CS F372

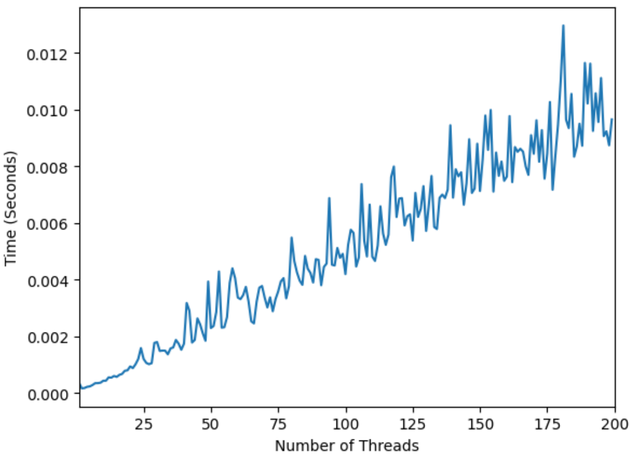
Operating Systems : Assignment-2

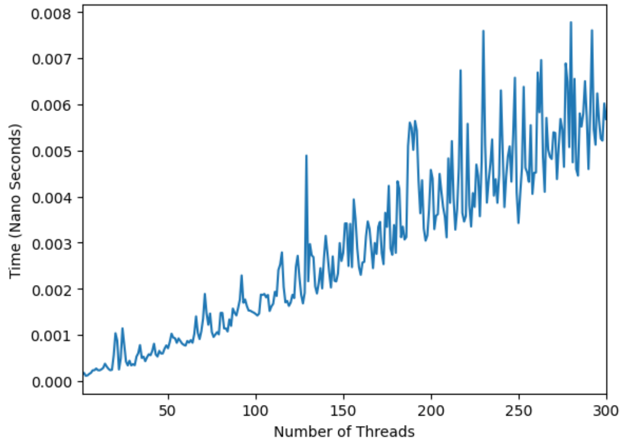
T.Praneeth Bhargav	2020A7PS1299H
M.Bhargav	2020A7PS0025H
S.V.S.Rahul	2020A7PS0204H
D.V.Sasanka	2020A7PS0005H
Tumu Akshar	2020A7PS0003H
Gattu V Hridik Krishna	2020A7PS0102H

Observations and Plottings

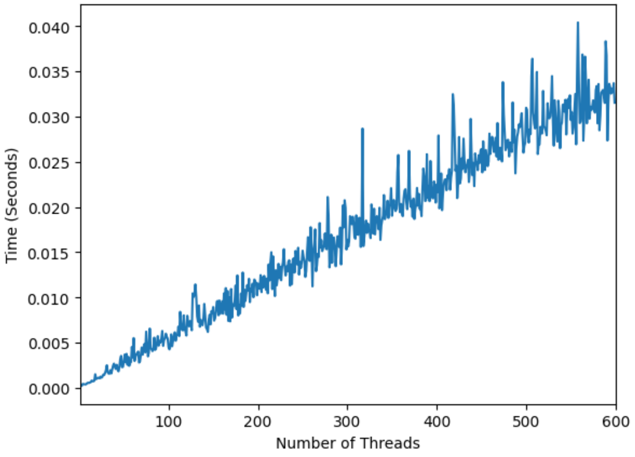
b)iii)

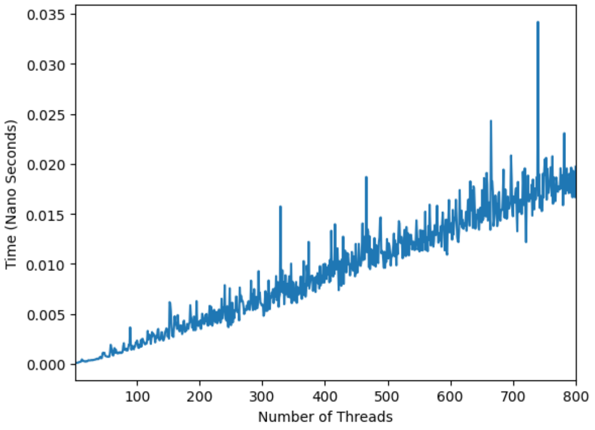
Plots of P1 and P2 for workload of 10x20 and 20x30 matrices.



