A REPORT ON

OS Assignment - 2

BITS Pilani
Hyderabad Campus

Under Supervision of

Dr Dipanjan Chakraborty

Course Code: CS F372

Course Title: Operating Systems

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI HYDERABAD CAMPUS

(November 2022)

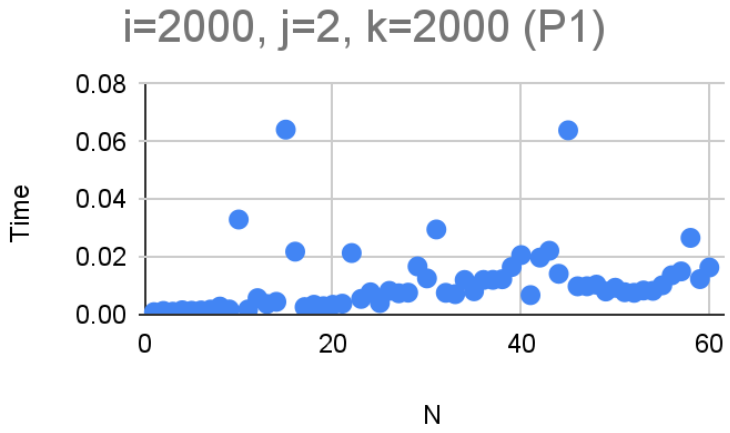
Members (Group 26)

- Dhruv Saxena - 2019B4A71369H (f20191369@hyderabad.bits-pilani.ac.in)
- Hitesh Garg - 2019B3A70466H (f20190466@hyderabad.bits-pilani.ac.in)
- Nikhil - 2019B5A71079H (f20191079@hyderabad.bits-pilani.ac.in)
- Shashwat Anand - 2019B3A70718H (f20190718@hyderabad.bits-pilani.ac.in)
- Vineeth Kumar - 2019B3A70220H (f20190220@hyderabad.bits-pilani.ac.in)

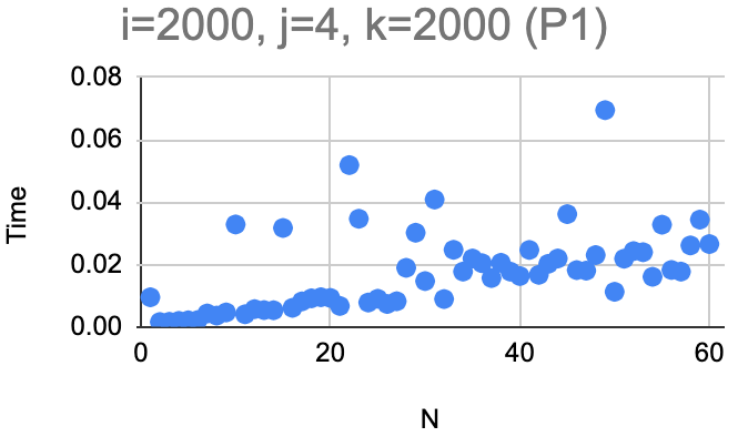
Q1. Time vs No. of threads for P1 (reading the input files) for different input sizes.

The input size varies from (i = 2000, j = 2, k = 2000) to (i = 2000, j = 2000, k = 2000), i.e the matrix size ranges from 2000x2 (or 2x2000) to 2000x2000.

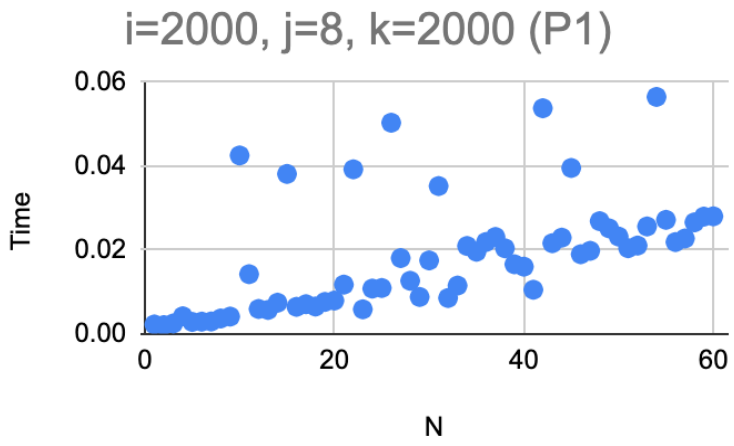
The following are the time taken by the respective no. of threads to read from both the input files.



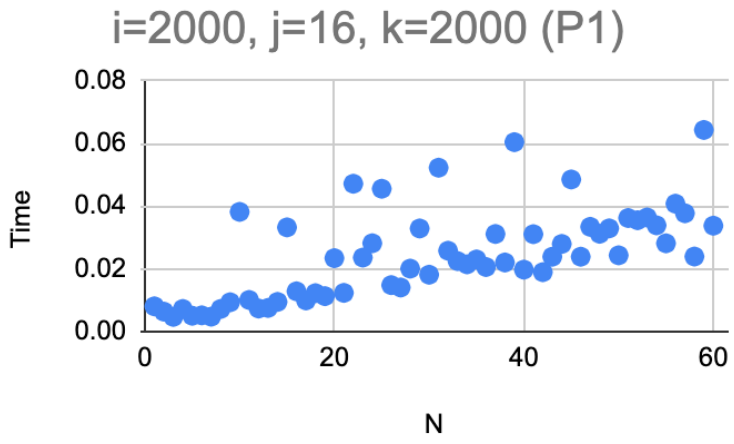
Time vs N for i=2000, j=2, k=2000 for P1



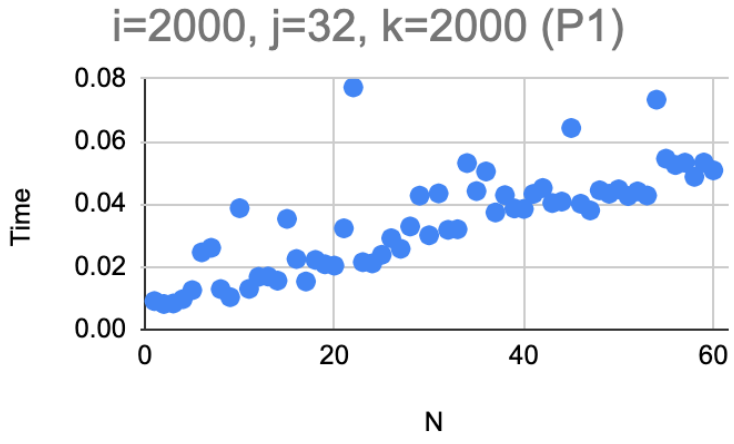
Time vs N for i=2000, j=4, k=2000 for P1



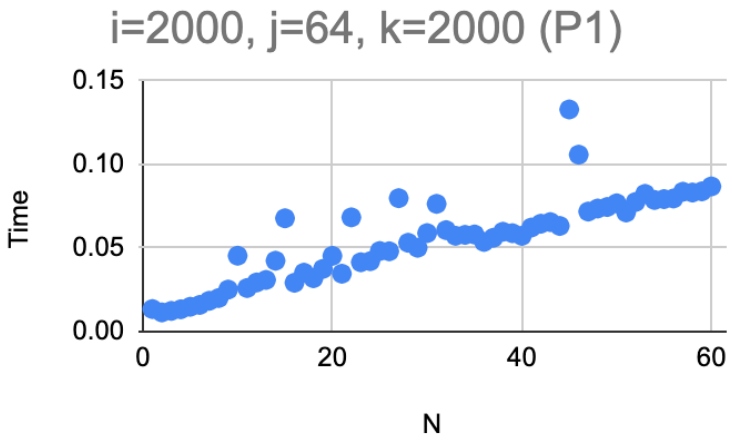
Time vs N for i=2000, j=8, k=2000 for P1



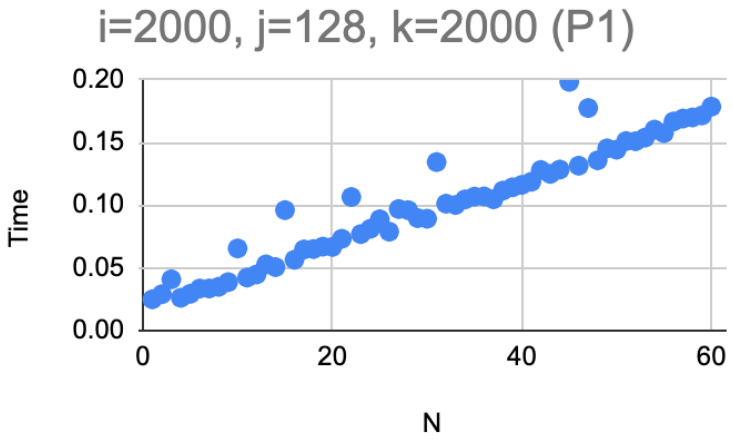
Time vs N for i=2000, j=16, k=2000 for P1



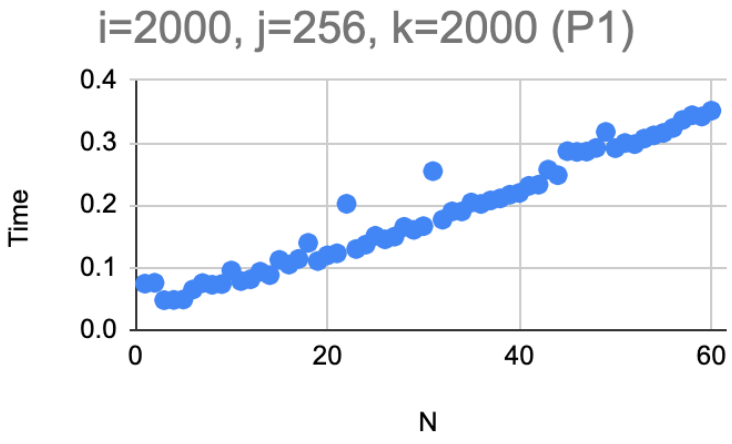
Time vs N for i=2000, j=32, k=2000 for P1



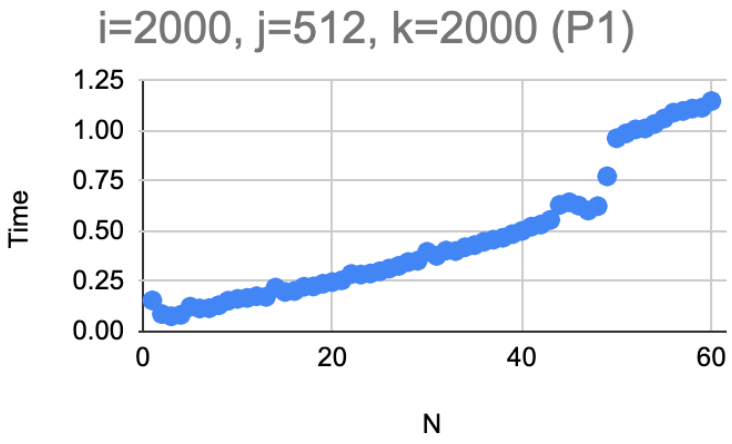
Time vs N for i = 2000, j=64, k=2000 for P1



