CSDS 438 Semester Long Project Proposal (Group 10)

CPU vs GPU performance comparison of sorting algorithms (OpenMP + CUDA)

Abstract

This research project focuses on conducting a comprehensive performance evaluation of five non-comparison sorting algorithms—Radix Sort, Counting Sort, Bucket Sort, Pigeonhole Sort, and Bitonic Sort—across different programming paradigms, specifically comparing CPU-based implementations with GPU-accelerated CUDA implementations. The study utilizes a carefully controlled synthetic dataset of 32-bit unsigned integers with five distinct data distributions (uniform random, Gaussian, exponential, sorted, and reverse sorted) and varying dataset sizes from thousands to millions of elements. By implementing each algorithm in both sequential/parallel CPU versions using C++ and OpenMP, as well as parallel GPU versions using CUDA C++, the research aims to provide empirical evidence for optimal algorithm selection based on data characteristics, hardware architecture, and computational requirements.

The project employs rigorous experimental methodology with comprehensive performance metrics including execution time, memory usage, GPU utilization, and scalability analysis across different dataset sizes and distributions. The research will generate quantitative comparisons revealing how non-comparison sorting algorithms perform across modern CPU and GPU architectures, identifying performance bottlenecks, optimal algorithm-paradigm combinations, and providing practical guidelines for practitioners working with large-scale data sorting tasks. The expected outcomes include open-source implementations, benchmark datasets for future research, and peer-reviewed publications that will contribute valuable insights to the high-performance computing community regarding the effectiveness of parallel processing paradigms for fundamental algorithmic operations.