

PARALLEL COMPUTING ASSIGNMENT 2

BHUVNESH JAIN (2014A7PS028P)
CHIRAG AGARWAL (2014A7PS033P)

Problem : Assignment Problem

a. OpenMP Version

The below results were obtained on "Intel Core i7" quad-core processor using 8 threads. The programs were written in "C++" language and OpenMP was used for writing the parallel code. 2 different datasets were considered. Each dataset was run for 5 times and the average execution time was recorded below. The datasets were as follows:

Number of people = 18, Cost of assigning task \leq 100

| Number of Threads | Execution Time | Speedup |
|-------------------|----------------|---------|
| 1 | 4.787970 | - |
| 2 | 2.510890 | 1.906 |
| 4 | 1.95368 | 2.4507 |
| 8 | 1.764598 | 2.7133 |

Number of people = 20, Cost of assigning task \leq 1000

| Number of Threads | Execution Time | Speedup |
|-------------------|----------------|---------|
| 1 | 67.929322 | - |
| 2 | 36.007260 | 1.886 |
| 4 | 21.064709 | 3.224 |
| 8 | 17.23487 | 3.9413 |

Conclusion:

The scaling was almost linear till 4 threads but sub-linear afterwards due to hyper-threading playing a major role.

b. MPI Version

The below results were obtained on a cluster of 2 nodes each with "Intel Core i7" quad-core processor. The programs were written in "C++" language and MPI was used for writing the parallel code. 2 different datasets were considered. Each dataset was run for 5 times and the average execution time was recorded below. The datasets were as follows:

Number of people = 18, Cost of assigning task ≤ 100

(Sequential code on single processor took 0.554 seconds)

| Number of Processors | Execution Time | Speedup |
|----------------------|----------------|---------|
| 2 | 0.601991 | 0.9202 |
| 4 | 0.376410 | 1.4717 |
| 8 | 0.268912 | 2.060 |
| 16 | 0.439047 | 1.2618 |

Number of people = 20, Cost of assigning task ≤ 1000

(Sequential code on single processor took 6.435 seconds)

| Number of Processors | Execution Time | Speedup |
|----------------------|----------------|---------|
| 2 | 6.958607 | 0.924 |
| 4 | 3.281485 | 1.9610 |
| 8 | 1.937892 | 3.3206 |
| 16 | 4.246829 | 1.5152 |

Conclusion:

The time taken by 2 processors was slower than the single one as communication cost added overheads. Afterwards the scaling was almost linear till 4 processors, then sub-linear till 8 processors as the speed of interconnect would be the determining factor in this case. Since, a total of 8 processors were available on these 2 computers, the time taken by calling 16 processors increased as almost half of the processors remained idle for most of the time. Also, the master node never did the work of solving the task as just did the work of communication, hence the scale up was also less.