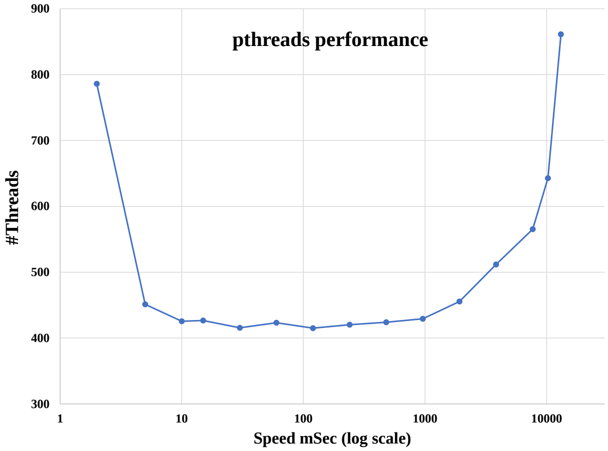
TLP Project Report

Ahmed H. Mahmoud

1 pthreads:

The serial code was parallelized using pthreads by dividing the computation on number of threads. Since the 2D picture is mapped into 1D array (*char* pic*), the 1D array elements were distributed to the number of threads specified by the user. Each thread carried the computation for certain number of elements and updated the globally-defined 1D array. The number of threads specified should be divisible by total number of elements. We tested varying number of threads and for each experiments we reported the performance (in msec) as shown in Figure 1. We can see that there is a range for which changing the number of threads does not affect the performance. This range is from $10 - 1920$ threads, given that that 2D picture is 1920×1920. For higher than 1920 threads, the cost of lunching more threads overcomes the actual computation. The best performance was obtained using 120 threads which gave 415.097331 mSec.

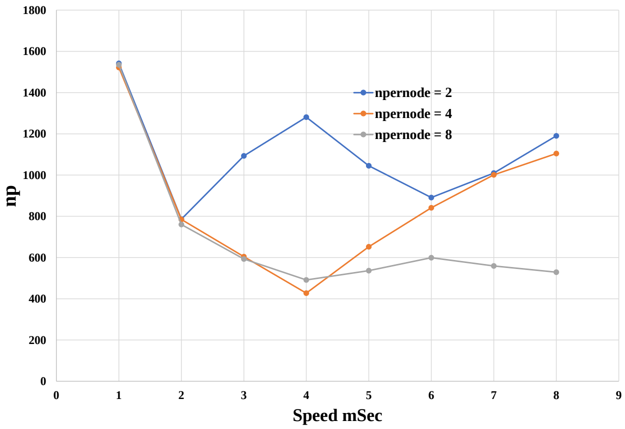


(a) pthread

Figure 1: Performance of pthreads using varying number of threads

2 MPI:

The serial code was parallelized using MPI using master-slave approach. One processor was responsible of scattering the 1D array across the other processor and then gather the data after all processors have finished their assigned computation. Similar to the pthreads, each processor was responsible of updating a part in the 1D array as assigned by the master processor. We tested the performance for varying configuration as shown in Figure 2 where increasing the number of nodes is not always beneficial. The best performance can be obtained using 4 nodes where each node launch 4 processes with which the computation finishes in 427.4139 mSec. This still does not outperform the speed up obtained from pthreads.



(a) pthread

Figure 2: Performance of MPI using varying processes per node (npernode) and number of processors (np)

3 Comparison:

By viewing the prices of dual-core Intel processors vs quad-core (https://en.wikipedia.org/wiki/List_of_Intel_Core_2_microprocessors), we can see that the prices for the quad-core is almost 2.5 times the dual-core one. Given the test program we run, since only a comparable performance can be obtained using 4 processors on MPI, we recommend to use dual-core thread-capable processor since it is more cost effective.