Parallel Computing

CSCE 435 - 2022 Summer

HW 1

916000990 Yasin Alam *Instructor: Vivek Saren*

All code was run on HPRC Grace.

Problem 1:

(1.1 - 1.3)

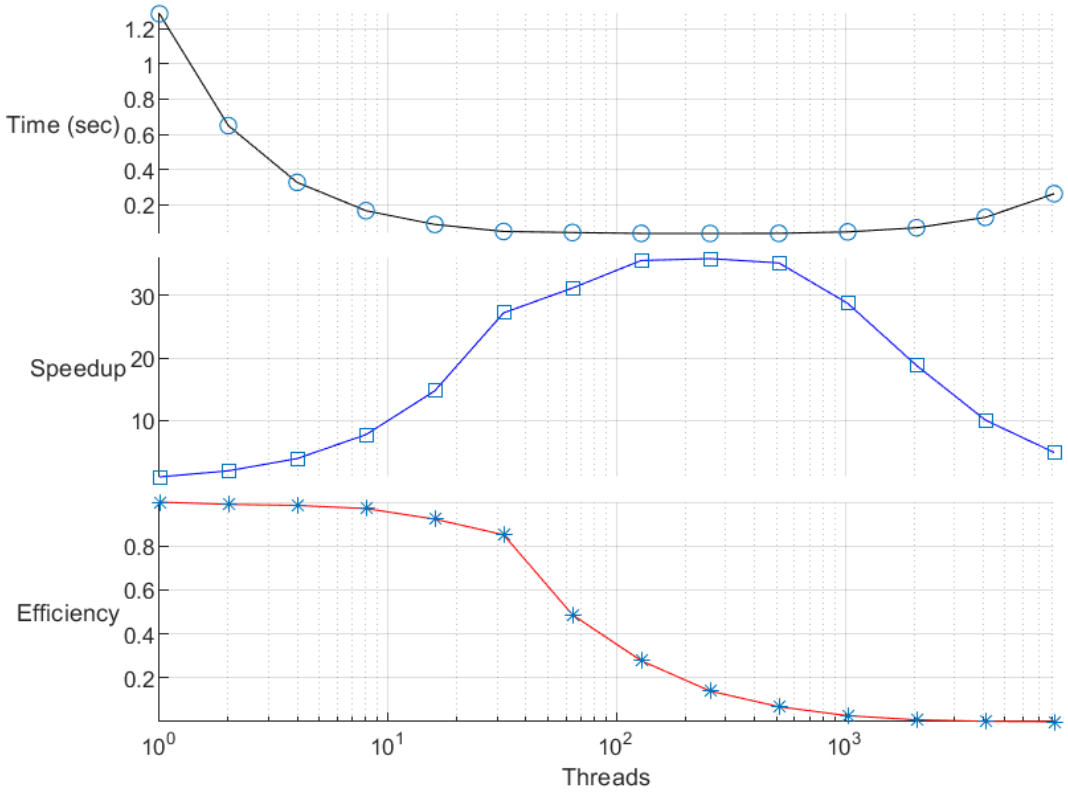

Figure 1: Executions time, speedup, and efficiency versus p with $n = 10^8$.

(1.4) The value of p that minimizes the parallel run time is $p = 256$ with a time (sec), speedup, and efficiency tuple of $(0.0358, 35.9106, 0.1403)$.

Problem 2:

