PARALLEL PROGRAMMING PARADIGM WITH C++ AND RUST

Overview

In a multiprocessor system utilizing Single Instruction, Multiple Data (SIMD) architecture, data parallelism occurs when each processor simultaneously executes identical tasks on distinct distributed data sets. This parallelization strategy involves spreading data across various nodes, where each node operates on its designated data concurrently. Data parallelism is particularly effective for regular data structures such as arrays and matrices, enabling parallel processing of individual elements.

Goals

1. Our goal would be focused on extracting parallel programming features from the languages Rust and C++ and compare the serial and parallel execution time accordingly.
2. We would also analyze the differences in performance achieved through existing directories for parallel computation in each of the languages namely Rayon for Rust and OpenMP for C++.

Specifications

We would gauge the performance of the languages through the implementation of different algorithms like mergesort in both languages.

Group Members

Ashray Kashyap	2020B3A70494G	f20200494@goa.bits-pilani.ac.in
Naman Markhedkar	2020B5A71862G	f20201862@goa.bits-pilani.ac.in
Mohammed Aman	2020B3A70607G	f20200607@goa.bits-pilani.ac.in
Saahir Vaidya	2020B3A71142G	f20201142@goa.bits-pilani.ac.in