MTCS-103P : Lab Work- 11

Floyd’s Algorithm

Regd no: 22555

Introduction

The Floyd Warshall Algorithm is used for solving the all pairs shortest path problems. The problem is to find the shortest distances between every pair of vertices in a given edge-weighted directed Graph.

Computation time of sequential algorithm:

The time complexity of Floyd’s sequential algorithm is O(n^3). The scope of parallelism is in the innermost two loops of Floyd's algorithm.

Computation time of parallel algorithm:

Time complexity of the parallel Floyd’s algorithm is O(n^3/p + (n^2)log p) where p is the number of processes available.

Communication time complexity:

The communication cost of Floyd’s algorithm is (n logP(λ+4nβ)) where λ is the message latency and β is the bandwidth of communicating among processes.

Execution time complexity:

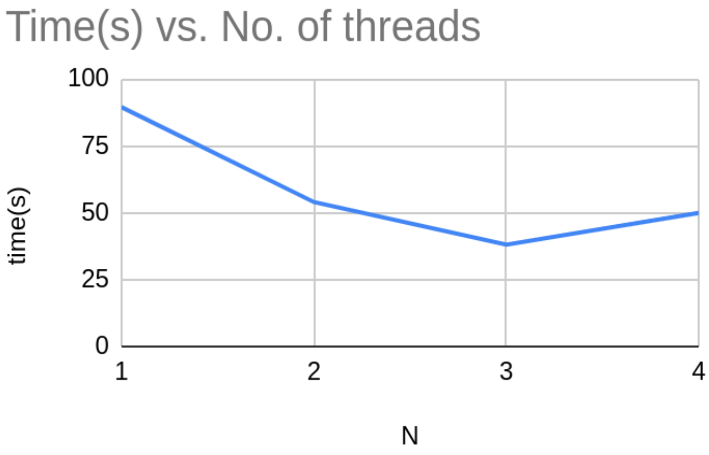
The total execution time complexity of the Floyd’s algorithm is the sum of computation time complexity and communication time complexity.

Therefore the execution time complexity is: (n^3 (O(n^3)) + n logP(λ+4nβ))

MPI:

Matrix size: 2k

N	Time(s)
4	49.95356
3	38.1578
2	53.9854
1	89.4575

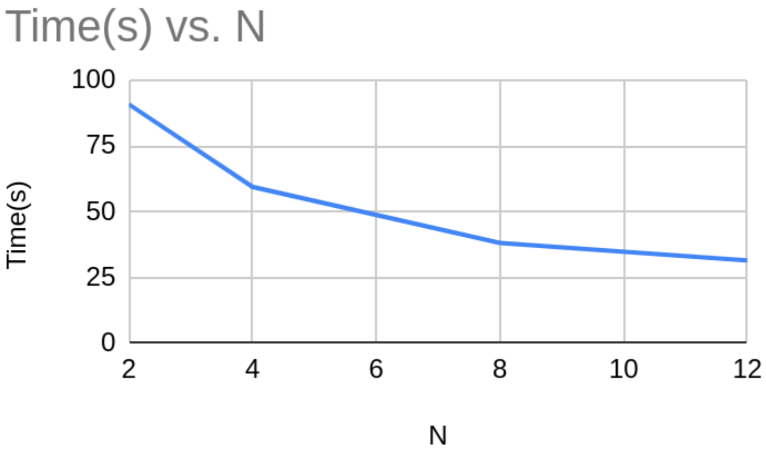


It can be seen from the below graph and above table as the number of processes increases computation time decreases but after a certain number of threads computation time increases again because the parallel overhead increases.

OPENMP:

Matrix size: 2k

N	Time(s)
12	31.4654
8	38.1578
4	59.4564
2	90.8546



It can be seen from the above graph and table as the number of threads increases computation time decreases.

Conclusions:

The parallel program outperforms sequential code as the time almost becomes half of the sequential code. But when the number of processes or threads increases then the communication time also increases which therefore leads to more execution time as the synchronization process is a costly operation.