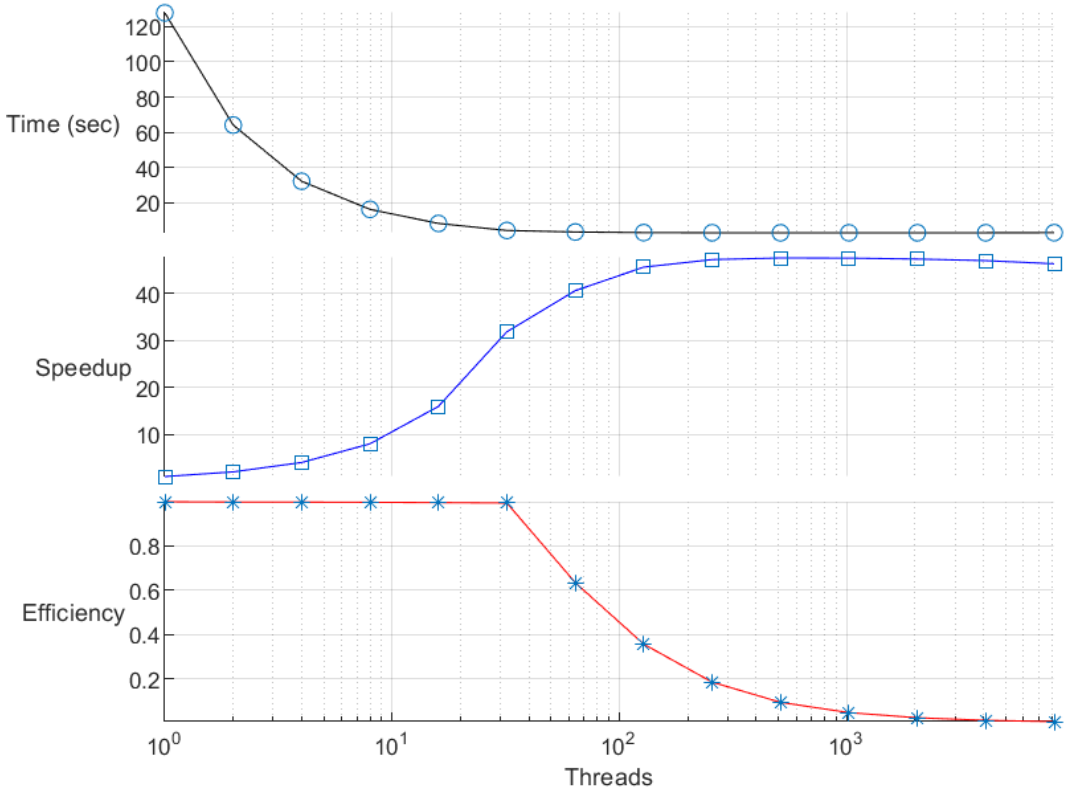

Figure 2: Executions time, speedup, and efficiency versus p with $n = 10^{10}$.

(2.1) The value of p that minimizes the parallel run time is $p = 512$ with a time (sec), speedup, and efficiency tuple of $(2.6911, 47.5264, 0.0928)$.

(2.2) The runtime will increase beyond a certain point due to the overhead of managing all of the processes. There is some small part of the program that cannot be parallelized, or cannot take advantage of all the threads at the same time. Some threads will have too much downtime. This is shown as past $p = 256, 512$ for Fig. 1 and Fig. 2 respectively, the runtime starts to go back up.

Problem 3:

It is very clear that for different n's, there would be a different number of optimal threads. Experiments confirm this, see Table. 1.

n	Minimum time (sec)	Threads (p)
10^8	0.0358	256
10^{10}	2.6911	512

Table 1: Comparison of n and number of threads p. The optimal number of threads differ based on n.

Problem 4:

