PARALLELIZING SORTING ALGORITHM

Semester 5

OCTOBER 10, 2023

FAST NUCES

TABLE OF CONTENT

PROJECT PROPSAL

Group Members

21K-3212 Ahmed Ali

21K-3211 Shayan Haider

21K-4680 Taha Hassan

Introduction:

In the realm of computer science and data processing, the efficient sorting of large data sets is a fundamental operation with a wide range of applications, from database management to scientific simulations. Sorting algorithms are vital components in numerous computing tasks, and as the volume of data continues to grow, the need for optimizing these algorithms becomes increasingly significant. This project, "Comparative Analysis of Parallelized Sorting Algorithms for Large Data Sets," aims to address this need by examining ten different sorting algorithms and their parallelized implementations in the context of large data sets.

Project Objectives:

1. **Algorithm Selection**: We will explore ten different sorting algorithms, including classic methods like Quicksort, Mergesort, and Heapsort, as well as modern variations and hybrid algorithms.

2. **Parallelization Techniques**: Our study will focus on parallelizing these algorithms using various techniques, such as multi-threading, distributed computing, and GPU acceleration.

3. **Performance Metrics**: To evaluate the algorithms, we will use a comprehensive set of performance metrics, including execution time, memory usage, scalability, and stability under different data distributions.

4. **Large Data Sets**: The project will emphasize the sorting of large data sets, ranging from KB to MB, to represent real-world big data scenarios.

5. **Comparative Analysis**: We will conduct a detailed comparative analysis to highlight the strengths and weaknesses of each sorting algorithm and its parallelized version.

6. **Optimization Strategies**: We will explore potential optimization strategies for each algorithm and assess their impact on performance and efficiency.

7. **Real-World Applications**: The project will discuss how the findings of this research can be applied to real-world applications, such as big data processing, database management, and scientific simulations.

8. **Open-Source Implementations**: As part of the project, open-source implementations of parallelized sorting algorithms will be provided to benefit the broader development community.

Significance:

Efficient sorting algorithms are essential for a wide array of industries, from finance to scientific research, as they directly impact data processing and decision-making times. By analyzing and optimizing these algorithms for large data sets, this project seeks to contribute to the field of high-performance computing, making data processing more efficient and accessible for various applications.

Expected Outcomes:

The project's expected outcomes include a comprehensive comparative analysis report, open-source code repositories for the parallelized sorting algorithms, and recommendations for selecting the most suitable sorting algorithm for specific data processing tasks.

Scope:

This project will focus on parallelized sorting algorithms and their performance in the context of large data sets, excluding other data processing tasks. It will provide valuable insights into the efficiency and scalability of sorting algorithms and their practical applications in modern computing environments.

Tools and Technologies:

1. C/C++ Development Tools: Ubuntu provides a robust development environment for C/C++ programming. You can use GCC (GNU Compiler Collection) for compiling your C code.

2. Parallel Programming Libraries:

- POSIX Threads (pthread): For multi-threading in C.

- OpenMP: An API for parallel programming that simplifies multi-threading.

- MPI (Message Passing Interface): For distributed computing across multiple nodes or machines.