

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із лабораторної роботи №3
із дисципліни «Розподілені і хмарні обчислення»

Виконав:

студент групи КМ-03

Шаповалов Г. Г.

Керівник:

Ліскін В. О.

Київ — 2023

Мета роботи

Проінтегрувати методом редукції.

Проінтегрувати методом правих прямокутників за різної кількості потоків та інтервалів розбиття.

$$\int_0^1 \ln(t + 1) dt$$

Опис програми

Завдання 1 – порахуємо інтеграл в отопоточному режимі для визначення його значення. Кількість інтервалів розбиття = 10000

```
Task 1:  
f = ln(t + 1), a=0, b=1, steps=10000  
Result: 0.38629436111989074
```

Завдання 2 – змінюємо інтервали розбиття, кількість потоків та порівнюємо час

Інтервали розбиття	Потоки	Кращий час виконання	Середній час виконання	Гірший час виконання
10	1	1.8464 μ s	1.8943 μ s	1.9565 μ s
10	2	1.9286 μ s	2.1238 μ s	2.3294 μ s
10	4	2.4385 μ s	2.4856 μ s	2.5377 μ s
10	8	3.6720 μ s	3.7353 μ s	3.8050 μ s
10^3	1	60.605 μ s	63.460 μ s	67.404 μ s
10^3	2	33.189 μ s	34.054 μ s	35.143 μ s
10^3	4	15.294 μ s	15.617 μ s	15.996 μ s
10^3	8	23.131 μ s	23.453 μ s	23.770 μ s
10^6	1	41.794 ms	45.173 ms	48.580 ms
10^6	2	26.001 ms	26.126 ms	26.262 ms
10^6	4	6.8540 ms	7.1791 ms	7.5387 ms
10^6	8	3.4796 ms	3.5489 ms	3.6233 ms

Лістинг програми:

```
use std::{sync::Arc};  
use lab_3::integration::integral_reduction;  
use std::process::Command;  
  
fn main() {
```

```

    testing();
    task_1();
    task_2();
}

fn task_1() {
    println!("\n\n\nTask 1:");
    let f = |t: f64| (t + 1.0).ln();
    let a = 0.0;
    let b = 1.0;
    let steps = 10000;
    println!("f = ln(t + 1), a={}, b={}, steps={}", a, b, steps);

    let af = Arc::new(f);

    let res = integral_reduction(&af, a, b, steps);
    println!("Result: {}", res);
}

fn task_2() {
    println!("\n\n\nTask 2:");
    println!("Запускаємо cargo bench");
    match Command::new("cargo").args(&["bench"]).status() {
        Ok(status) => if status.success() {
            println!("cargo bench виконано успішно");
        } else {
            eprintln!("cargo bench завершилося з помилкою");
        },
        Err(e) => eprintln!("Помилка при запуску cargo bench: {}", e),
    }
}

fn testing() {
    println!("\nЗапускаємо cargo test");
    match Command::new("cargo").args(&["test"]).status() {
        Ok(status) => if status.success() {
            println!("cargo test виконано успішно");
        } else {
            eprintln!("cargo test завершилося з помилкою");
        },
        Err(e) => eprintln!("Помилка при запуску cargo test: {}", e),
    }
}

```