### **Lab 9:** **Performance in Rust**

**Exercise: Performance Optimization**

In this exercise, We will work on a simple Rust program to calculate the sum of squares for a range of numbers. We will apply performance optimizations and benchmark the different implementations to compare their performance.

1. Create a new Rust project using cargo:
2. Open your terminal/command prompt and run the following command:

cargo new performance\_optimization

cd performance\_optimization

Open the main.rs file in the src directory of your project. You can use any code editor for this.

Implement three functions to calculate the sum of squares for a given range of numbers:

b. Parallel Sum of Squares using Rayon (requires adding rayon as a dependency):

Add the following to your Cargo.toml:

[dependencies]

rayon = "1.5"

In the main function, call these three functions for a range of numbers and measure their execution time using the std::time::Instant:

use rayon::prelude::\*;

use std::time::Instant;

fn sum\_of\_squares\_basic(start: u32, end: u32) -> u64 {

    let mut sum = 0;

    for num in start..=end {

        sum += (num \* num) as u64;

    }

    sum

}

fn sum\_of\_squares\_parallel(start: u32, end: u32) -> u64 {

    (start..=end).into\_par\_iter().map(|num| (num \* num) as u64).sum()

}

fn main() {

    let start = 1;

    let end = 1\_000\_000;

    let start\_time = Instant::now();

    let result\_basic = sum\_of\_squares\_basic(start, end);

    let elapsed\_time\_basic = start\_time.elapsed();

    let start\_time = Instant::now();

    let result\_parallel = sum\_of\_squares\_parallel(start, end);

    let elapsed\_time\_parallel = start\_time.elapsed();

    println!("Result (Basic): {}", result\_basic);

    println!("Result (Parallel): {}", result\_parallel);

    println!("Time taken (Basic): {:?}", elapsed\_time\_basic);

    println!("Time taken (Parallel): {:?}", elapsed\_time\_parallel);

}

1. Save the file and return to your terminal/command prompt.
2. Build and run your program using cargo run --release:

The --release flag enables optimizations when building, improving performance.

cargo run --release

The program will calculate the sum of squares for the given range of numbers using three different implementations and display their results and execution times.

**Happy coding!**