Project Proposal - Maojun Xu(maojunx2) & Zhicheng Tang (zt17)

1. Introduction

This project will implement parallel BFS algorithm to compute the distances to all vertices starting with any random vertex. The implementation mainly utilizes MPI and OpenMP as toolkits. Generally, we will develop a non-parallel BFS method as our baseline. Then, we will develop three parallel BFS solutions with OpenMP, MPI and a hybrid version respectively. Finally, we will compare the performance and scaling poperties of different methods.

2. Implementation of Non-Parallel Breadth First Search

2.1 Data Structure and Representation

The graph data is represented by an adjacency matrix A, stored in a two-dimensional array. The value of A_{ij} denotes the distance between vertex i and vertex j. The diagonal terms represent the degree of each vertex. The BFS algorithm implementation makes use of a queue and an array to record the visited vertices.

2.2 Algorithm

i. Initialize an empty two-dimensional array, an empty queue, and a visited array with all values set to false.

ii. Randomly select a vertex and push it into the queue.

iii. While the queue is not empty, perform the following steps:

a. Pop the first vertex from the queue and assign it as the current vertex.

b. Record the distances between the current vertex and its neighbors using the adjacency matrix.

c. Push the unvisited neighbors of the current vertex into the queue.

d. Update the visited array values for the current vertex's neighbors to be true.

Iv. Repeat the step 3 until the queue is empty.

3. Parallel Implementation of Breadth First Search

For OpenMp, we plan to try two methods. One method is to use #pragma omp parallel for, it can parallelize the loop that visits all the neighbors of the current node. Another is to use #pragma omp task, whenever we discover a new node, it can spawn a task and assign the task to visit that new node.

For MPI, we plan to try "MPI_Bcast", “MPI_Scatterv”, “MPI_Send” and “MPI_Recv”. In the beginning, we scatter parts of nodes across all members of the group. In each level of BFS, the root process collects all visited neighbors from other members, and then sends nodes which are in the current level to all other members.