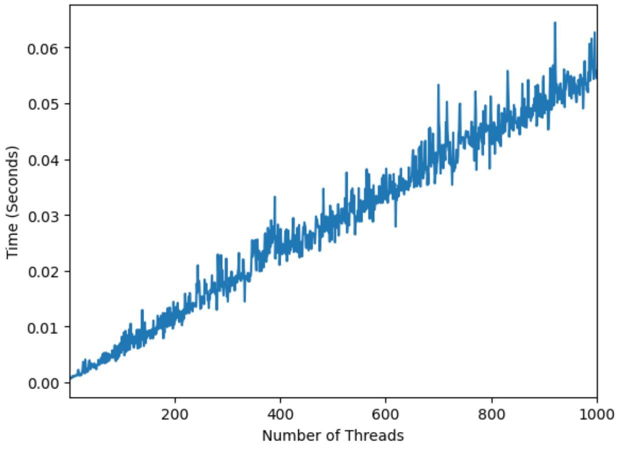


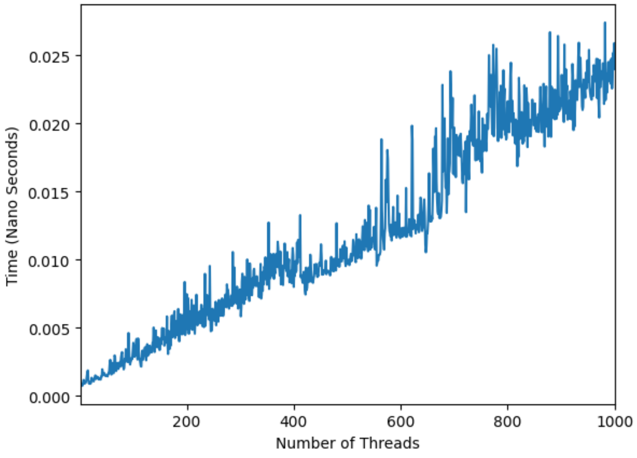
Plots of P1 and P2 for workload of 20x30 and 30x40 matrices.



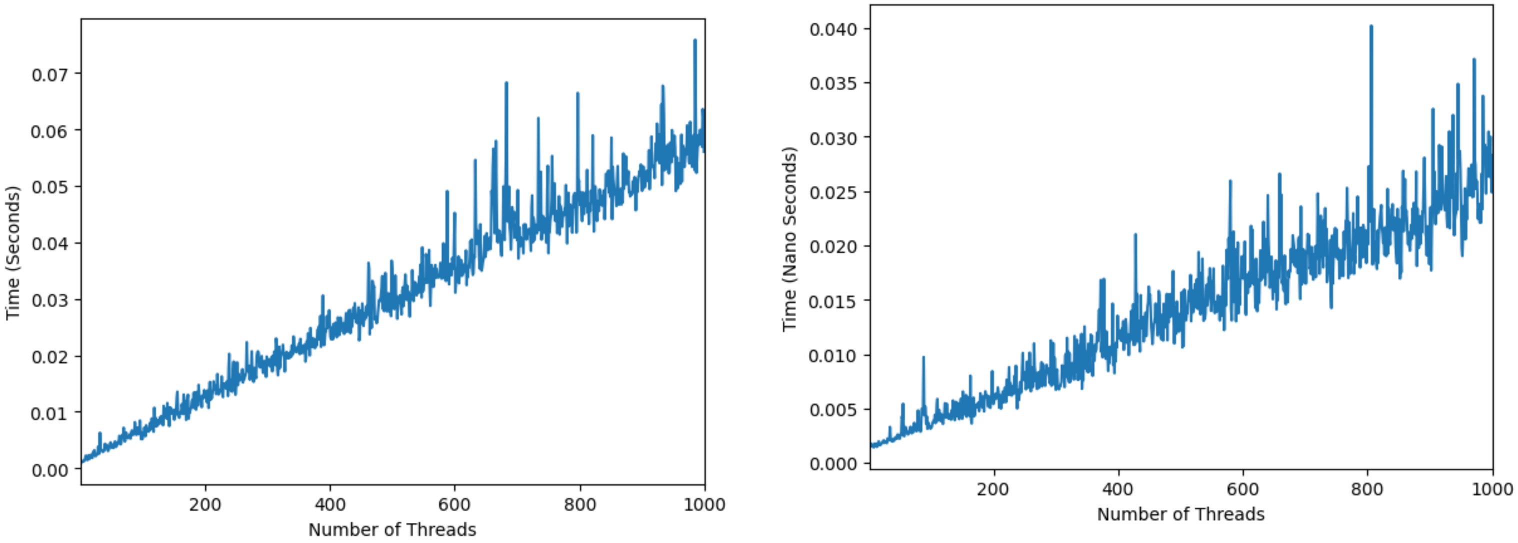


Plots of P1 and P2 for workload of 40x50 and 50x60 matrices.

