Milestone Report

Summary of Work Completed

In our project, we have successfully implemented both coarse-grained and fine-grained versions of a graph and directed graph structure, using an adjacency list approach to represent the directed graphs. This approach involves implementing the adjacency list of a directed graph as a sorted linked list of vertex-nodes, where each vertex-node roots a sorted linked list of edge-nodes. These edge-nodes maintain pointers to the corresponding vertex-nodes, enabling efficient graph traversals.

Coarse-Grained Graph Implementation

Our initial step was the development of a basic graph structure with a coarse-grained locking mechanism. This foundational model is crucial for understanding the intricacies of graph data structures and serves as a baseline for comparing more advanced implementations.

Fine-Grained Graph Implementation

We then progressed to the fine-grained implementation, which is also based on a fine-grained linked list. This implementation offers improved concurrency control by reducing the granularity of locks, thereby allowing for more parallel operations and enhancing overall efficiency in scenarios involving multiple concurrent operations.

Problems

Non-Directed Graph Implementation Complexity

Implementing the non-directed graph in a fine-grained manner has been more challenging and time-consuming than the directed graph. I've decided to focus on the directed graph for now and revisit the non-directed graph later.

Non-Blocking DFS Implementation Challenges

The literature suggests that a non-blocking DFS may require a non-blocking tree structure, which could be complex to implement within our timeframe. I am considering using existing open-source non-blocking tree implementations to integrate with our DFS.

Revised Goals Based on Current Progress

Plan to Achieve

- **Coarse-Grained Graph:** Completed.
- **Fine-Grained Graph:** Completed.
- **Lock-Free Graph:** Prioritize the lock-free directed graph implementation; defer non-directed graph to future work.
- **Efficient Reachability Queries:** Continue as planned for the directed graph versions.
- **Performance and Scalability Analysis:** No changes; continue as planned post lock-free directed graph implementation.

Hope to Achieve

- **Dynamic Graph Optimization:** Continue as planned, with a focus on the directed lock-free graph.
- **Parallel Graph Algorithms Integration:** Focus on simpler parallel algorithms suitable for the directed lock-free graph; complex algorithm integration may be

revisited later.

Schedule

Days Remaining	Primary Goals
Days 1-3	Implementation of Non-Blocking Directed Graph
Days 4-6	Implement Directed Graph Based BFS and DFS
Days 7-9	Implementation of Non-Blocking Graph
Day 10	Make Dynamic Demo for Non-Blocking Graph