Parallel and Distributed Computing

Name	Roll No	Section
Haider Zia	22i-1196	CS-F
Muhammad Iqrash Qureshi	22i-1174	CS-F
Hammad Shabbir	22i-1140	CS-F

Phase 1 - Presentation: Research Paper Summary and Parallelization Strategy

Selected Paper:

Title: A Parallel Algorithm Template for Updating Single-Source Shortest Paths in Large-Scale Dynamic Networks

Authors: Arindam Khanda, Sriram Srinivasan, Sanjukta Bhowmick, Boyana Norris, Sajal K. Das

1. In-Depth Study of the Paper

This paper presents a scalable parallel framework to efficiently update Single-Source Shortest Paths (SSSP) in large dynamic graphs. Unlike static graphs, dynamic graphs change over time due to edge insertions or deletions. The key idea is to avoid full recomputation of shortest paths after each change by instead identifying and updating only the affected parts of the SSSP tree.

Key challenges addressed include:

- Efficiently identifying affected subgraphs.
- Parallel updating of SSSP tree while avoiding synchronization bottlenecks.
- Ensuring correctness and convergence to accurate path distances.

The authors design a platform-independent template that is implemented and evaluated using both shared-memory (OpenMP) and GPU (CUDA) parallelism.

2. Key Contributions

1. **Rooted Tree Representation for SSSP**
   a. The SSSP is stored as a tree rooted at the source node.
   b. Maintains distances, parent-child relationships, and affected status for each node.
2. **Two-Step Parallel Update Algorithm**
   a. Step 1: Identify subgraphs affected by edge changes.
   b. Step 2: Iteratively update distances in affected subgraphs until convergence.

3. **Asynchronous and Batch Processing**
    a. Supports asynchronous updates to minimize synchronization.
    b. Batches edge changes to improve load balancing and performance.
4. **Implementation and Results**
    a. Demonstrated on large real and synthetic datasets.
    b. Achieved up to 5.6× speedup over traditional re-computation approaches using OpenMP and CUDA.

3. Proposed Parallelization Strategy

To implement and scale this framework, the following strategy is proposed:

A. MPI (Inter-node Communication)

- Use MPI to distribute graph partitions across compute nodes.
- After local SSSP updates, synchronize affected boundary vertices between nodes.
- Ensure global consistency of distances across partitions.

B. OpenMP (Intra-node Parallelism)

- Use OpenMP within each node to parallelize updates:
    - Process edge insertions and deletions concurrently.
    - Parallelize updates to SSSP distances using dynamic scheduling.
    - Avoid locks by using convergence-based iterative refinement.

C. METIS (Graph Partitioning)

- Use METIS to divide the graph into balanced subgraphs:
    - Reduces inter-node communication.
    - Ensures work is evenly distributed.
    - Maintains edge locality to optimize performance of local SSSP updates.

4. Conclusion

This paper introduces a robust parallel framework for updating SSSP in dynamic graphs, leveraging shared-memory and GPU architectures. With the proposed parallelization strategy using MPI, OpenMP, and METIS, the approach can scale to large distributed systems. This work is highly relevant for dynamic applications like traffic routing, network monitoring, and social network analysis.