Implementation and Analysis of Parallel Graph Algorithms Using MPI/OpenCL, OpenMP, and METIS

This project aims to deepen students' understanding of parallel and distributed computing by engaging them in the implementation of parallel graph algorithms. Students will work in groups of two or three to select, present, and implement one algorithm from a curated list of research papers focusing on parallel solutions to various graph and network-related problems.

You can make group with other sections as long as the other section is being taught by the same instructor. Choose your group wisely. This does not necessarily mean to select group partners with good grades rather make group with students who enjoy spending time with you and are willing to respect your desire to learn and complete a project that fulfills the aim of learning together.

Project Objectives:

1. **Literature Review and Presentation:** Each group will select a research paper from the provided list, study its content, and deliver a comprehensive presentation. This presentation should cover the problem addressed, the proposed parallel algorithm, and the results obtained in the paper.

2. **Implementation:** Groups will implement the parallel algorithm discussed in their chosen paper using MPI and (OpenMP or OpenCL). Given the challenges associated with partitioning graphs for MPI processing, students will utilize the METIS tool for effective graph partitioning. Student will deploy MPI cluster on multiple machines.

3. **Scalability Evaluation:** Students will evaluate the scalability of their implementations by experimenting with multiple publicly available datasets. They will analyze how the performance of their parallel algorithms varies with different data sizes and system configurations. Students will use the MPI performance analyser for this purpose.

4. **Version Control and Progress Tracking:** Each group is required to maintain a GitHub repository to document their progress. This repository should include:

• Incremental commits reflecting the development stages of the project.
• Source code of the implementation.
• Presentation materials.
• Documentation detailing the implementation process, challenges faced, and solutions devised.

Project Phases:

1. **Phase 1 - Presentation: Due 20th April. Presentations will start on 21st**

• Select a research paper from the provided list.
• Conduct an in-depth study of the paper.

• Prepare and deliver a presentation summarizing the key aspects of the paper.

- Present the selected paper's key contributions.
- Propose a parallelization strategy using:
  - MPI (for inter-node communication).
  - OpenMP or OpenCL (for intra-node parallelism).
  - METIS (for graph partitioning).

2. **Phase 2 - Implementation and Demonstration: 4th of May. Demo will start on 5th of May**

• Implement the parallel algorithm using MPI and (OpenMP or OpenCL).
• Employ METIS for graph partitioning to facilitate MPI processing.
• Test the implementation using multiple datasets.
• Evaluate and document the scalability and performance of the algorithm.
• Demonstrate the implementation and discuss findings.

- Implement the algorithm on **public datasets**
- Compare performance: **Sequential vs. MPI vs. MPI+(OpenMP/OpenCL)**.
- Demo:
  - Show **speedup/scalability** (weak/strong scaling).
  - Visualize results (e.g., execution time, partitioning efficiency).

Research Papers for Selection:

Choose **one** of the following papers:

1. A Parallel Algorithm for Updating a Multi-objective Shortest Pathin Large Dynamic Networks https://dl.acm.org/doi/pdf/10.1145/3624062.3625134
2. Parallel social behavior-based algorithm for identification of influential users in social network https://drive.google.com/file/d/1vp5he-8ogdPJRFPiT6BDGPVZoH9CEnAh/view?usp=sharing
3. A Parallel Algorithm Template for Updating Single-Source Shortest Paths in Large-Scale Dynamic Networks https://drive.google.com/file/d/1Cj7u6bLfbwfjSwtZhfDwv4V5wIuiGB9y/view?usp=sharing
4. Parallel Algorithms for Butterfly Computations https://epubs.siam.org/doi/pdf/10.1137/1.9781611976021.2
5. A community detection-based parallel algorithm for quantum circuit simulation using tensor networks (GPU based solution) https://link.springer.com/content/pdf/10.1007/s11227-025-06918-3.pdf
6. A parallel algorithm for constructing multiple independent spanning trees in bubble-sort networks https://drive.google.com/file/d/1leYyetxK3SqK8abssBw3QsNsD-T_Xwn_/view?usp=sharing

Special Reference: This can be used to identify the strategies for parallelism in graphs

Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable https://dl.acm.org/doi/pdf/10.1145/3434393

Tools and Resources:

• **METIS:** A software package for partitioning unstructured graphs and computing fill-reducing orderings of sparse matrices. METIS will be used to partition graphs to enhance the efficiency of MPI-based parallel processing. https://www.lrz.de/services/software/mathematik/metis/
• **MPI:** A standardized and portable message-passing system designed to function on parallel computing architectures.
• **OpenMP:** An application programming interface that supports multi-platform shared memory multiprocessing programming.
• **OpenCL**

Assessment Criteria:

• Depth of understanding demonstrated in the presentation.
• Correctness and efficiency of the implemented algorithm.
• Effective use of METIS for graph partitioning.
• Comprehensive scalability analysis.
• Quality and consistency of documentation and version control in the GitHub repository. Periodic commits must be there to show evidence of gradual progress of the project.

Submission Guidelines:

• All deliverables, including presentations, source code, datasets used, and documentation, must be submitted via the group's GitHub repository.

• Ensure that the repository is well-organized and that all commits are properly documented to reflect the project's progression.