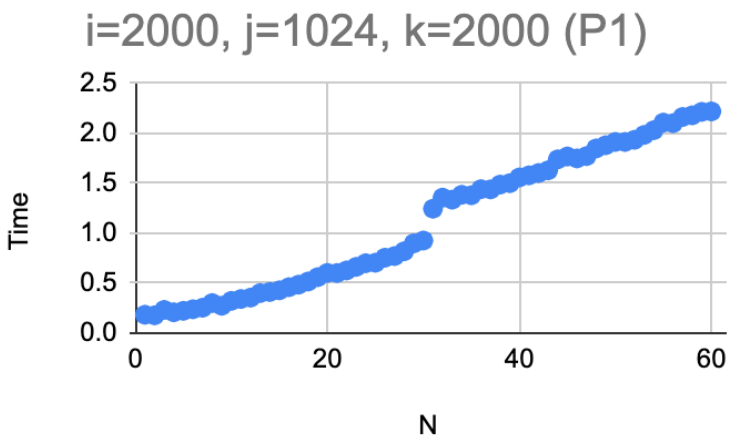
Time vs N for i=2000, j = 128, k=2000 for P1



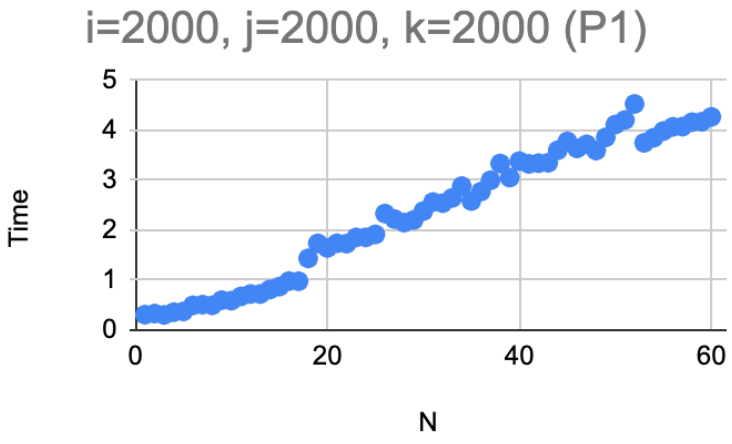
Time vs N for i=2000, j = 256, k=2000 for P1



Time vs N for i=2000, j = 512, k=2000 for P1



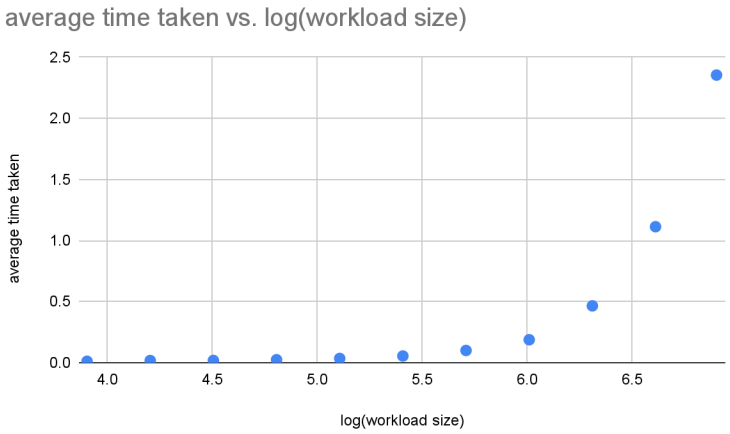
Time vs N for i=2000, j = 1024, k=2000 for P1



Time vs N for i=2000, j = 2000, k=2000 for P1

By observing the graphs the following can be inferred:

1. As the workload size increases, the time taken increases - which can be observed by looking at the values of average time taken vs the log(workload size).

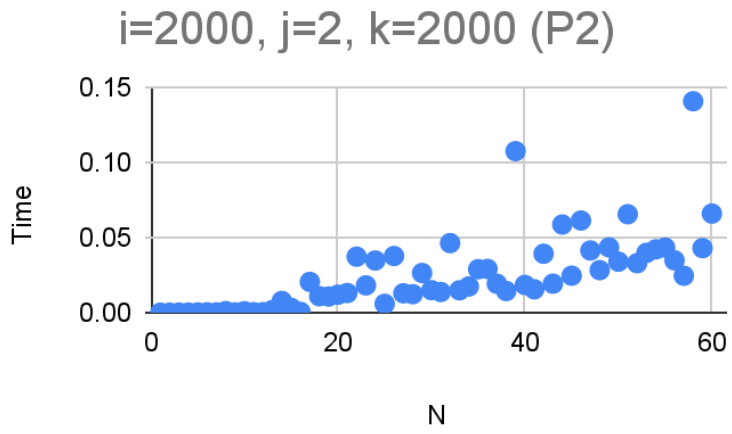


2. For a fixed workload size, as the No. of threads increases, the time taken first decreases but then increases. This can be explained by the fact that initially, more threads were optimising the reading process and hence the reading time fell, however as the no.of threads grow, the overhead time, and waiting time due to spawning more threads, killing and thread joining were larger than the time they were improving in reading, hence the reading time started rising.