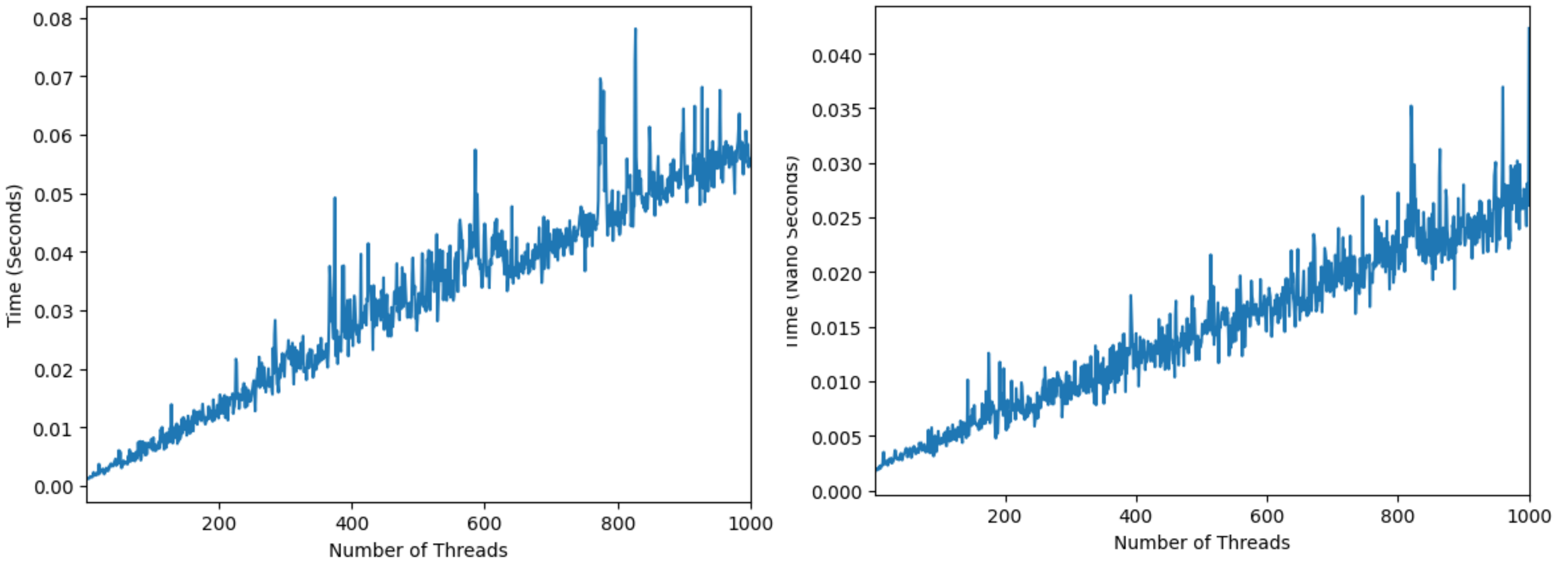




Plots of P1 and P2 for workload of 60x70 and 70x80 matrices.



Plots of P1 and P2 for workload of 100x100 and 100x100 matrices.



Analysis

We have Plotted the Graphs for the Programs P1 and P2 by varying the no of threads for a given time and calculating the time taken to complete the task assigned to the threads

