

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра програмування

**Паралельні та розподілені обчислення**

**ЛАБОРАТОРНА РОБОТА №2**

**Тема: «Множення матриць»**

Виконав:

*Ст. Лук'янчук Денис*

Група ПМІ-33

2025

**Тема:** «Множення матриць»

**Мета роботи:** Послідовно та паралельно виконати множення матриць, дослідити прискорення та ефективність паралельних обчислень.

## **Теоретичний матеріал**

**Послідовні обчислення** – це спосіб виконання програми, при якому всі інструкції виконуються одна за одною в одному потоці управління. У випадку матричних операцій це означає, що кожен елемент матриці обробляється по черзі, незалежно від інших.

### **Переваги:**

- Простота реалізації;
- Відсутність накладних витрат на синхронізацію;
- Передбачуваність результату.

### **Недоліки:**

- Використовується лише одне ядро процесора, інші залишаються незадіяними;
- Час виконання зростає пропорційно до обсягу задачі.

**Паралельні обчислення** – це спосіб організації виконання програми, при якому завдання ділиться на підзадачі та виконується одночасно кількома потоками або процесами. У задачах обробки матриць це означає, що кожен потік може обчислювати окремі рядки чи блоки матриці.

### **Переваги:**

- Використання кількох ядер процесора, що зменшує час обчислень;
- Підвищення продуктивності для великих задач;
- Можливість масштабування на багатопроцесорних системах.

### **Недоліки:**

- Необхідність додаткової синхронізації між потоками;
- Накладні витрати на створення потоків;
- Для малих задач паралельний алгоритм може працювати повільніше, ніж послідовний.

**Прискоренням** називається відношення часу виконання послідовного алгоритму до часу виконання паралельного алгоритму на  $k$  потоках:

$$S = \frac{T_{seq}}{T_{par}}$$

де:

- $T_{seq}$  – час виконання послідовного алгоритму;
- $T_{par}$  – час виконання паралельного алгоритму.

**Інтерпретація:** якщо  $S = 3$ , це означає, що паралельний алгоритм виконується утричі швидше, ніж послідовний.

**Ефективність** показує, наскільки добре використані ресурси (потоки або ядра) при паралельному обчисленні:

$$E = \frac{S}{k} \cdot 100\%$$

де  $k$  – кількість потоків.

- Якщо  $E \approx 100\%$ , це означає, що ресурси використовуються максимально ефективно.
- Якщо  $E < 100\%$ , це свідчить про втрати продуктивності, які можуть бути викликані накладними витратами на керування потоками, синхронізацію або нерівномірний розподіл навантаження.

## Хід роботи

Під час виконання лабораторної роботи я реалізував обробку матриць двома способами: послідовним і паралельним. У програмі я реалізував самостійне введення розміру матриці (кількість рядків та стовпців) і кількість потоків, на які будуть розподілені обчислення.

У випадку, якщо кількість рядків не ділиться на кількість потоків, неможливо рівномірно розподілити роботу між усіма потоками, тому спочатку обчислюється базова кількість рядків, яку отримує кожен потік. Потім визначається залишок рядків, які не ввійшли в цей рівномірний розподіл. Першим кільком потокам видається по одному додатковому рядку, щоб усі рядки матриці були оброблені.

Для початку я виконав множення матриць розміром  $100 \times 100$  з використанням 4, 8 та 16 потоків:

Матриця A:  $100 \times 100$   
Матриця B:  $100 \times 100$   
Кількість потоків: 4

Послідовне множення завершено за 3 мс (0,00 с)  
Паралельне множення завершено за 2 мс (0,00 с)

Прискорення: 1,50  
Ефективність: 37,50%

Матриця A:  $100 \times 100$   
Матриця B:  $100 \times 100$   
Кількість потоків: 8

Послідовне множення завершено за 3 мс (0,00 с)  
Паралельне множення завершено за 2 мс (0,00 с)

Прискорення: 1,50  
Ефективність: 18,75%

Матриця A:  $100 \times 100$   
Матриця B:  $100 \times 100$   
Кількість потоків: 16

Послідовне множення завершено за 3 мс (0,00 с)  
Паралельне множення завершено за 3 мс (0,00 с)

Прискорення: 1,00  
Ефективність: 6,25%

При малих розмірах матриць швидкість виконання послідовного та паралельного множення практично однакова, незалежно від кількості використаних потоків. Відповідно, прискорення також не спостерігається. Проте зі збільшенням кількості потоків ефективність роботи паралельного алгоритму зменшується через додаткові витрати на керування потоками.

