PARALLELIZATION OF PRIM’S ALGORITHM TO FIND THE MST

By Sarath Chandra Reddy Rayapu
Minimum Spanning Tree (MST) of a graph

- A spanning tree (a tree with all the nodes in the graph) where the sum of the edges is the least possible.
Applications of MST

- Design of cost-effective Networks and efficient Circuits
- Transportation Planning: to determine the most cost-effective routes for building roads, railways, or other transportation networks.
- Image Processing: used in Image Segmentation
Prim’s Algorithm (Sequential):

1. Initialize a tree with a single vertex, chosen arbitrarily from the graph.
2. Grow the tree by one edge: Of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree.
3. Repeat step 2 (until all vertices are in the tree)
4. Time = O(n^2)
Pseudo code for Parallel approach

- **Initialization:**
- Divide the set of vertices \( V \) into \( p \) subsets \( V_1, V_2, ..., V_p \)
- Assign each subset to a different process
- While vertices_in_MST is not equal to \( V \):
  - For each process \( p_i \):
    - Find the minimum-weight edge \( e_i \) (candidate) connecting MST to vertices in \( V_i \)
    - Send \( e_i \) to the root process using MPI_Reduce to find the global minimum-weight edge \( e_{\text{min}} \)
  - If rank of current process is root:
    - Select the minimum-weight edge \( e_{\text{min}} \) from the received edges
    - Add \( e_{\text{min}} \) to MST
  - Broadcast \( e_{\text{min}} \) to all processes
  - Continue this till all the vertices are in the MST
- Time = \( O(n^{2/p}) + O(n \log p) \)
- Partitioning of adjacency matrix among ‘p’ processors:
Results

- Input graph: 10000 nodes (5% density)
Results

- Input graph: 10000 nodes (10% density)
Results

- Input graph: 10000 nodes (20% density)
Observations

- This algorithm works best with larger datasets by gaining considerable speedups.
- Also, higher density graphs are better suited for this as we are using an adjacency matrix to store the graph.
References

• Parallelization of Minimum Spanning Tree Algorithms Using Distributed Memory Architectures
  http://www.scl.rs/papers/Loncar-TET-Springer.pdf
THANK YOU