Parallel and Distributed Computing (CS428)

Project Proposal

Course Instructor: Sir Anees

Group Members:

1. Hafsa Iftikhar 21B-028-CS
2. Abdul Moiz Noman 21B-046-CS
3. Asad Javed 21B-082-CS
4. Hiba Shakeel 21B-175-CS

Project Title:
Parallel Quicksort Implementation in Python

Objective:
The primary objective of this project is to implement the Quicksort algorithm both serially and in parallel using Python. Initially, the serial Quicksort implementation will be used to sort a sequence of integers provided in a text file. The second phase will involve parallelizing the Quicksort algorithm to enhance sorting performance.

Problem Statement:
Quicksort is an efficient and widely used sorting algorithm, but its performance can be further improved by parallelizing it in distributed systems. This project aims to explore how to adapt the Quicksort algorithm for parallel execution using Python. The challenge is to implement the parallel version effectively and handle the complexities of inter-process communication in a distributed environment.

Solution:

1. **Serial Quicksort Implementation:**
    - The first part of the project involves implementing the serial Quicksort algorithm in Python. This algorithm sorts a sequence of integers from a text file using the divide-and-conquer approach. The array is partitioned around a pivot, and each partition is recursively sorted.

2. **Parallel Quicksort Implementation:**
    - The parallel Quicksort will involve partitioning the array, sorting each part in parallel on different processors, and merging the results back into a single sorted sequence.

3. **Optimization and Load Balancing:**
    - Performance optimizations will be applied, such as minimizing inter-process communication and maximizing the efficiency of the sorting operation across multiple processors.

Tools and Technologies:

- **Programming Language:** Python
- **Input Format:** Text file containing an unsorted sequence of integers
- **Output Format:** Sorted sequence of integers
- **Compilers/IDEs:** Python 3.x, Jupyter Notebook/VS Code

Expected Outcome:

- A fully functional serial version of the Quicksort algorithm implemented in Python, capable of sorting integers from a text file.
- A parallelized version of Quicksort to distribute the data across multiple processors.
- Performance improvements in sorting time for larger datasets with the parallel version, showing how the number of processors affects performance.
- Insights into the challenges of parallelizing sorting algorithms, with a focus on inter-process communication and synchronization in distributed computing environments.

Conclusion:

This project will provide a comprehensive understanding of both serial and parallel versions of the Quicksort algorithm implemented in Python. The project will highlight the importance of optimizing sorting algorithms in parallel environments and demonstrate techniques for efficient communication across multiple processors.