Далі я продовжив збільшувати розмірність матриці до 500 на 500:

```
Матриця A: 500x500
Матриця B: 500x500
Кількість потоків: 4

Послідовне множення завершено за 428 мс (0,43 с)
Паралельне множення завершено за 144 мс (0,14 с)

Прискорення: 2,97
Ефективність: 74,31%
```

```
Матриця A: 500x500
Матриця B: 500x500
Кількість потоків: 8

Послідовне множення завершено за 429 мс (0,43 с)
Паралельне множення завершено за 115 мс (0,12 с)

Прискорення: 3,73
Ефективність: 46,63%
```

```
Матриця A: 500x500
Матриця B: 500x500
Кількість потоків: 16

Послідовне множення завершено за 448 мс (0,45 с)
Паралельне множення завершено за 90 мс (0,09 с)

Прискорення: 4,98
Ефективність: 31,11%
```

При збільшенні розміру матриць паралельне множення працює швидше за послідовне. Зі збільшенням кількості потоків швидкість обчислень та прискорення зростають, проте ефективність знижується через накладні витрати на керування потоками. Тому, як і в попередньому прикладі, найефективнішим залишається варіант із меншою кількістю потоків.

У наступних прикладах спостерігається аналогічна тенденція: зі збільшенням кількості потоків швидкість виконання зростає, а прискорення підвищується, проте ефективність зменшується.

Матриця A: 1000x1000 Матриця B: 1000x1000 Кількість потоків: 4  Послідовне множення завершено за 3496 мс (3,50 с) Паралельне множення завершено за 1190 мс (1,19 с)  Прискорення: 2,94 Ефективність: 73,45%	Матриця A: 2000x2000 Матриця B: 2000x2000 Кількість потоків: 4  Послідовне множення завершено за 32600 мс (32,60 с) Паралельне множення завершено за 11284 мс (11,28 с)  Прискорення: 2,89 Ефективність: 72,23%
Матриця A: 1000x1000 Матриця B: 1000x1000 Кількість потоків: 8  Послідовне множення завершено за 3490 мс (3,49 с) Паралельне множення завершено за 757 мс (0,76 с)  Прискорення: 4,61 Ефективність: 57,63%	Матриця A: 2000x2000 Матриця B: 2000x2000 Кількість потоків: 8  Послідовне множення завершено за 33341 мс (33,34 с) Паралельне множення завершено за 7184 мс (7,18 с)  Прискорення: 4,64 Ефективність: 58,01%
Матриця A: 1000x1000 Матриця B: 1000x1000 Кількість потоків: 16  Послідовне множення завершено за 3509 мс (3,51 с) Паралельне множення завершено за 675 мс (0,68 с)  Прискорення: 5,20 Ефективність: 32,49%	Матриця A: 2000x2000 Матриця B: 2000x2000 Кількість потоків: 16  Послідовне множення завершено за 33631 мс (33,63 с) Паралельне множення завершено за 6189 мс (6,19 с)  Прискорення: 5,43 Ефективність: 33,96%

Але якщо вказати кількість потоків більшу, ніж фізично доступно на комп'ютері, швидкість і прискорення вже не зростатимуть, а ефективність значно знизиться через додаткові накладні витрати на керування потоками. Наприклад, у мене на ноутбучі 16 потоків, і при спробі запуску з 24 потоками ефективність падає з 32 % до 20 %.

```
Матриця A: 1000x1000  
Матриця B: 1000x1000  
Кількість потоків: 16
```

```
Послідовне множення завершено за 3509 мс (3,51 с)  
Паралельне множення завершено за 675 мс (0,68 с)
```

```
Прискорення: 5,20  
Ефективність: 32,49%
```

```
Матриця A: 1000x1000  
Матриця B: 1000x1000  
Кількість потоків: 24
```

```
Послідовне множення завершено за 3517 мс (3,52 с)  
Паралельне множення завершено за 702 мс (0,70 с)
```

```
Прискорення: 5,01  
Ефективність: 20,87%
```

### Підсумок:

- При малих матрицях послідовне та паралельне множення працюють приблизно однаково, незалежно від кількості потоків, тому прискорення практично не спостерігається.
- Зі збільшенням розмірів матриць паралельний алгоритм стає швидшим за послідовний, і прискорення зростає.
- Зі збільшенням кількості потоків швидкість виконання та прискорення зростають, але ефективність трохи знижується через накладні витрати на керування потоками.
- Найефективнішим залишається використання кількості потоків, яка відповідає або трохи менша за фізичну кількість ядер процесора.
- Якщо вказати кількість потоків більшу, ніж доступно на комп'ютері, швидкість і прискорення більше не зростають, а ефективність різко падає.

**Висновок:** У лабораторній роботі я навчився реалізовувати обчислення добутку двох матриць послідовним та паралельним алгоритмами. Було проведено експерименти з різними розмірами матриць і кількістю потоків. Встановлено, що паралельний алгоритм значно прискорює обчислення на великих матрицях, особливо якщо кількість потоків відповідає фізичній кількості ядер процесора. Зі збільшенням кількості потоків ефективність зменшується через накладні витрати на керування потоками, а при перевищенні кількості фізичних потоків швидкість і прискорення вже не зростають. Отже, для оптимального використання паралельних обчислень важливо враховувати розмір матриць та кількість доступних апаратних потоків.