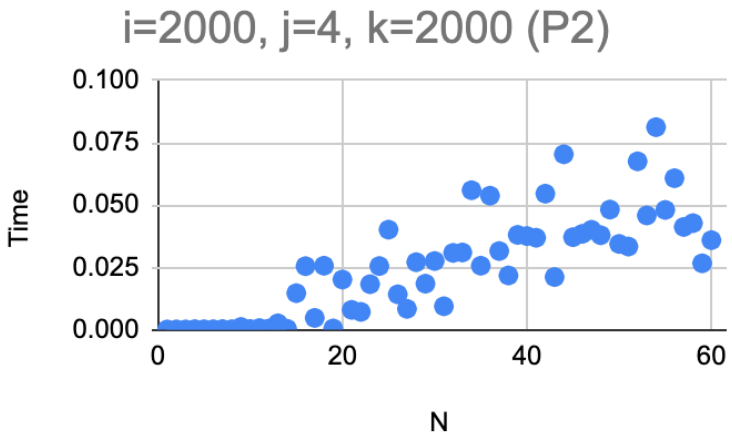
Q2. Time vs No. of threads for P2 (multiplying the matrices) for different input sizes.

The input size varies from i = 2000, j = 2, k= 2000 to i = 2000, j = 2000, k = 2000, i.e the matrix size ranges from 2000x2 (or 2x2000) to 2000x2000 matrix.

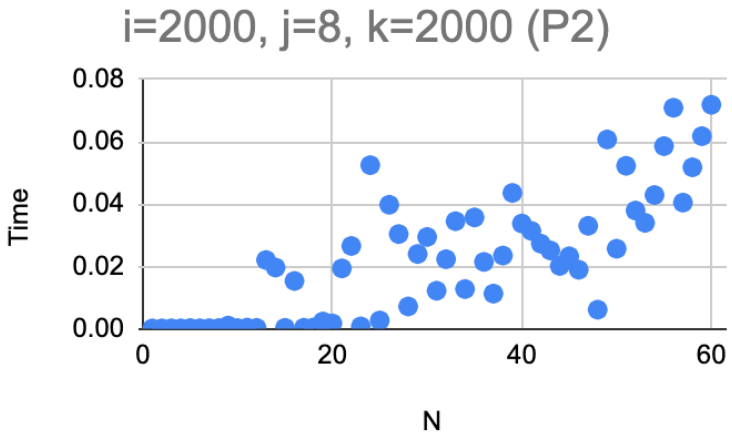
The following are the time taken by the respective no. of threads to multiply the matrices.



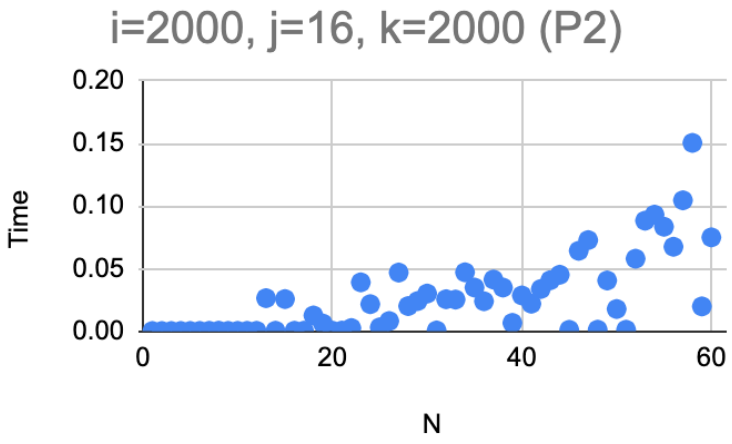
Time vs N for i=2000, j = 2, k=2000 for P2



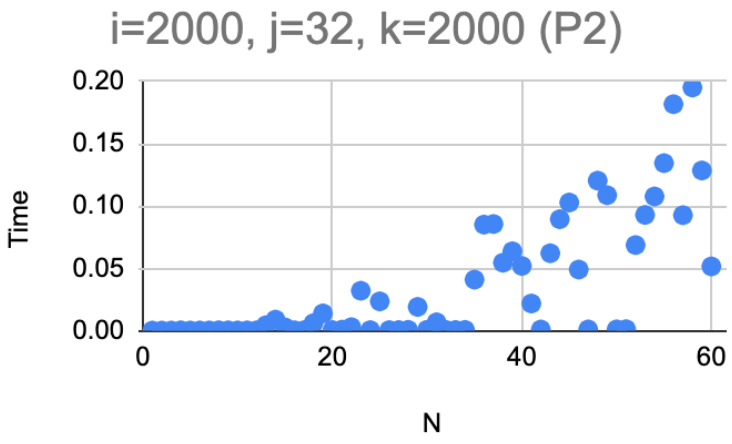
Time vs N for i=2000, j = 4, k=2000 for P2



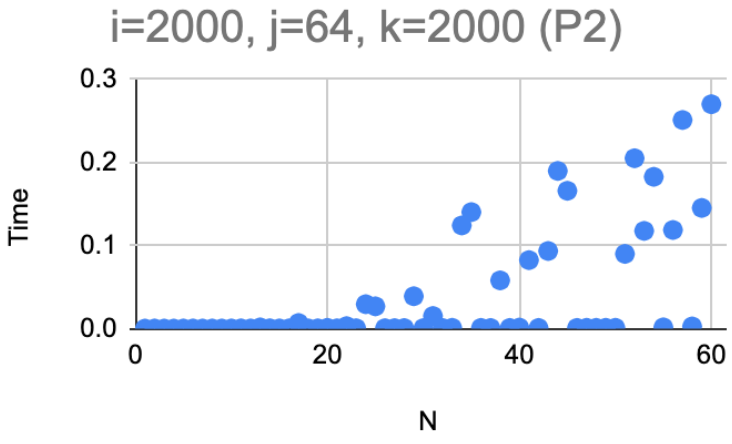
Time vs N for i=2000, j=8, k=2000 for P2