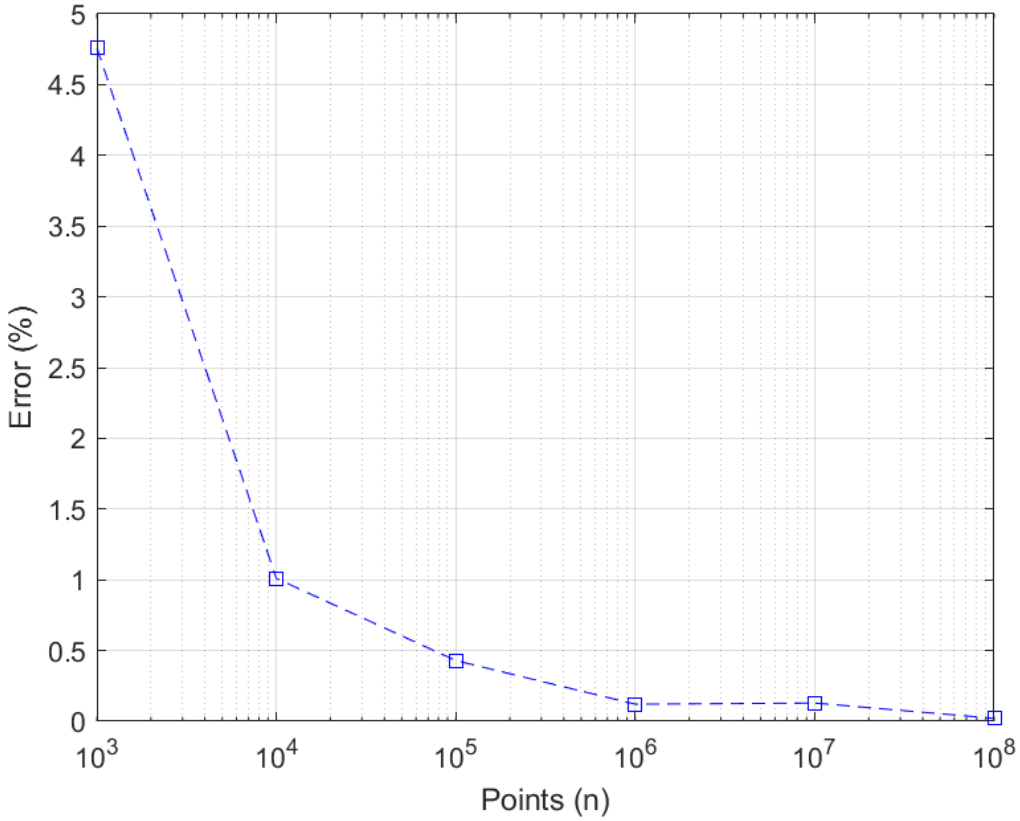

Figure 3: Error versus n with $p = 48$.

Problem 5:

(5.1 - 5.3)

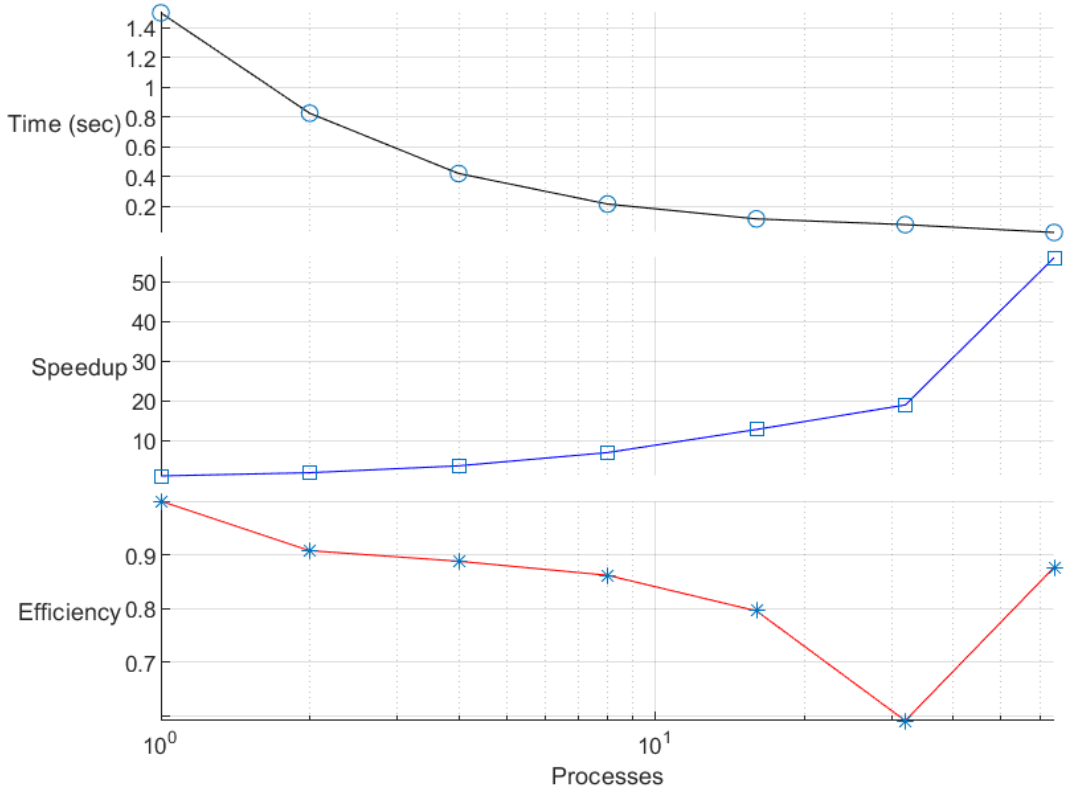

Figure 4: Executions time, speedup, and efficiency versus p with $n = 10^8$.

(5.4) The value of p that minimizes the parallel run time is $p = 64$ with a time (sec), speedup, and efficiency tuple of $(0.0267, 56.1161, 0.8768)$.

