CS508 - Assignment-3 Report

Aaditya Arora B17071

Note:

1. Instruction of executing a program is written inside each program as comment.
2. File structure P1/p1.c P2/p2.c P3/p3.c
3. All observations are taken with gtime (gnu time) shell command in Mac OS.

P1.

Performance gain or loss depends on the size of the input array.
In the case of Sequential -- There is no extra work of threading.
In the case of MPI -- There is extra work of passing the array from one array to another.
In the case of OpenMPI -- There is extra work of handling fork / join.

For Array Size : 1e5

Sequential : 12.36s
MPI : 1.50s
OpenMP : 0.01s

For Array Size : 1e4

Sequential : 0.65s
MPI : 0.130s
OpenMP : 0.01ss

For Array Size : 1e3

Sequential : 0.00s
MPI : 0.11s
OpenMP : 0.00s

In all cases OpenMP is faster because of nested thread level parallelism. In the case of MPI time spent in Message Passing is a bottleneck for a smaller size of array whereas In sequential the computation time is higher because of O(n^2) complexity.

P2.

In this program, main parallelization happens while computing the 4x8 histogram of the image.

In this we are doing the same kind of parallelization as in MPI but because of shared memory the performance of OpenMP will be higher than that of MPI.

For Image Size : 540 x 360

Sequential : 0.03s
MPI : 0.51s
OpenMP : 0.00s

For Image Size : 1024 x 1024

Sequential : 0.04s
MPI : 0.11s
OpenMP : 0.03s

Order of execution time : OpenMP < Sequential < MPI.
The Main reason for low performance of MPI is because message passing is a bottleneck here.

P3.
This question is mostly the same as of P2 but instead of one image we are loading two images.
In the case of OpenMp, we are utilising nested thread level parallelism.

For Image Size : 540 x 360

Sequential : 0.02s
MPI : 0.11s
OpenMP : 0.01s

For Image Size : 1024 x 1024

Sequential : 0.10s
MPI : 0.15s
OpenMP : 0.06s

In this case order of execution Sequential ~ OpenMp < MPI
The reason for the higher execution time of OpenMp is because of the time spent in message passing which is the bottleneck.