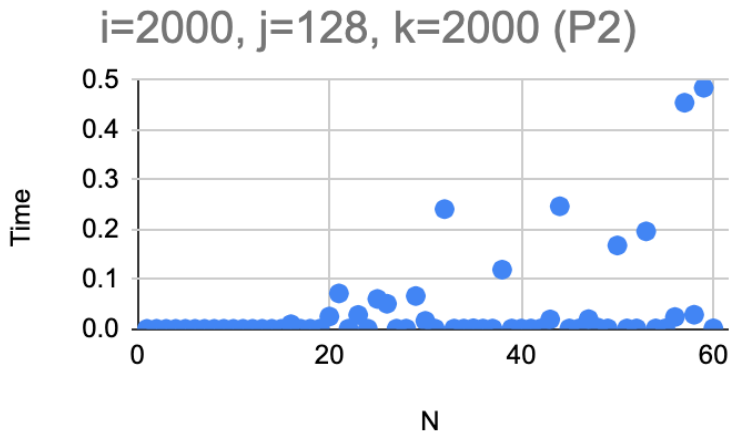
Time vs N for i=2000, j=16, k=2000 for P2



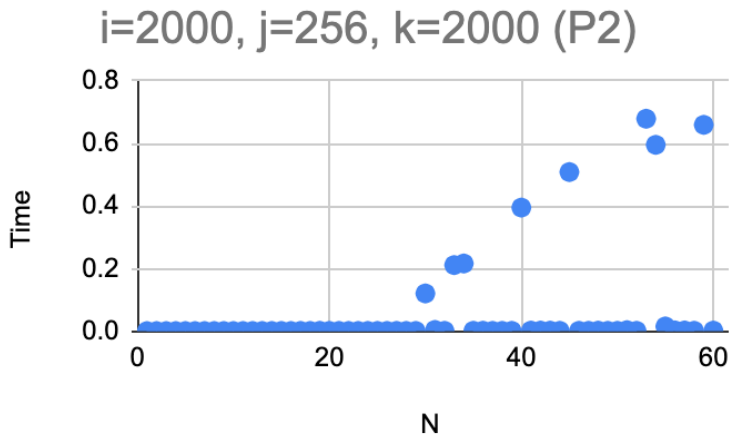
Time vs N for i=2000, j=32, k=2000 for P2



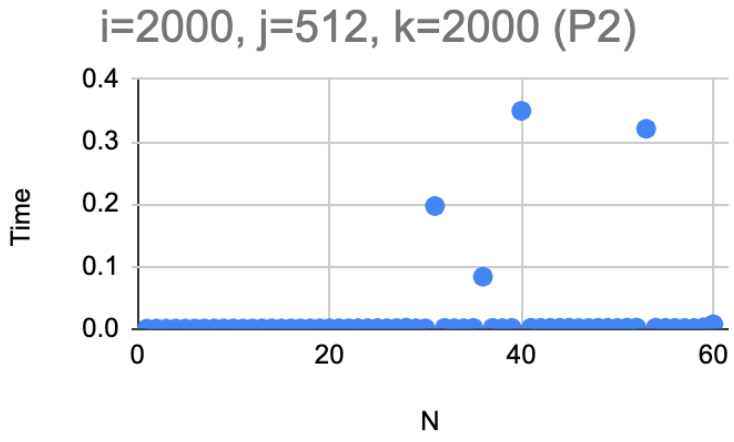
Time vs N for i=2000, j=64, k=2000 for P2



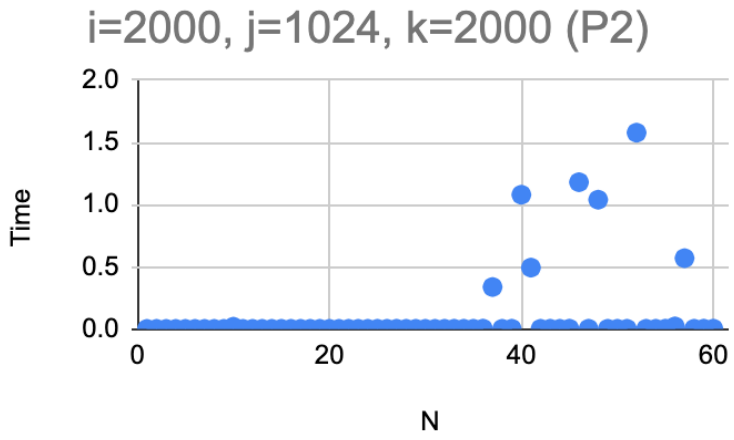
Time vs N for i=2000, j=128, k=2000 for P2