Problem 6:

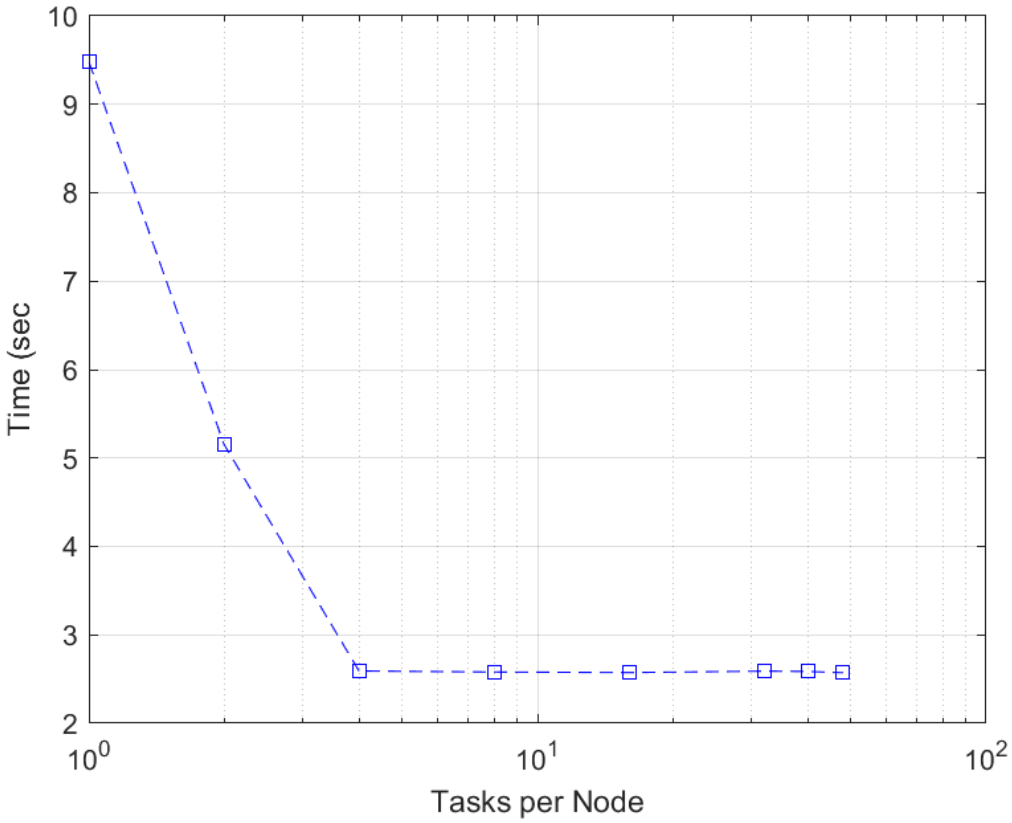

Figure 5: Tasks per node versus time with $n = 10^{10}, p = 64$.

The number of tasks per node that produces the minimal run time in our experiments was 48 with a time of 2.5721 sec; however, the number of tasks per node remains almost constant after 4 tasks per node and all of the points after are within a 100th of a second of this minimum time. It is difficult to tell whether this is an artifact of loads on grace or if there is an actual difference without repeated trials. As there are 64 processes with 16 nodes, it seems optimal to utilize 4 tasks per node and increasing this number is redundant and seems meaningless.

Problem 7:

(7.1)