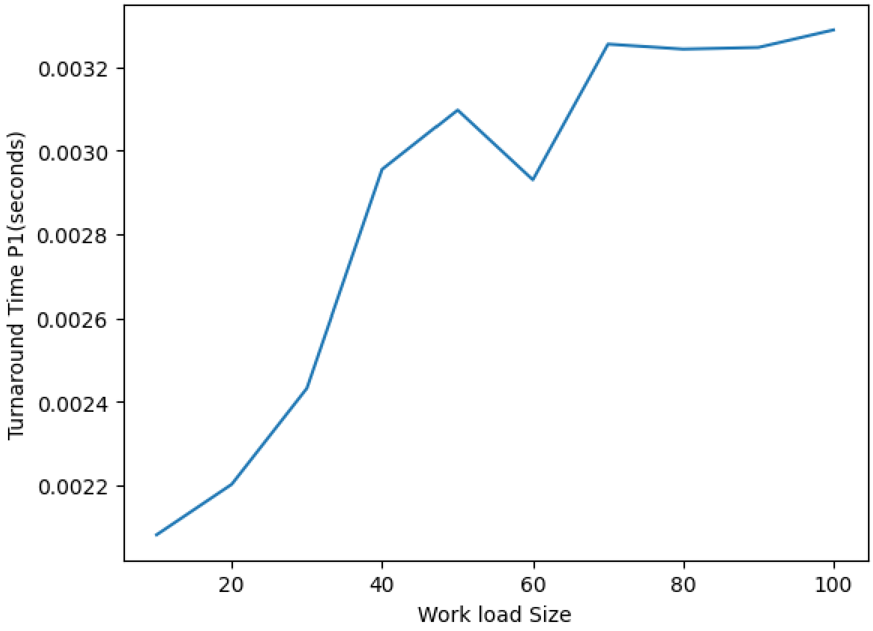
We have found that using 2 threads for P1 (1 for reading matrix 1 and 1 for reading matrix 2) is giving us an optimal result for all the input sizes

We are getting that, the use of 1 thread for reading the rows of the first matrix and 1 thread for reading columns of the second matrix is optimizing for P1. We have taken different input sizes and found the above conclusions.

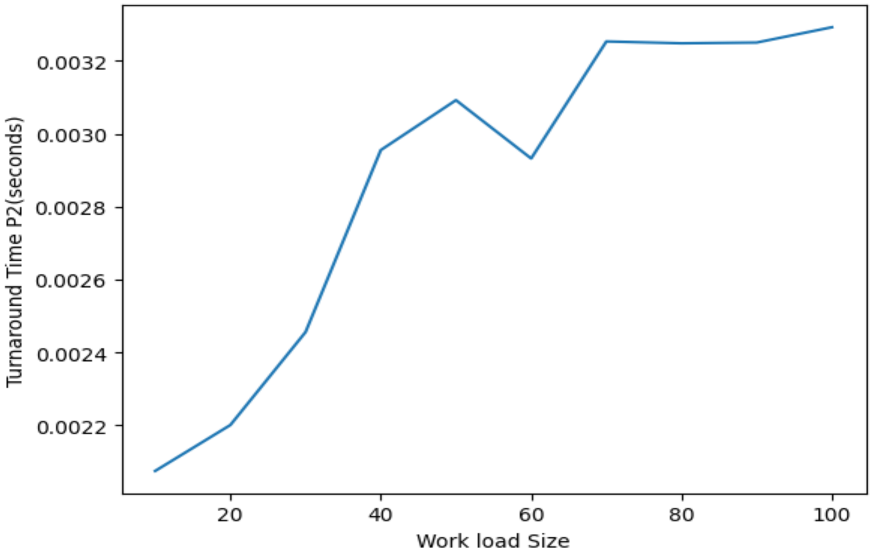
We are maintaining a boolean shared memory so that during scheduling , to compute the final value of the product matrix, if both the values to be multiplied are read prior to this computation, then only the threads proceed with calculating the other products of the respective row and column. Hence it does not have the problem of inconsistency and the threads are utilized to the fullest. **Hence this is the optimal solution we could figure out** Hence we are able to achieve consistency through just 2 threads for P1 and as the thread overhead is less for a lesser number of threads we have chosen to go with 2 threads for reading matrices as in P1 and 1 thread for multiplication of the matrices provided (P2). This conclusion is clear from the graph as well.

d)iv)

