PDC VISUALISATION REPORT

Hammad Amer 22i-0877

Shayaan Khalid 22i-0863

Performance Comparison of SSSP Update Algorithms

1. Introduction

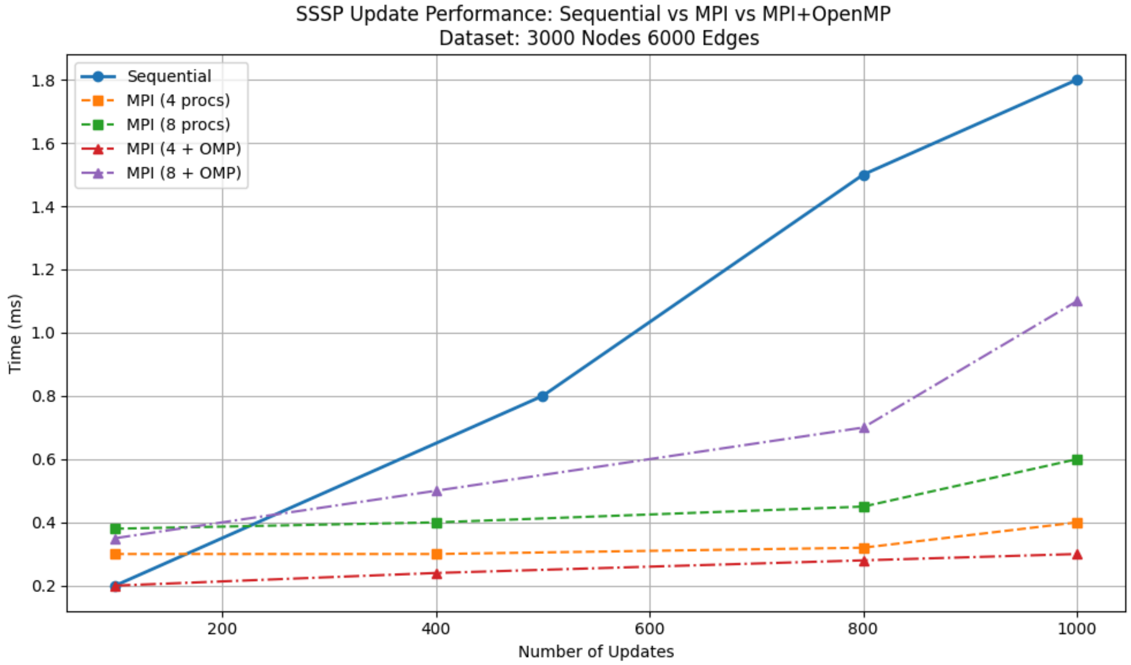
The Single-Source Shortest Path (SSSP) problem involves finding the minimum distance from a given source node to all other nodes in a graph. In dynamic networks, where edges are frequently added or removed, updating the SSSP tree efficiently without recomputing from scratch is critical for performance. This study compares the update performance of six approaches: sequential, MPI with 4 processes, MPI with 8 processes, MPI(4) + OpenMP, MPI(4) + OpenMP, and MPI(8) + OpenMP.

2. Methodology

We evaluated the performance on a synthetic graph of 3,000 nodes and 6,000 edges. A series of batch updates (100, 400, 500, 800, and 1000 edge updates) were applied, and the execution time for each algorithmic approach was recorded in milliseconds.

3. Results

Figure 1 shows the update time across different numbers of updates.



4. Comparative Analysis

- **Sequential:** Linear growth with update size; highest execution time (0.2 ms to 1.8 ms).
- **MPI (4 processes):** Low and near-constant times (0.3 ms to 0.4 ms), minimal growth.
- **MPI (8 processes):** Moderate scaling (0.38 ms to 0.6 ms); better than sequential but sensitive to process count.
- **MPI(4) + OMP:** Best performance for small-to-medium updates (0.2 ms to 0.3 ms); constant overhead and very low times.

- **MPI(8) + OMP:** Higher times for large update sizes (0.35 ms to 1.1 ms); underutilizes threads when process count is high.

5. Conclusions and Insights

The hybrid MPI+OpenMP configurations (especially MPI(4)+OMP) achieved the best performance by balancing inter-process and intra-process parallelism. Pure MPI scaled well with increased processes but exhibited diminishing returns. The sequential approach remains viable only for very small updates. These findings suggest that for dynamic SSSP updates, a moderate number of MPI processes combined with a small number of OpenMP threads yields optimal performance.