**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики

Кафедра програмування

Звіт про виконання

лабораторної роботи №2

**Розпаралелення множення матриць**

з курсу “Паралельні та розподілені обчислення”

Виконав:

Студент групи ПМО-21с

Заяц Ростислав Васильович

Перевірив:

Гошко Богдан Мирославович

**Львів – 2025**

**Тема:** Написання програми множення матриць з використанням паралельного та послідовного програмування

**Мета:** Закріпити знання роботи паралельних алгоритмів з матрицею, практично використати в своїй лабораторній роботі паралельні алгоритми.

**Теоретична частина**

**Прискорення (Speedup)**

Визначення:  
 Прискорення — це відношення часу виконання послідовного алгоритму до часу виконання паралельного алгоритму. Воно показує, наскільки швидше працює паралельна версія в порівнянні з послідовною.

Формула:

**Speedup = T\_посл / T\_парал**

Де:  
- T\_посл — час виконання послідовного алгоритму;  
- T\_парал — час виконання паралельного алгоритму.

**Ефективність (Efficiency)**

Визначення:  
 Ефективність показує, наскільки ефективно використовуються потоки (чи процесори) при паралельному виконанні. Це відношення прискорення до кількості потоків.

Формула:

**Efficiency = (Speedup / k) \* 100%**

Де:  
- Speedup — прискорення;  
- k — кількість потоків (процесорів).

**Хід роботи.**

Взявши за основу програму з попередньої лабораторної, я удосконалив її, додавши до класу **Matrix,** перевизначений оператор множення, щоб використати його в своїх послідовних обчисленнях. Пізніше я написав функцію **multMatriececParts**. Вона приймає в аргументи матрицю А та Б, також результуючу матрицю, куди вносить результати обчислень і початковий та кінцевий рядок для обчислень. Саме обчислення починається після перевірки рівності кількості колонок матриці А і кількості рядків матриці Б. У коді реалізовано множення матриць методом трьох вкладених циклів. Для кожного елемента результуючої матриці обчислюється скалярний добуток відповідного рядка з матриці A та стовпця з матриці B. Обчислення обмежено діапазоном рядків (startRow до endRow), що дозволяє ефективно реалізувати паралельну обробку. Паралельність було реалізовано ідентично як у попередній лабораторній:

Ділимо кількість рядків на кількість потоків, а залишок, який не перевищує step, віддаємо останньому потоку на обробку.

Для обчислення часу використав бібліотеку ctime, порівняв результати на різних розмірностях матриці та створив таблицю:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Matrix size | Threads | Sequential time (s) | Parallel time (s) | Speedup | Efficiency (%) | Performance gain (s) |
| 100x100 | 2 | 0.040 | 0.049 | 0.82 | 40.82 | -0.009 |
| 100x100 | 4 | 0.028 | 0.147 | 0.19 | 4.76 | -0.119 |
| 100x100 | 6 | 0.027 | 0.022 | 1.23 | 20.45 | 0.005 |
| 100x100 | 8 | 0.021 | 0.014 | 1.50 | 18.75 | 0.007 |
| 500x500 | 2 | 3.122 | 2.391 | 1.31 | 65.29 | 0.731 |
| 500x500 | 4 | 2.916 | 1.222 | 2.39 | 59.66 | 1.694 |
| 500x500 | 6 | 3.163 | 1.530 | 2.07 | 34.46 | 1.633 |
| 500x500 | 8 | 2.577 | 1.216 | 2.12 | 26.49 | 1.361 |
| 1000x1000 | 2 | 19.614 | 14.05 | 1.40 | 69.80 | 5.564 |
| 1000x1000 | 4 | 20.321 | 9.209 | 2.21 | 55.17 | 11.112 |
| 1000x1000 | 6 | 18.208 | 9.154 | 1.99 | 33.15 | 9.054 |
| 1000x1000 | 8 | 21.659 | 9.329 | 2.32 | 29.02 | 12.33 |
| 2500x2500 | 2 | 282.537 | 140.000 | 2.02 | 101.0 | 142.537 |
| 2500x2500 | 4 | 263.559 | 142.730 | 1.85 | 46.16 | 120.829 |
| 2500x2500 | 6 | 252.945 | 134.391 | 1.88 | 31.37 | 118.554 |
| 2500x2500 | 8 | 258.060 | 126.726 | 2.04 | 25.45 | 131.334 |

**Висновок:** Виконуючи цю лабораторну роботу, я закріпив свої знання роботи паралельних алгоритмів з матрицями. Аналізуючи таблицю хочу зазначити що краще використовувати паралельне обчислення множення матриці з розмірами >= [1000]x[1000], також ми бачимо як стрімко падає ефективність при збільшені потоків. Найоптимальнішим варіантом, я вважаю використання 4-ох потоків для обчислень адже, так ми отримуємо відносно найкращу ефективність та прискорення.