CSC2200: Computer Science II
Honors Project Final Phase
Winter 2025

Student name: Mirshod Sobirov

I executed my source program on the WSU Grid using 16 CPU cores. My program initialized three matrices, the first with 1, the second with 2 and the resulting matrix with a 0. Then, I performed the matrix multiplication inside a nested loop ($O(N^3)$) and appended the resulting values to the result matrix. Overall, the whole process of individually conducting the 24 trials on the grid took me close to 3 hours, and here are the plots for each scheduling type, comparing the number of threads to the execution time. These plots were made using GNUPlot.

For static, static,8 and guided scheduling, a similar pattern was observed where the execution time was the largest for the fewest number of threads (1) and exponentially decreased as the number of threads doubled. The graphs for these three scheduling types are similar to that of exponential decay which shows that these three scheduling types behave similarly and the execution time reduces as more threads are involved.

For dynamic scheduling however, the pattern is a bit different compared to the other three types of scheduling. As you can observe from the plot, dynamic scheduling had a rather high execution time for all the variations of number of threads compared to the other scheduling types with the minimum execution time being over 400 seconds. This shows that dynamic scheduling takes the longest time to execute a program like matrix multiplication. This could be explained by the behavior of the nested loops inside the program. Since each loop iteration roughly takes equal time (basic multiplication and adding the value to the matrix), static scheduling might be more suited whereas dynamic scheduling is usually used for instances where each loop iteration takes a different amount of time because dynamic scheduling better distributes the work evenly amongst the number of threads.

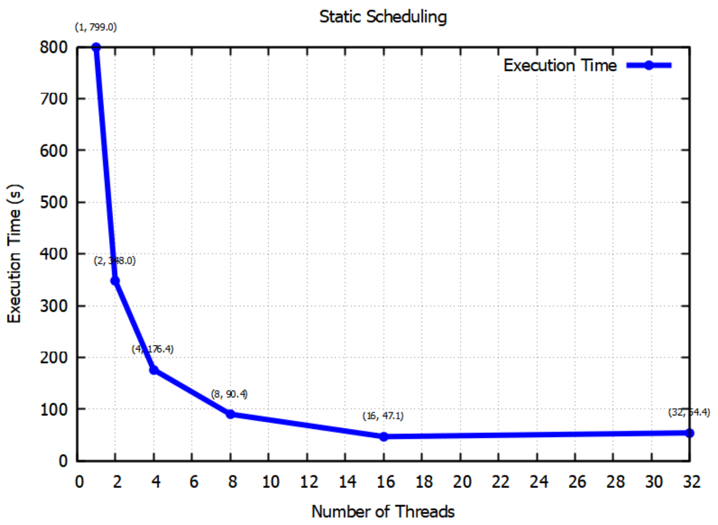

Figure 1: Execution Time vs. Number of Threads for Static Scheduling

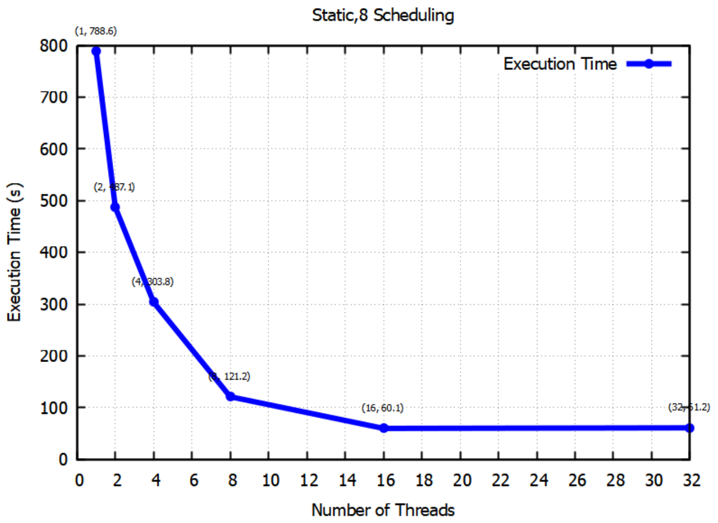

Figure 2: Execution Time vs. Number of Threads for Static,8 Scheduling

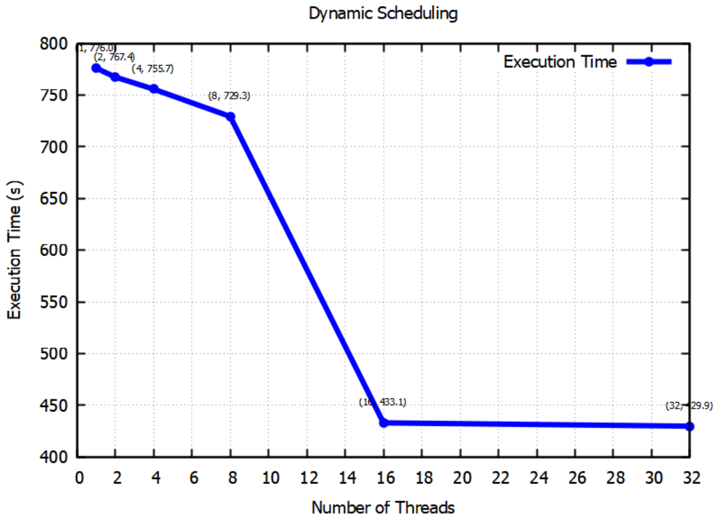

Figure 3: Execution Time vs. Number of Threads for Dynamic Scheduling

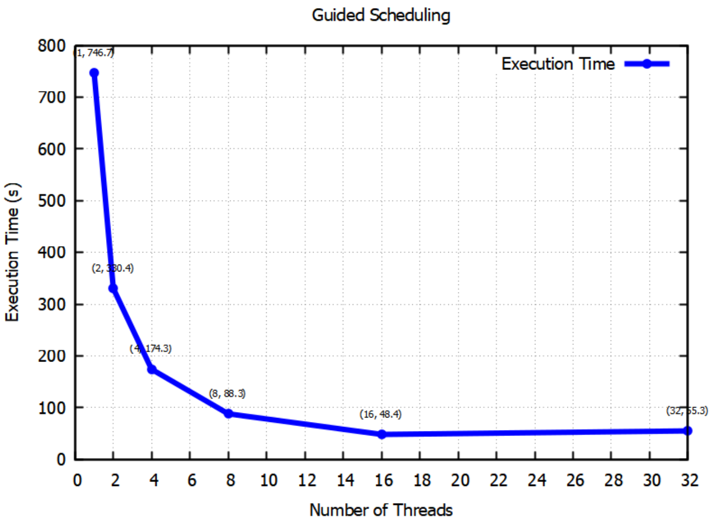

Figure 4: Execution Time vs. Number of Threads for Guided Scheduling