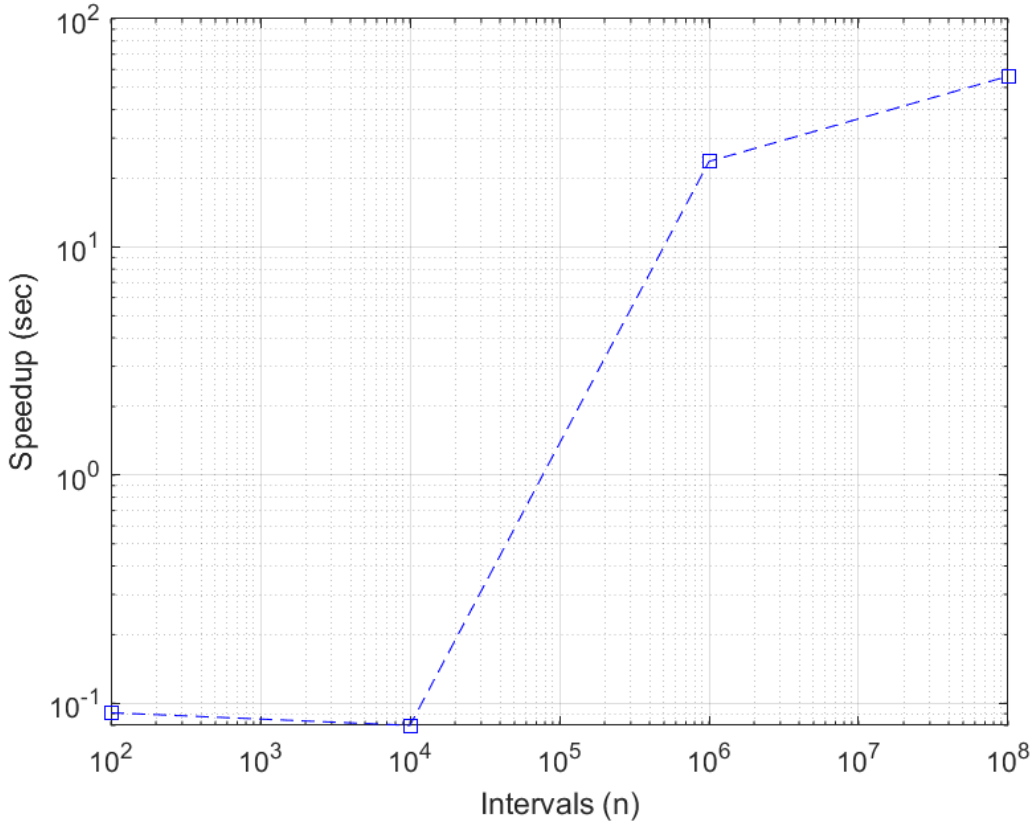

Figure 6: Speedup versus n on $p = 64$ with respect to $p = 1$.

(7.2)

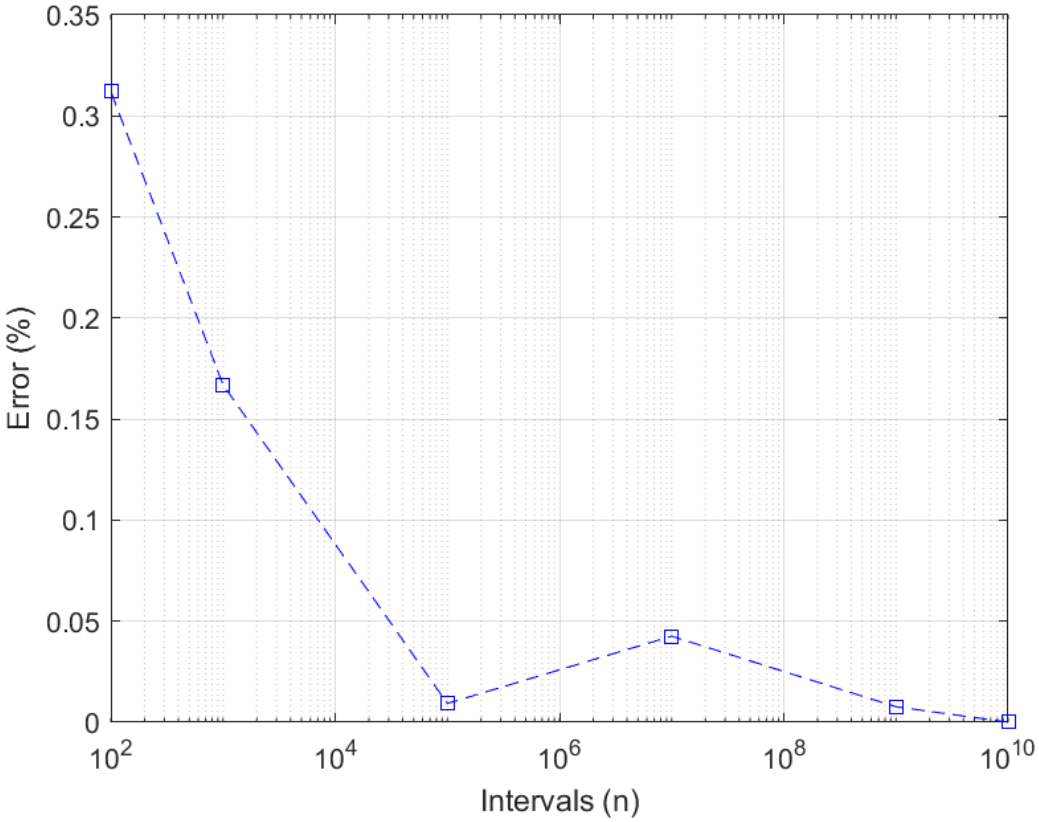

Figure 7: Error versus n.