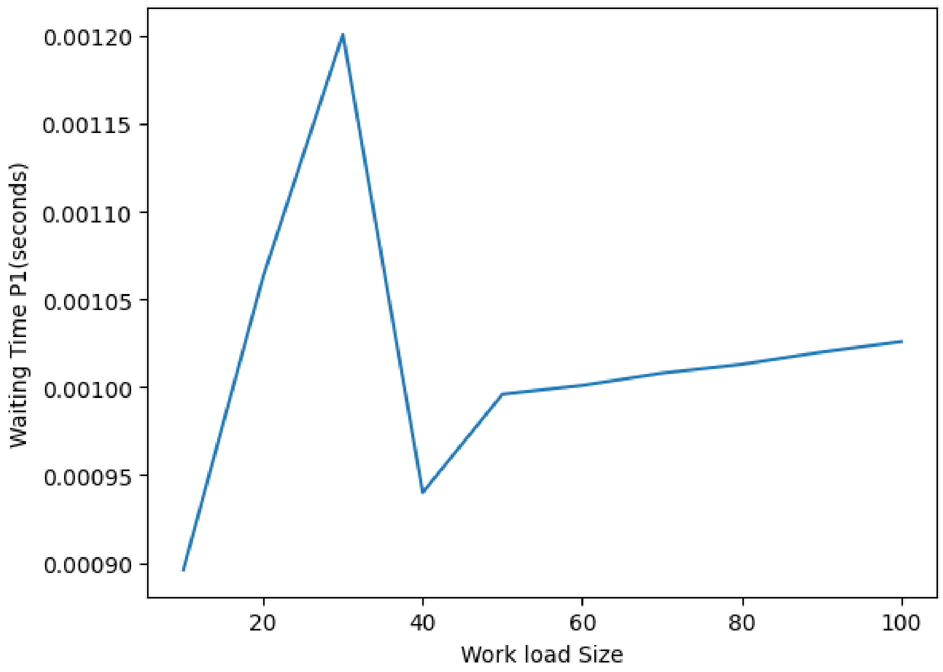
Plots for time quantum q = 1ms



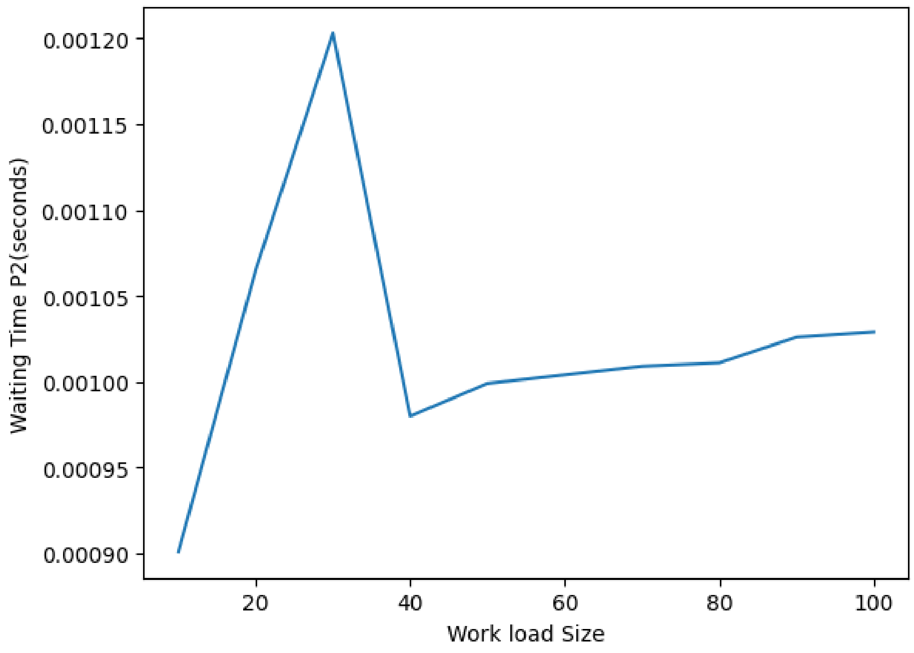
Turnaround time of P1 vs Workload size



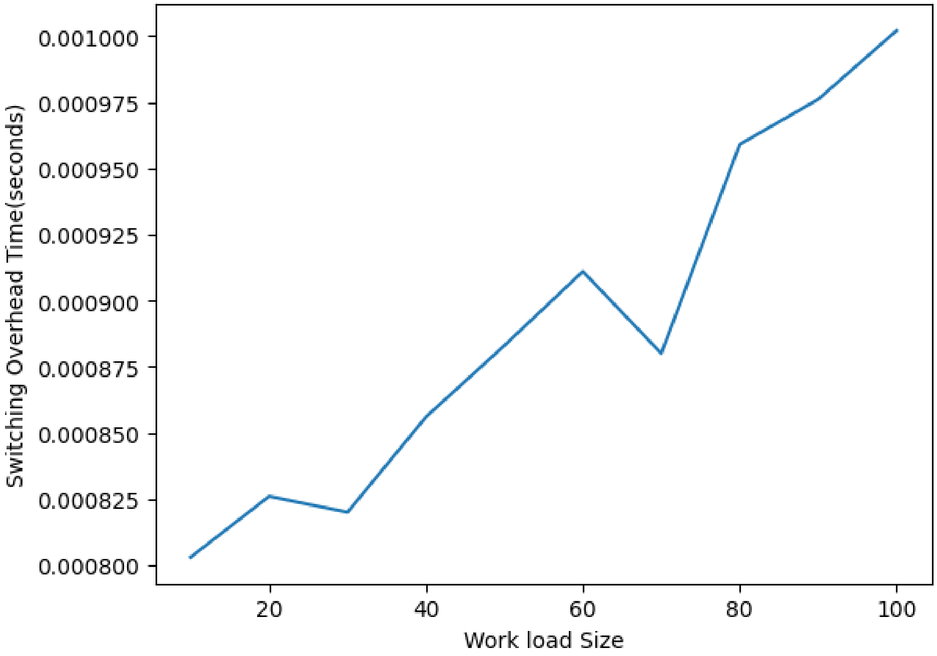
Turnaround Time of P2 vs Workload Size



