

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-01  
Скорденко Д. О.

Керівник:  
доцент кафедри ПМА  
Ліскін В. О.

## ЗМІСТ

Вступ .....	3
1 Основна частина .....	4
2 Опис програми [Тестовий приклад] .....	5
Висновки .....	7
Додаток Код лістинги .....	8

## ВСТУП

**Мета:** порівняти інтегрування методом редукції за різної к-сті потоків.

Дослідний приклад:

$$\left\{ \begin{array}{l} f = \frac{1}{\sin^2(2x)} \\ a = 0 \\ b = \pi/2 \end{array} \right.$$

## 1 ОСНОВНА ЧАСТИНА

**Опис програми:** Для реалізації паралелізму буде використовуватись 'Rayon'. Для порівняння швидкості обчислень буде використовуватись 'Criterion'.

Порівняння буде проведено на різних  $k$ -стях відрізків  $n \in [64, 1e5, 1e7]$ , та при різній  $k$ -сті потоків  $nworkers \in [1, 2, 4, 8]$



Таблиця 2.1 – Порівняння швидкодії

n	nworkers	time		
64	1	8.0789 $\mu$ s	8.2451 $\mu$ s	8.4262 $\mu$ s
64	2	12.358 $\mu$ s	12.759 $\mu$ s	13.170 $\mu$ s
64	4	18.333 $\mu$ s	18.765 $\mu$ s	19.234 $\mu$ s
64	8	29.844 $\mu$ s	30.635 $\mu$ s	31.469 $\mu$ s
1e5	1	1.7601 ms	1.7908 ms	1.8307 ms
1e5	2	912.96 $\mu$ s	923.10 $\mu$ s	934.10 $\mu$ s
1e5	4	542.83 $\mu$ s	560.27 $\mu$ s	578.18 $\mu$ s
1e5	8	484.52 $\mu$ s	519.02 $\mu$ s	575.76 $\mu$ s
1e7	1	165.89 ms	167.28 ms	168.80 ms
1e7	2	84.522 ms	85.443 ms	86.488 ms
1e7	4	43.958 ms	44.848 ms	45.868 ms
1e7	8	36.668 ms	37.263 ms	37.933 ms

## ВИСНОВКИ

На малих об'ємах обчислень збільшення к-сті потоків призводить до погіршення продуктивності.

На більших об'ємах збільшення потоків призводить до збільшення продуктивності, однак після певної к-сті потоків ефект покращення продуктивності стає незначним.

## Додаток

## Код лістинги

\*Примітка: У код лістингах при копіюванні втрачається форматування (не копіюються пробіли). Файли прикріплено до цього pdf (вкладка "прикріплені файли").

## Listing 1: integration.rs

```
use std::sync::{Arc};
use std::marker::{Sync, Send};
use rayon::prelude::*;

pub fn integral_reduction(f: &Arc<impl Fn(f64) → f64 + Sync + Send>, a: f64, b: f64) -> f64 {
    let dx = (b - a) / steps as f64;

    (0..steps).into_par_iter().map(move |i| {
        let i = i as f64;
        let x = a + i * dx;

        let function = f(x);
        function * dx
    }).reduce(|| 0.0, |a,b| a + b)
}
```



```

#[cfg(test)]
mod tests {
    use super::integral_reduction;
    use std::sync::Arc;
    use std::f64::{consts::PI, INFINITY};

    #[test]
    fn test_example_sample() {
        println!("Task 1");
        let f = |x: f64| 1.0 / (f64::sin(2.0*x)).powi(2);
        let a = 0.0;
        let b = PI/2.0;
        let steps = 64;

        let af = Arc::new(f);

        let res = integral_reduction(&af, a, b, steps);
        assert_eq!(res, INFINITY)
    }
}

```

Listing 2: lib.rs

```
pub mod integration;
```

Listing 3: main.rs

```

use std::{sync::Arc, f64::consts::PI};
use lab_3::integration::integral_reduction;

```

```
fn task_1() {  
    println!("Task 1");  
    let f = |x: f64| 1.0 / (f64::sin(2.0*x)).powi(2);  
    let a = 0.0;  
    let b = PI/2.0;  
    let steps = 64;  
  
    let af = Arc::new(f);  
  
    let res = integral_reduction(&af, a, b, steps);  
    println!("Result: {}", res);  
}  
  
fn main() {  
    task_1();  
}
```