НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Звіти до комп’ютерних практикумів з кредитного модуля

“Parallel computing”

**Виконав**

**Студенти групи ІТ-04**

**Гавриленко Я.С**

**Перевірив:**

Київ – 2022

**Комп‘ютерний практикум No 6**

**«Розробка паралельного алгоритму множення матриць з використанням МРІ-методів обміну повідомленнями «один-до-одного» та дослідження його ефективності»**

ЗАВДАННЯ

Зображення, що містить текст, Шрифт, знімок екрана, ряд

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт, інформація

Автоматично згенерований опис

**ВИКОНАННЯ**

Повний код лабораторної: <https://github.com/yan14171/parralel_calcullation_labs/tree/master/lab>6

**Завдання 1**

Методи синхроннізованого пересилання даних Send та Recv представляють собою блокуючі методи, які зупиняють роботу потоку до моменту надсилання даних до буферу обміну.

У свою чергу, функції Isend та Irecv мають неблокуючий характер, так після виконання методу повертається об’єкт типу Request. Тільки після виклику методу Wait цього об’єкту можна буде певними, що дані надійшли до буферу обміну або до відповідного потоку у випадку отримання даних.

**Завдання 2**

При реалізації алгоритму, будемо розсилати частини матриці А та повну матрицю Б до кожного потоку. Після обрахування на кожному потоці, відправимо результати назад та об’єднаємо у кінцеву матрицю С.

Результати роботи можна побачити на рисунку 1.1.

Зображення, що містить знімок екрана, візерунок, текст, дизайн

Автоматично згенерований опис

Рисунок 1.1 Виконання синхронного методу множення

**Завдання 3**

Далі, замінимо виклики до синхронних методів викликами Isend та Irecv. Так, методи відправки частин матриці на початку виконання, та передачу результатів у кінці не будемо очікувато, водночас закінчення отримання буде очікуватися, аби продовжити роботу алгоритму.

Результати роботи можна побачити на рисунку 1.2.

Зображення, що містить знімок екрана, квадрат, візерунок, дизайн

Автоматично згенерований опис

Рисунок 1.2 Виконання асинхронного методу множення

**Завдання 4**

Обчислимо показник ефективності для алгоритму асинхронного множення, варіюючи розмір матриць та кількість потоків.

Ефективність за формулою

Дані можна побачити у таблиці 1.

Таблиця 1. –Ефективність асинхронного алгоритму

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ефективність | | | | |
| ElementCnt/ThreadCnt | 500 | 1000 | 1500 | 1750 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 0.832061 | 0.731563 | 0.929561 | 0.997265 |
| 6 | 0.290667 | 0.418565 | 0.567905 | 0.76397 |
| 12 | 0.134568 | 0.209018 | 0.283953 | 0.286339 |

Результати дослідження по синхронних методах можна побачити на таблиці 2.

Таблиця 2. –Ефективність синхронного алгоритму

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ефективність | | | | |
| ElementCnt/ThreadCnt | 500 | 1000 | 1500 | 1750 |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 0.541667 | 0.794885 | 0.962193 | 0.991559 |
| 6 | 0.179063 | 0.376306 | 0.555601 | 0.64336 |
| 12 | 0.072222 | 0.178491 | 0.219208 | 0.259147 |

Так, можемо побачити що за допомогою використання асинхронних методів змогли досягти більшого відсотку часу активної роботи потоків, відповідно збільшилась загальна ефективність алгоритму.

**ВИСНОВОК**

В рамках роботи, ми реалізували паралельне множення матриць за допомогою MPJ. Були використані як синхронні методи комунікації потоків, так і асинхронні . Ми наочно побачили збільшення складності алгоритму при збільшенні кількості елементів матриць.

В роботі обчисли порівняли показники ефективності алгоритмів синхронної взаємодії та асинхронної. Так, асинхронний метод дає значне покращення ефективності.