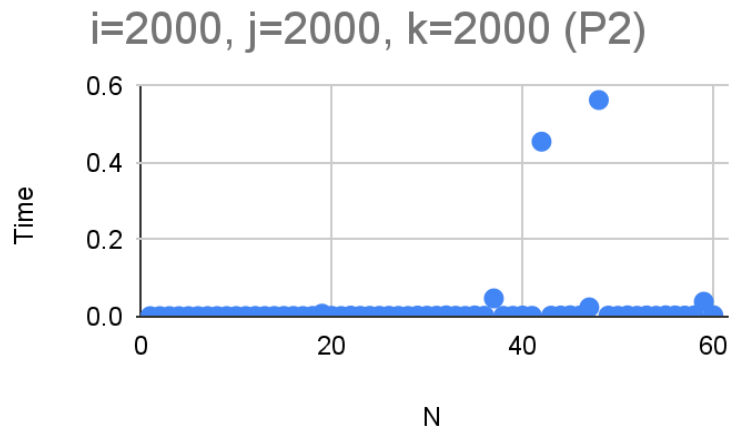
Time vs N for i=2000, j=256, k=2000 for P2



Time vs N for i=2000, j=512, k=2000 for P2



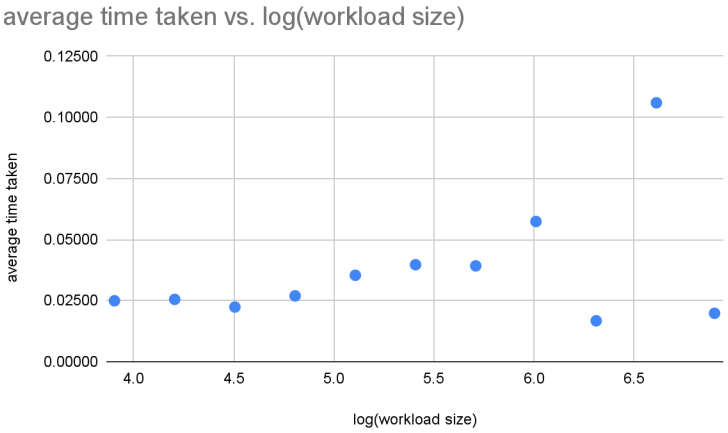
Time vs N for i=2000, j=1024, k=2000 for P2



Time vs N for i=2000, j=2000, k=2000 for P2

By observing the graphs the following can be inferred:

1. As the workload size increases, the time taken increases - which can be observed by looking at the average values of Time taken for different workload sizes.



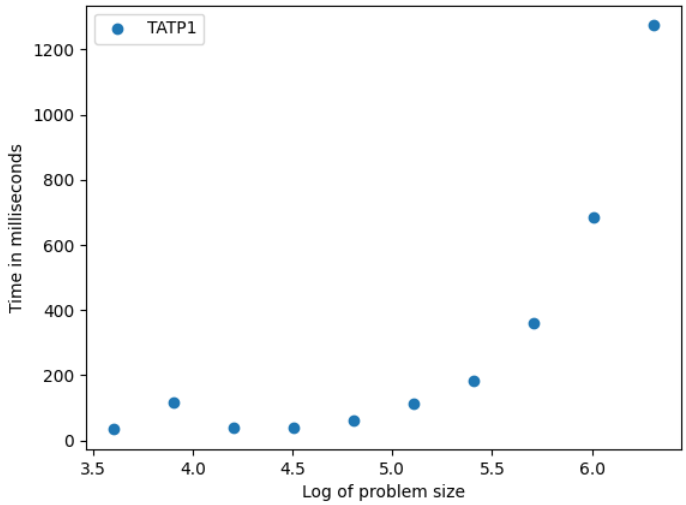
2. For a fixed workload size, 2 patterns are observed:
    a. For small workload sizes, as the no. of threads increases, the time taken to multiply increases due to high thread spawning, kill and thread joining overheads.
    b. For large workload sizes, the time taken becomes approximately constant irrespective of the number of threads because, with large workload sizes, the number of ALU operations per thread decreases but due to the overhead time, the total time becomes constant asymptotically.

Q3.

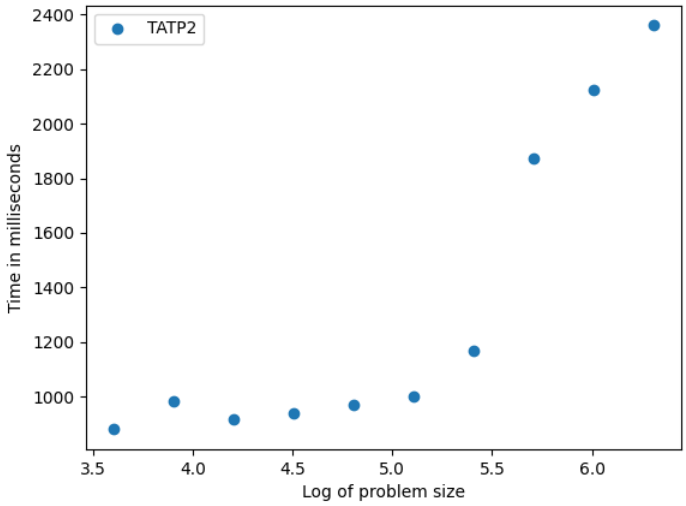
i) Round robin with quantum = 2ms

The total number of elements (workload size) = i*j + j*k, and for the maximum size of the 2000x2000 matrix, multiplied by 2000x2000 results in $8x10^6$ elements, hence it's better to take a log scale of the workload size.

The following graph shows the Turnaround Time v/s Log(workload size) for the scheduling process with a time quantum of 2ms for P1 and P2 respectively.