Waiting Time of P1 vs Workload Size



Waiting Time of P2 vs Workload Size

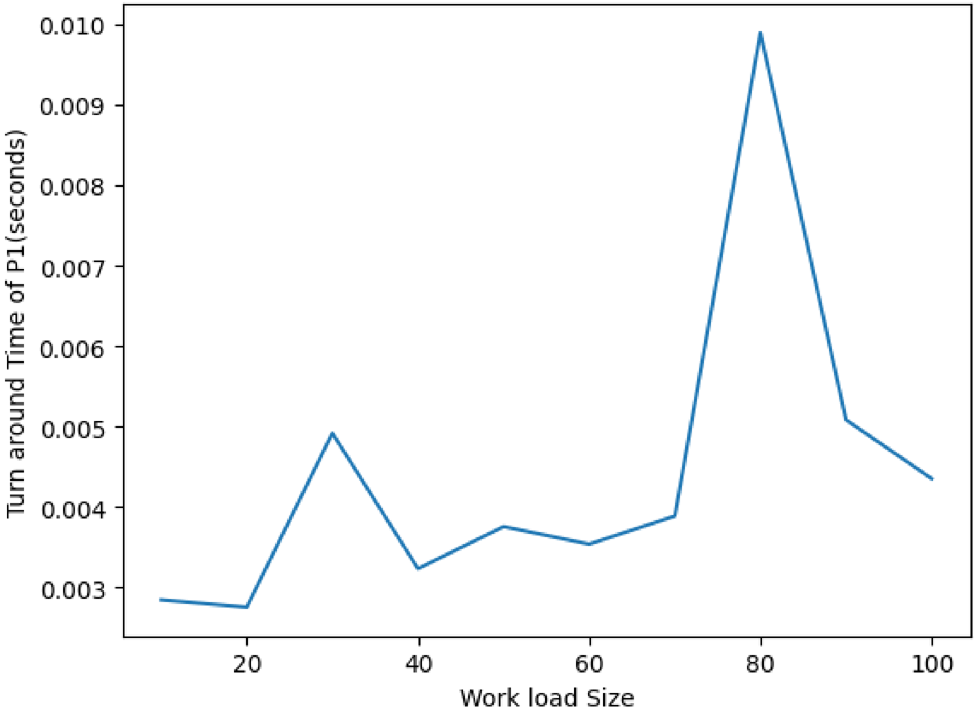


Switching Overhead vs Workload Size

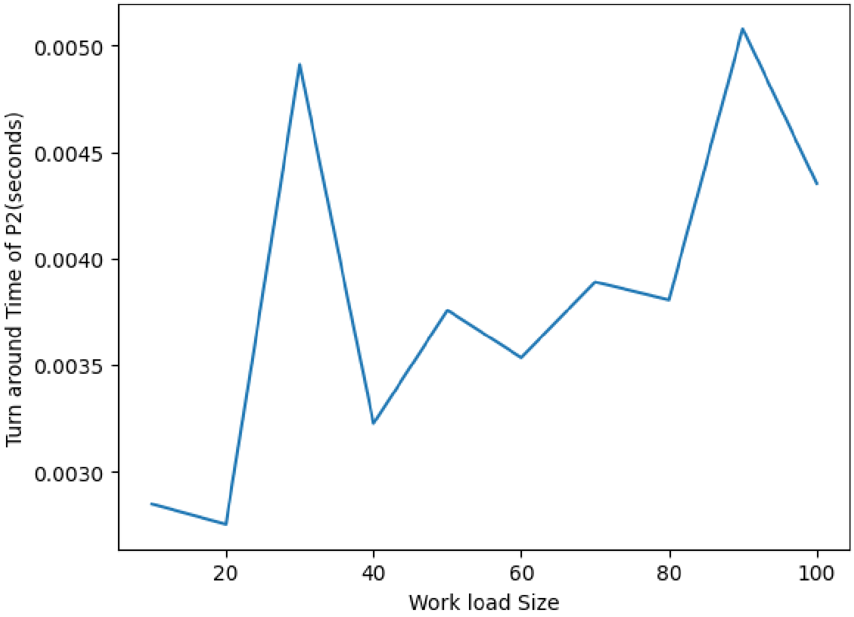
Analysis:

