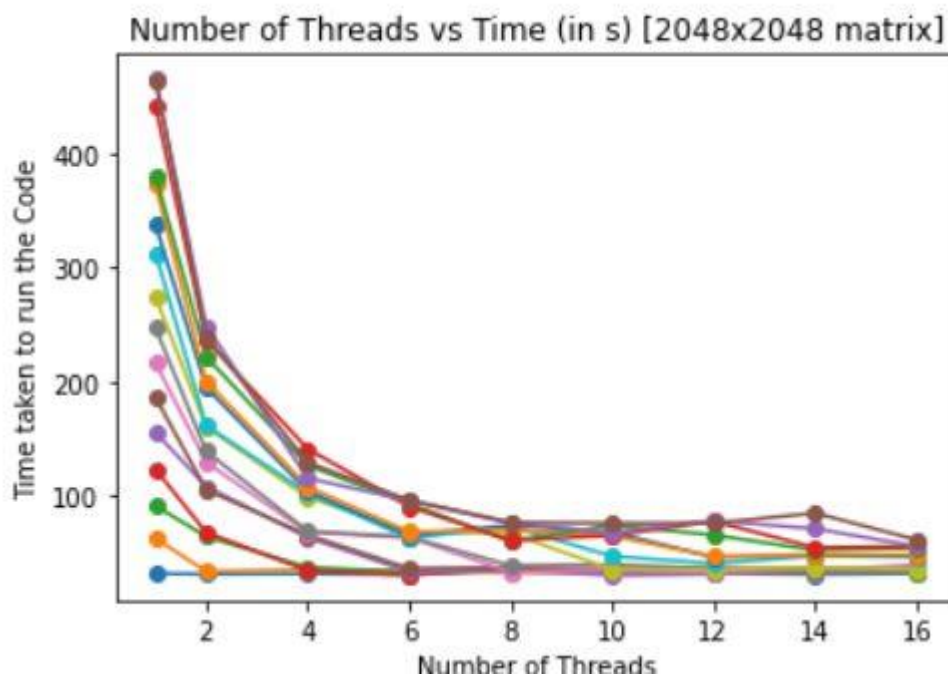
GRAPH FOR 512\*512 MATRIX SIZE



### GRAPH FOR 1024\*1024 MATRIX SIZE



### GRAPH FOR 2048\*2048 MATRIX SIZE



**Q2 and Q4, graphs are plotted in the ipynb file**

**INFERENCE:**

Through this lab we have inferred the following:

- The run time decreases as no. of threads increase.
- The runtime increases as power increases
- As block size increase, time decreases