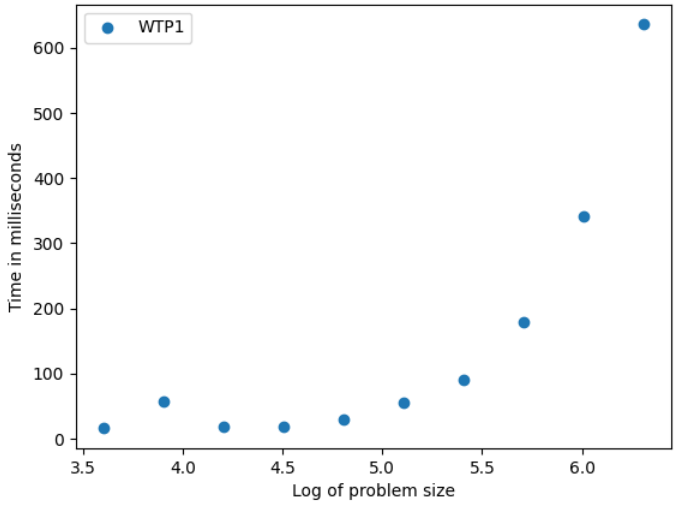
Turnaround time for P1



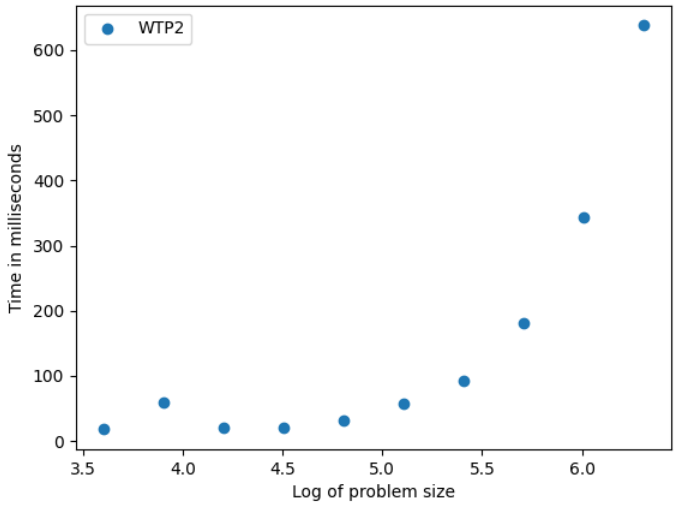
Turnaround time for P2

Total turnaround time for the scheduling can be calculated by adding the turnaround times of P1 and P2.

The following graph shows the Waiting Time v/s Log(workload size) for the scheduling process with a time quantum of 2ms for P1 and P2 respectively.



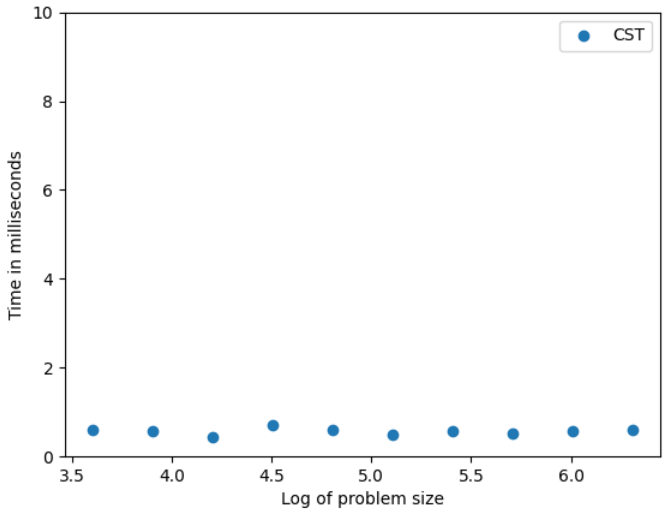
Waiting time for P1



Waiting time for P2

Total waiting time for the scheduling can be calculated by adding the turnaround times of P1 and P2.

The following graph shows the Switching Overhead Time v/s Log(workload size) for the scheduling process with a time quantum of 2ms.