We have drawn plots for various workload sizes from
Matrix1 : 10x20 , Matrix2: 20x30
To Matrix1: 100x100 and Matrix2:100x100.
We have plotted the TurnAround Time,Waiting Time and Context Switching Time for P1 and P2 Programs for quantum of 1ms for Round Robin Scheduling Algorithm
We can see that the TurnAround Time increases as the size of the input increases, it is mainly due to the fact that computations take more time to complete hence time increases. Same is the case for P1 and P2 and in most of the cases the graph we are getting is consistent with this result. The Waiting time is also increasing with the workload size for P1 and P2, for some workload size it is decreasing since the execution time of a program is being almost the same as the quantum time. Context Switching time is being increased due increase in no of context switches as input size increases and decreasing for some higher inputs where the execution time of the program is comparable to quantum time

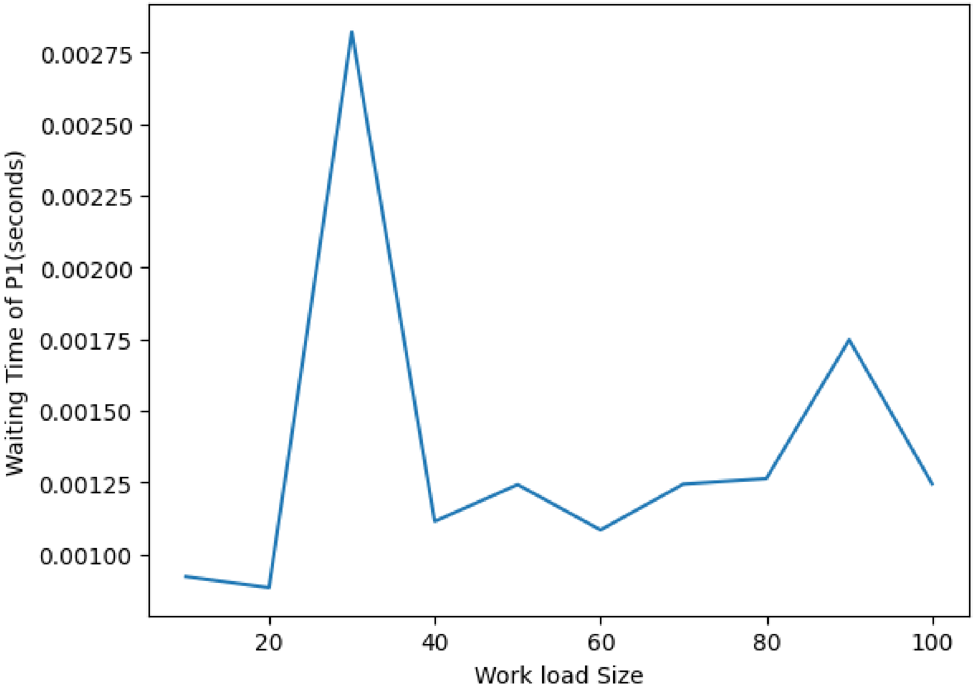
Plots for time quantum q = 2ms



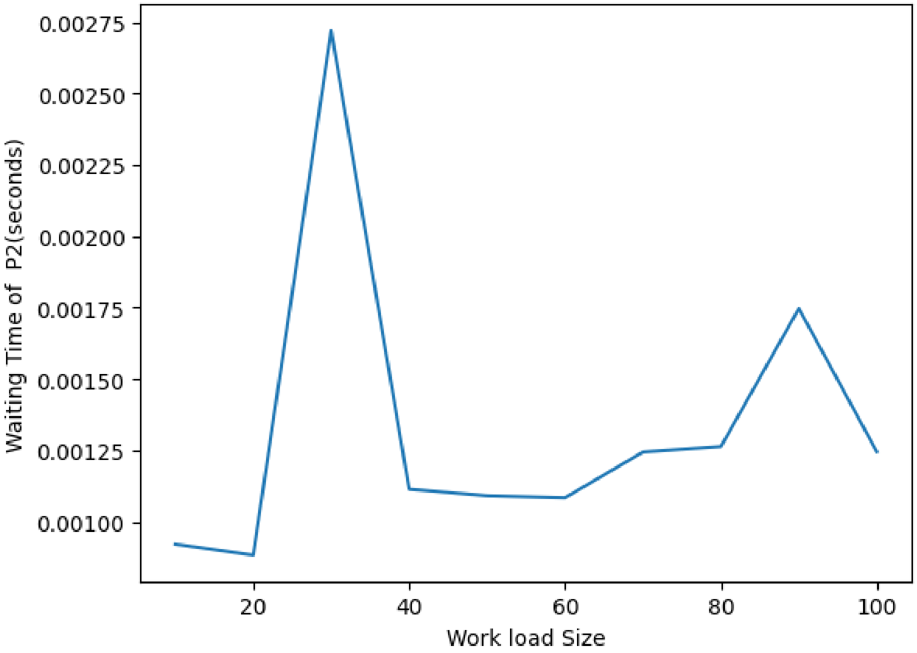
Turnaround Time of P1 vs Workload Size



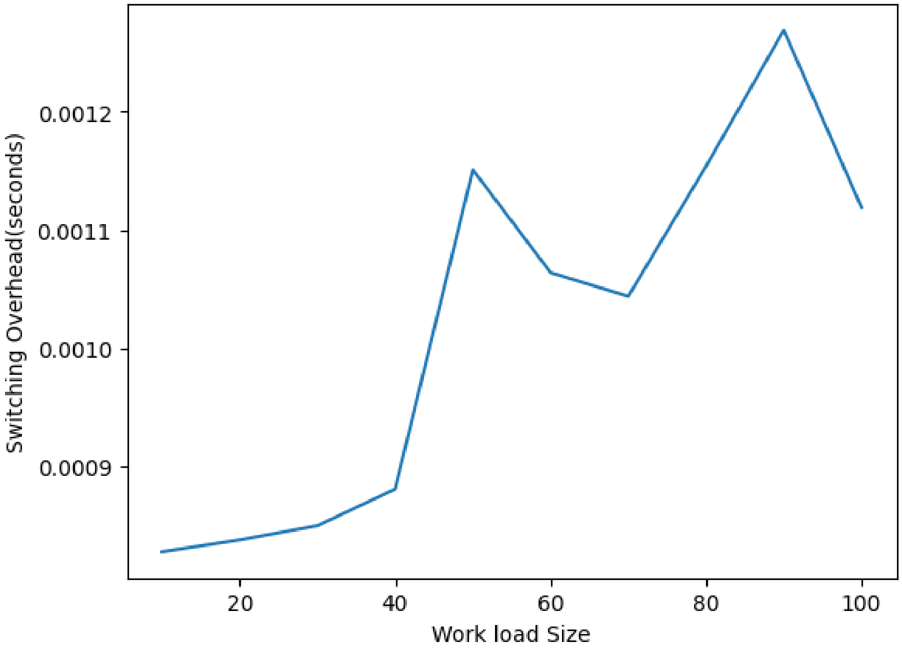
Turnaround Time of P2 vs Workload Size



Waiting Time of P1 vs Workload Size



Waiting Time of P2 vs Workload Size



Switching Overhead vs. Workload Size

Analysis:

We have drawn plots for various workload sizes from
Matrix1 : 10x20 , Matrix2: 20x30
To Matrix1: 100x100 and Matrix2:100x100.
We have plotted the TurnAround Time,Waiting Time and Context Switching Time for P1 and P2 Programs for quantum of 2ms for Round Robin Scheduling Algorithm
We can see that the TurnAround Time increases as the size of the input increases, it is mainly due to the fact that computations take more time to complete hence time increases. Same is the case for P1 and P2 and in most of the cases the graph we are getting is consistent with this result. The Waiting time is also increasing with the workload size for P1 and P2, for some workload size it is decreasing since the execution time of a program is being almost the same as the quantum time. Context Switching time is being increased due increase in no of context switches as input size increases and decreasing for some higher inputs where the execution time of the program is comparable to quantum time