

Homework 2 Report

Group Members:

Name: Pranay Morye

UCINetID: pmorye

Name: Junaid Magdum

UCINetID: jmagdum

System Description:

1. Processor: Intel (R) Core (TM) i7-7500U CPU @ 2.70GHz with 4 cores
2. RAM Memory: 8GB RAM
3. Operating System: Linux (Ubuntu)

Execution Time:

1. Single Thread Program:

Execution Time = 14.3452 seconds

Multi-Thread Program:

Number of Threads	Execution Time
1	14.586565
2	7.379754
4	5.525782
8	5.497344
16	5.520126
32	5.504304
64	5.510526
128	5.489648

Analysis:

Speedup of program in Multi-threaded version: As we can see from the data, the performance of the program initially increases manifolds with the increase in the number of threads, however it gradually slows down to around 5.5 seconds. What we can infer from this is that even though in theory, doubling the number of threads should halve the run-time, this does not factor in other tasks such as thread-management done by the program, thread creation, deletion, number of cores available, etc.

Thus, initially when we increase number of threads from 1 to 2, we can see halving of run-time because we have more than 2 cores, but eventually this run-time halving slows down with the doubling of the number of threads because of the above mentioned factors. In fact, we can also see some slowdown as we increase threads because this is where the thread-management overhead becomes more than the benefit of having more threads.

We can conclude that we need to figure out an optimum number of threads to run our program for it to run efficiently.

Single-threaded VS Multi-threaded: As we can see from the data above, the single-threaded version runs a little faster than the multi-thread version with only 1 thread. The reason for this is that the multi-thread version has to do thread creation, management and destruction even for that 1 thread. So, even though both programs have technically the same run time complexity, we can see that the multi-thread program takes more clock-time.