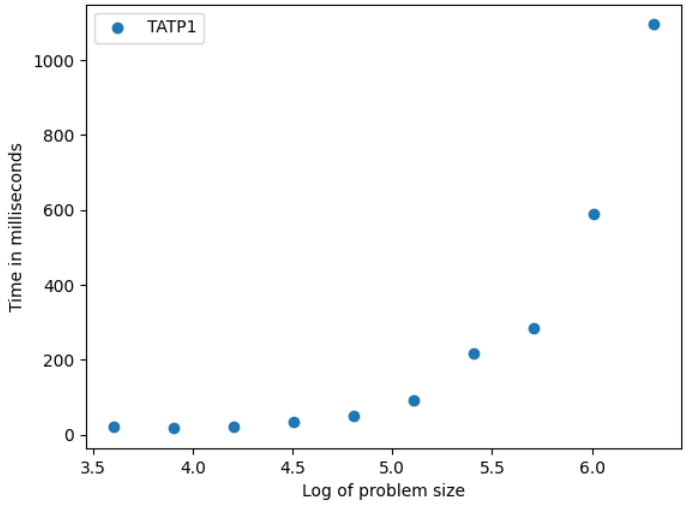


Switching overhead time for the scheduling process

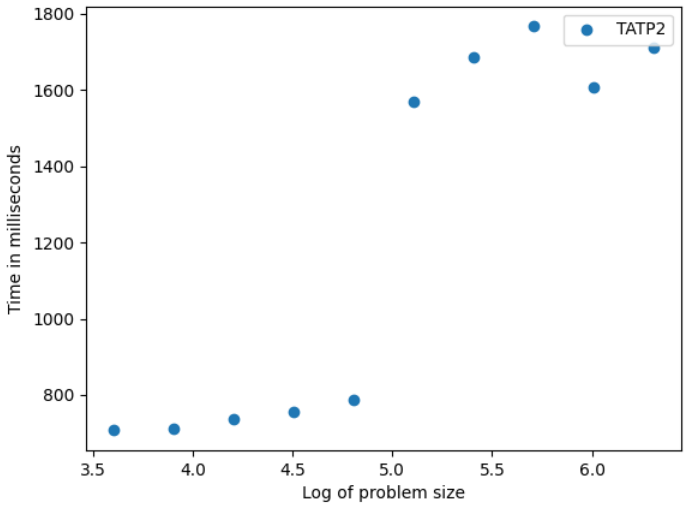
ii) Round robin with quantum = 1ms

Along similar lines, log(workload size) is taken for round robin scheduling process having a time quantum of 1ms.

The following graph shows the Turnaround Time v/s Log(workload size) for the scheduling process with a time quantum of 1ms for P1 and P2 respectively.



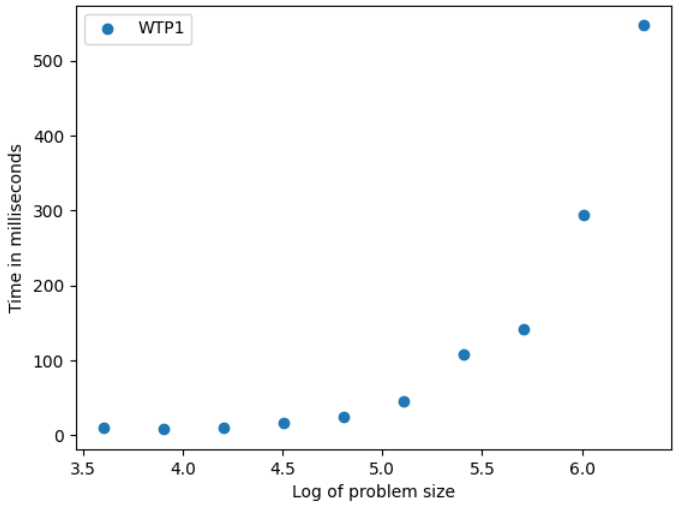
Turnaround time for P1



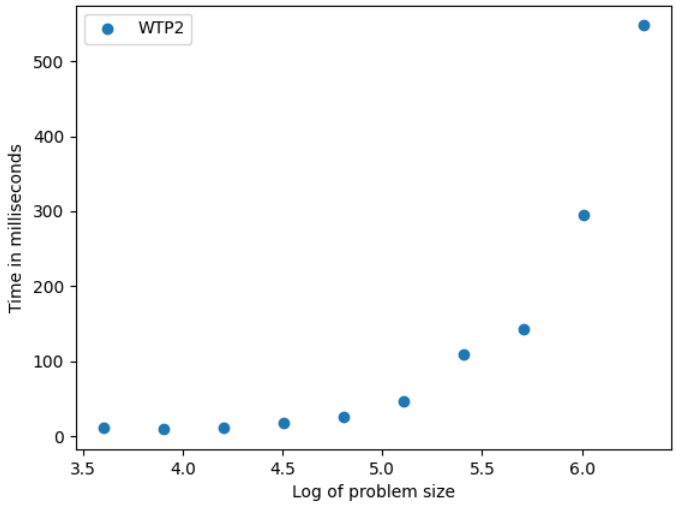
Turnaround time for P2

The total turnaround time for the scheduling can be calculated by adding the turnaround times of P1 and P2.

The following graph shows the Waiting Time v/s Log(workload size) for the scheduling process with a time quantum of 1ms for P1 and P2 respectively.



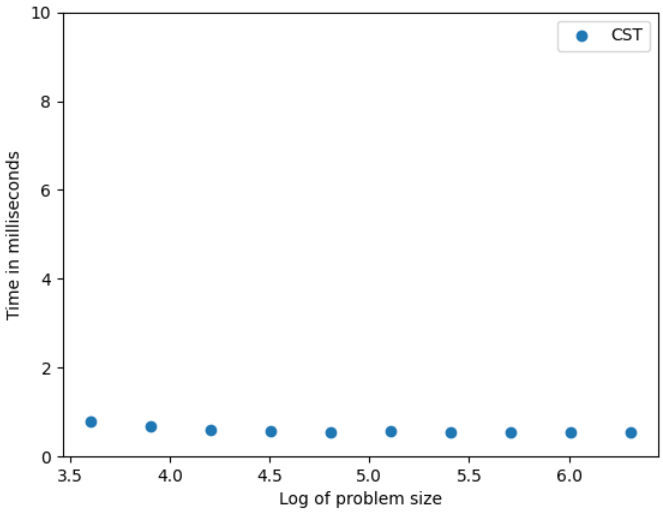
Waiting time for P1



Waiting time for P2

The total waiting time for the scheduling can be calculated by adding the turnaround times of P1 and P2.

The following graph shows the Switching Overhead Time v/s Log(workload size) for the scheduling process with a time quantum of 1ms.



Switching overhead time for the scheduling process

By observing the graphs of turnaround times of P1 and P2 for the two scheduling processes, it can be seen that total turnaround time is higher in the case of scheduling with time quantum = 2ms.
Similarly, the waiting time is also higher in the case of scheduling with time quantum = 2ms.