# Parallelization Strategy for Updating Single Source Shortest Paths in Dynamic Networks

## **Submitted By**

### **M Azeem Ashfaq 22I-1057**

### **M Bilal Tariq 22I-1297**

### **M Nabeed Haider 22I-0871**

### 

### **Date: 20th April 2025**

## **Abstract**

This document proposes a parallelization strategy for updating Single Source Shortest Paths (SSSP) in large-scale dynamic networks using a combination of MPI for inter-node communication, OpenMP for intra-node parallelism, and METIS for graph partitioning. The goal is to efficiently update SSSP in response to dynamic changes in the network structure.

## 

## 

## **1. Introduction**

Dynamic networks frequently require efficient algorithms to update properties such as shortest paths as the structure changes. The proposed strategy leverages parallel computing techniques to enhance performance in processing large dynamic graphs, as outlined in the work by Khanda et al. (2021).

## **2. Methodology**

### **2.1 Graph Partitioning with METIS**

* Objective: Divide the graph into smaller, manageable partitions to minimize inter-node communication.
* Approach:
  + Use METIS to partition the graph into p partitions, where p is the number of available nodes.
  + Each partition will be assigned to a different node in the MPI environment.
  + Ensure that the partitions are balanced in terms of the number of vertices and edges to optimize load balancing, as dynamic networks often have unstructured and sparse characteristics.

### **2.2 Inter-Node Communication with MPI**

* Objective: Facilitate communication between different nodes processing different graph partitions.
* Approach:
  + Initialize MPI and determine the rank of each node.
  + Each node will process its assigned graph partition independently, updating the SSSP tree based on local changes.
  + Use MPI\_Send and MPI\_Recv for communication between nodes to share necessary data, such as updated distances or affected vertices.
  + Implement collective communication (e.g., MPI\_Allreduce) to gather results from all nodes when necessary, especially after processing edge insertions and deletions.

### 

### 

### **2.3 Intra-Node Parallelism with OpenMP**

* Objective: Utilize multi-threading within each node to process the graph partition efficiently.
* Approach:
  + Use OpenMP to parallelize the computation of SSSP within each partition.
  + Implement parallel loops to update distances and parent relationships for vertices in the partition, as described in the proposed algorithms in the document.
  + Use OpenMP directives (e.g., #pragma omp parallel for) to distribute the workload among available threads.
  + Ensure proper synchronization to avoid race conditions when updating shared data, particularly during the iterative updates of affected vertices.

## **3. Implementation Steps**

1. Graph Partitioning:
   * Load the graph data and apply METIS to partition the graph.
   * Distribute the partitions to different nodes.
2. MPI Initialization:
   * Initialize MPI and determine the number of processes and their ranks.
3. Processing Each Partition:
   * For each node:
     + Use OpenMP to parallelize the SSSP computation on the assigned partition.
     + Update the SSSP tree and maintain a list of affected vertices based on edge insertions and deletions.
4. Communication:
   * Use MPI to communicate updates between nodes.
   * Gather results and update the global SSSP tree as necessary.
5. Finalization:
   * Finalize MPI and clean up resources.

## 

## **4. Expected Outcomes**

* Improved performance in updating SSSP in dynamic networks, achieving significant speedup compared to recomputing from scratch.
* Efficient utilization of resources through parallel processing, leveraging both inter-node and intra-node parallelism.
* Scalability to handle large graphs with millions of vertices and edges, as demonstrated in the experimental evaluations of the proposed algorithms.

## **5. Conclusion**

This proposed parallelization strategy combines the strengths of MPI, OpenMP, and METIS to efficiently process dynamic graphs. By partitioning the graph, leveraging inter-node communication, and utilizing intra-node parallelism, we aim to achieve significant performance improvements in dynamic graph processing, as highlighted in the work by Khanda et al. (2021).

## **6. References**

* Khanda, A., Srinivasan, S., Bhowmick, S., Norris, B., & Das, S. K. (2021). A Parallel Algorithm Template for Updating Single-Source Shortest Paths in Large-Scale Dynamic Networks. IEEE Transactions on Parallel and Distributed Systems.
* METIS: A Software Package for Partitioning Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices.