Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Комп’ютерний практикум № 8**

з дисципліни «Технології паралельних обчислень»

**Тема:** «Розробка алгоритмів для розподілених систем клієнт-серверної архітектури»

|  |  |
| --- | --- |
| **Виконав:**  студент групи ІТ-04  Чапча Святослав  Дата здачі \_\_\_\_\_\_\_\_  Захищено з балом \_\_\_\_\_\_\_ | **Перевірила:**  ас. кафедри ІПІ  Дифучина О.Ю. |

Київ 2023

**Завдання**

1.    Розробити веб-застосування клієнт-серверної архітектури, що реалізує алгоритм множення  матриць (або інший обчислювальний алгоритм, який був Вами реалізований іншими методами розподілених обчислень в рамках курсу «Паралельні та розподілені обчислення») на стороні сервера. Розгляньте два варіанти реалізації 1) дані для обчислень знаходяться на сервері та 2) дані для обчислень знаходяться на клієнтській частині застосування. **60 балів.**

Скріншоти програми:

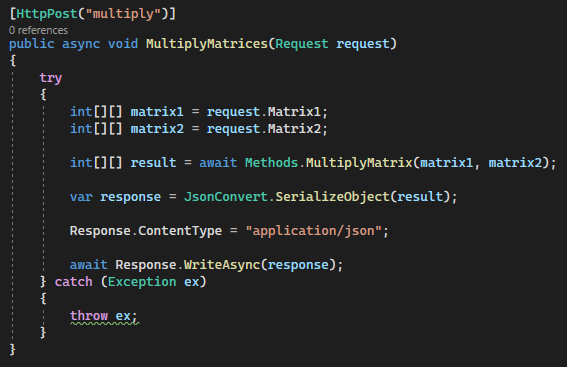
****

Рис. 1. Метод MultiplyMatrices()

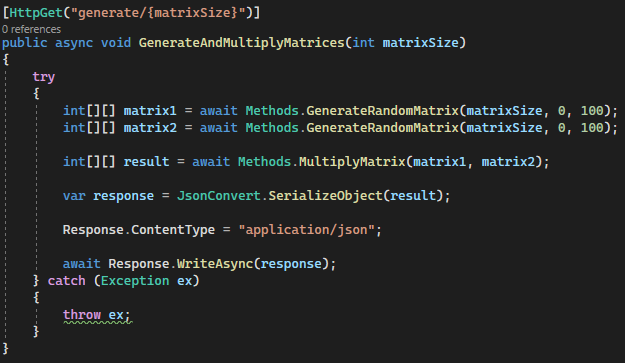
****

Рис. 2. Метод GenerateAndMultiplyMatrices()

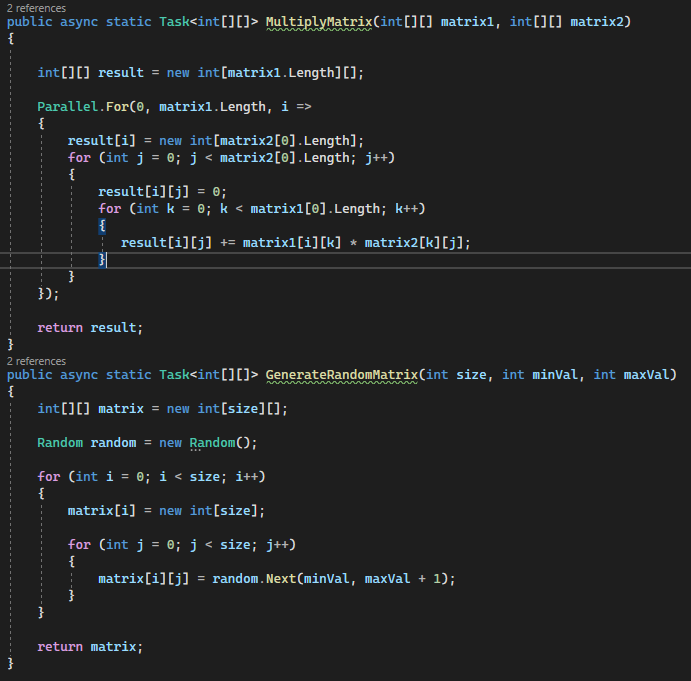
****

Рис. 3. Methods.cs

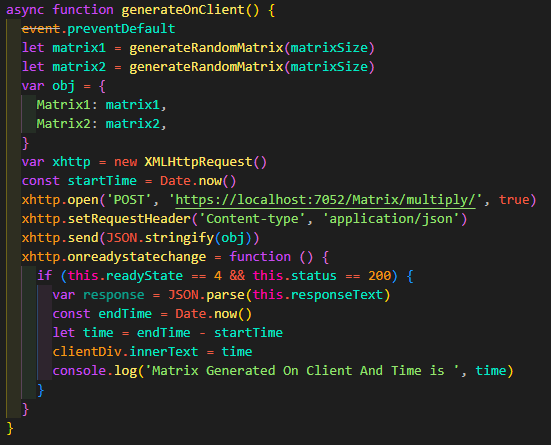


Рис. 4. Метод generateOnClient()

