НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №2

із дисципліни «Розподілені і хмарні обчислення»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-03 | Ліскін В. О. |
| Шаповалов Г. Г. |  |

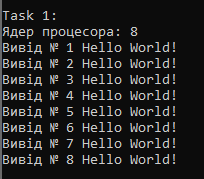
Київ — 2023

Мета роботи

Порівняти однопоточну та багатопоточну версії матричного множення та додавання елементів матриці

Опис програми

Завдання 1 – вивести стільки разів ‘привіт світ!’ скільки ядер має процесор



Завдання 2 – дослідити час для множення та додавання матриць у одно/багатопоточному режимах:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Потоків | Розмірність матриці | Кращий час | Середній час | Гірший час |
| Множення матриць | 1 | 150 | 12.010 ms | 12.432 ms | 12.903 ms |
| 8 | 150 | 3.5382 ms | 3.5930 ms | 3.6572 ms |
| Додавання матриць | 1 | 1000 | 4.7238 ms | 4.8649 ms | 5.0323 ms |
| 8 | 1000 | 1.5417 ms | 1.5542 ms | 1.5701 ms |

Багатопоточна реалізація дає приріст у часі.

Лістинг програми:

use rayon::{current\_num\_threads};

use std::process::Command;

fn main() {

    task\_1();

    task\_2();

}

fn task\_1() {

    println!("\nTask 1:");

    let num\_cores: usize = current\_num\_threads();

    println!("Ядер процесора: {}", num\_cores);

    for i in 0..num\_cores {

        let n = i + 1;

        println!("Вивід № {} Hello World!", n);

    }

}

fn task\_2() {

    println!("\nTask 2:");

    match Command::new("cargo").args(&["bench"]).status() {

        Ok(status) => if status.success() {

            println!(" ");

        } else {

            eprintln!("cargo bench завершилося з помилкою");

        },

        Err(e) => eprintln!("Помилка при запуску cargo bench: {}", e),

    }

}