Project Milestone Report:
Parallel Union-find

Group members: Zhaowei Zhang, Eric Zhu

Project Website: https://pentene.github.io/parallel-union-find/

Work Completed:

We began our project by implementing two versions of the Union-Find data structure on the CPU: a serial implementation and a coarse-grained locking version. In addition to these, we designed test cases to verify the correctness, ensuring that they correctly handle basic `union` and `find` operations. The project structure has been set up to facilitate additional parallel implementations and future optimizations.

Update to Goals:

We are now confident that we can achieve the goal we set first in our project proposal. We should be able to complete all the goals, even though we had to be hurried for the project. Moreover, when we are using OpenMP to complete the parallel work, we believe that we can also implement the version with MPI in the context of different private addresses, and this would be added to our "nice to haves". Here is an updated list of the goals of our project.

Planned goals:

- Implement multiple parallel implementations on the CPU with OpenMP using:
  - Coarse-grained locking
  - Fine-grained locking
  - Lock-free synchronization
- Implementation of Union-Find optimizations (path compression, union by rank) along with synchronization methods
- Measure and compare throughput, latency, and scalability of each synchronization and optimization method on CPU under diverse workloads, including:
  - Varying graph sizes and densities (sparse, dense)
  - Different access patterns (find-heavy, union-heavy workloads)
  - Real-world datasets

Aspirational goals:

- Investigate dynamic selection of locking vs. lock-free approaches depending on workload size or distribution.
- Demonstrate parallel Union-Find performance improvements in practical applications such as image segmentation or network analysis.
- Implement the data structure using MPI.

Plan for Poster Session:
At the poster session, we plan to present:

- Graphs comparing the throughput and scalability of the three primary parallel implementations (coarse-grained locking, fine-grained locking, and lock-free).
- An analysis of any performance trade-offs and optimization gains.
- A potential demo illustrating the benefits of parallel Union-Find in a real application (time permitting).

Issues:
We still need to finalize or generate comprehensive workload datasets that effectively stress different aspects of parallel implementations. And this would be the key part of the project. Because our proposal was revised and confirmed on 8 April, we need to follow a tighter schedule to ensure that all goals are met.

Updated Schedule (Tentative)

Week	Planned Activities
3.26 - 4.8	Update Proposal
4.9 - 4.15	Implement serial Union-Find and coarse-grained locks. Design test cases to verify correctness.
4.16 - 4.19	Implement fine-grained locks (Eric Zhu) and lock-free Union-Find. (Zhaowei Zhang)
4.20 - 4.22	Evaluate CPU scalability under diverse workloads and evaluate again after optimizations. (Eric Zhu) Perform optimizations to aline with goals. (Zhaowei Zhang)
4.23 - 4.25	Test on real-world workloads and analyze performance. (Eric Zhu) Attempt on implementing using MPI (if possible, Zhaowei Zhang)
4.26 - 4.28	Analyze trade-offs of locking strategies across workloads. (Zhaowei Zhang) Generate figures for performance comparisons. (Eric Zhu)
4.29	Poster session.