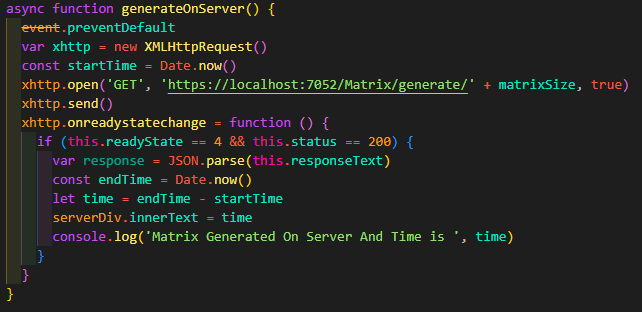
****

Рис. 5. Метод generateOnServer()

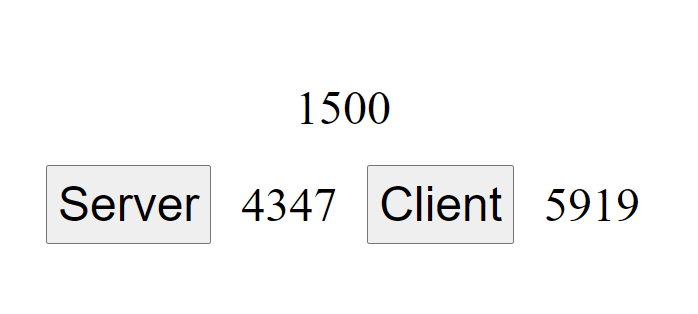


Рис. 6. Результат виконання програми

Для виконання цього завдання, я розробив серверну архітектуру використовуючи C# та ASP.NET Web API та клієнтську частину завдяки JavaScript. На сервері виконуються два методи – це паралельне множення матриць та генерування випадкової матриці. Було створено два методи GET та POST, один з яких просто оброблює та повертає інформацію з сервера, а другий приймає і повертає результат. На клієнтській частині було створено два методи, які роблять запроси на сервер. Один з цих методів генерує матриці і відправляє їх на сервер.

Якщо ми порівняємо отриманий час, то помітимо, що коли інформація зберігається на сервері, запити надходять швидше. Різниця приблизно у 1.36 разів.

2.    Дослідити швидкість виконання запиту користувача при різних обсягах даних. **20 балів.**

Для дослідження швидкості побудуємо таблицю:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Інформація на сервері | Інформація на клієнті | Різниця |
| 100х100 | 19 | 21 | 1.10 |
| 500х500 | 222 | 271 | 1.22 |
| 1000х1000 | 1347 | 1692 | 1.25 |
| 1500х1500 | 4347 | 5919 | 1.36 |
| 2000х2000 | 9794 | 13824 | 1.41 |

Як можемо побачити, зі збільшенням розміру матриці збільшується і час виконання роботи. Також, що більш цікаво, збільшується різниця між двома випадками.

Порівняти реалізацію алгоритму в клієнт-серверній системі та в розподіленій системі з рівноправними процесорами.**20 балів.**

У клієнт-серверній системі, клієнт взаємодіє з сервером за допомогою мережевого протоколу. Якщо алгоритм може бути розбитий на окремі запити, то клієнт-серверна система може бути ефективним варіантом реалізації. Однак, якщо алгоритм потребує велику кількість обрахунків, це може призвести до надмірного трафіку мережі та затримки в обробці запитів клієнтів.

У розподіленій системі з рівноправними процесорами, кожен процесор виконує окрему частину алгоритму. Процесори обмінюються даними та результатами за допомогою мережевого протоколу. Якщо алгоритм може бути розбитий на незалежні операції, які можна виконувати паралельно, то розподілена система може бути ефективним варіантом реалізації. Проте, якщо алгоритм не може бути розбитий на незалежні операції, то розподілена система може працювати повільніше ніж клієнт-серверна система.

До переваг клієнт-серверної системи можна віднести простоту у використанні та керуванні, оскільки всі запити обробляються на сервері, що дозволяє легко відстежувати та контролювати виконання алгоритму. Також, клієнт-серверна архітектура може бути більш безпечною, оскільки сервер може контролювати доступ до даних та забезпечувати захист від зловживань.

Розподілена система з рівноправними процесорами має перевагу у масштабованості та швидкодії, оскільки кожен процесор може виконувати свою частину алгоритму незалежно та паралельно з іншими процесорами. Це дозволяє прискорити обробку даних та підвищити продуктивність системи в цілому. Розподілена система також може бути більш надійною, оскільки при відмові одного процесора, решта продовжать працювати.

Програмний код до всіх завдань наведений в репозиторії:

<https://github.com/whitetark/multithreading/tree/main/lab08>

**Висновки**

Під час виконання лабораторного практикуму №8, я познайомився з реалізацією алгоритму у клієнт-серверній системі. З замірів я зрозумів, що розмір даних сильно впливає на швидкодію, а також на неї впливає, де саме зберігаються дані.