

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 8 з дисципліни
«Технології паралельних обчислень»

«Розробка алгоритмів для розподілених систем клієнт-серверної
архітектури»

Виконав(ла)

ІП-14 Сергієнко Ю. В.
(шифр, прізвище, ім'я, по батькові)

Перевірив

Дифучина О. Ю.
(прізвище, ім'я, по батькові)

Київ 2024

Комп'ютерний практикум 8

Тема: Розробка алгоритмів для розподілених систем клієнт-серверної архітектури.

Виконання:

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 з використанням паралельних обчислень. Розгляньте два варіанти реалізації 1) дані для обчислень знаходяться на сервері та 2) дані для обчислень знаходяться на клієнтській частині застосування. **60 балів.**

Для виконання даного завдання було обрано алгоритм множення матриць Фокса. Фактично, необхідно розгорнути два застосунки: клієнт, де, за вибором користувача, можливо надіслати матриці на обробку або надіслати повідомлення про початок обробки серверних матриць. Також клієнт повинен приймати результуючу матрицю (буде дочікуватись її надсилання).

Щодо серверної частини, необхідно приймати матриці із клієнта або створювати їх тут. Після їх ініціалізації необхідно провести їх множення паралельно, у даному випадку за допомогою ForkJoin framework. Коли отримаємо результат множення, повернемо матрицю до клієнта.

Створимо допоміжний сервіс MatrixService, у якому будемо генерувати вхідні файли значень матриць та проводити усі дії над матрицями (розділення на блоки, множення об'єктів матриць, відправка даних). Згенеруємо початкові файли на рисунку 1.

1	-8;-17;-48;48;30;-3;23;-32;-18;-38;-38;-14;5;4;5;-18;28;-41;15;-3
2	23;-41;-44;-1;41;34;49;12;-39;11;3;2;25;-45;6;-24;-8;34;8;-29;33;
3	-33;-19;3;11;4;-2;-16;48;22;-24;-18;12;37;-49;-48;-49;6;8;43;26;2
4	-18;-29;-30;-17;45;34;26;14;-3;-16;14;45;-2;-7;-3;29;-47;30;-38;2
5	22;-21;28;-21;38;-6;-8;8;-11;-35;19;41;4;32;29;-39;-36;-26;12;33;
6	18;17;4;-28;31;-36;-3;-39;-3;13;47;6;48;13;-41;-40;-14;-5;-32;30;
7	-27;-20;21;-3;-4;47;-1;29;35;6;6;20;-28;-44;-18;16;-18;28;-41;24;
8	16;-2;-50;7;13;11;-8;-47;-31;38;46;-47;18;-46;-20;32;5;39;27;28;-
9	6;23;-41;20;-14;-26;-28;10;-25;-44;-45;43;32;-35;-17;-36;-33;-39;
10	4;-41;47;-13;38;-16;-34;-48;-48;22;-40;17;43;-44;29;28;-37;39;42;
11	-2;13;-39;-2;40;-26;-9;-14;-45;-25;16;44;-15;12;-26;21;-19;31;41;
12	46;33;48;4;20;-15;-3;31;-28;-39;42;8;-4;41;-19;-31;-43;37;18;-6;-
13	20;15;8;-46;11;-48;27;19;-19;-26;-14;29;23;30;45;21;-38;20;-13;15
14	46;-2;28;44;-35;-33;30;1;-34;-24;21;-44;-47;3;-30;-44;-27;-11;-30
15	-42;41;5;33;27;-44;0;-46;32;34;-39;-41;24;-19;8;42;32;42;14;-7;28
16	35;-29;25;35;39;34;-39;38;-31;-6;2;28;47;-18;38;13;46;9;-16;-6;14
17	-24;41;-33;-9;48;4;10;37;-6;-9;9;36;9;2;-23;38;-15;-20;-29;41;39;

Рисунок 1 – Генерація початкових файлів

Запустимо сервер та клієнт. Сервер буде очікувати надходження даних, після чого помножить матриці та поверне результат. Клієнт, після надсилання даних, дочікується результату множення. Після чого, будемо порівнювати результат Фокса з результатом звичайного множення матриць. Результати роботи клієнта та сервера відображено на рисунках 2-5 відповідно.

```
Matrix size: 1000

Choose the type:
(1) get matrices from the client
(2) get matrices from the server
1/2: 1
Time took: 758 ms
Are same: true
```

Рисунок 2 – Відпрацювання клієнта (випадок 1)

```

Matrix size: 1000

Choose the type:
(1) get matrices from the client
(2) get matrices from the server
1/2: 2
Time took: 643 ms
Are same: true

```

Рисунок 3 – Відпрацювання клієнта (випадок 2)

```

Matrices will be received from the client.
Waiting for the matrices..
Matrices are set!
Multiplying -> time took: 341 ms

```

Рисунок 4 – Відпрацювання сервера (випадок 1)

```

Matrices will be set now.
Setting matrices on the server..
Matrices are set!
Multiplying -> time took: 352 ms

```

Рисунок 5 – Відпрацювання сервера (випадок 2)

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

Будемо досліджувати два запити: на множення клієнтських та серверних матриць. Заздалегідь зрозуміло, що множення матриць без їх пересилання повинно бути швидше, але необхідно це дослідити (таблиця 1, де C2S – client to server, S2S – server to server, S – sending, M – multiply).

Таблиця 1 – Результати досліджень

Size	C2S, ms		S2S, ms		Acc, t	
	S+M	S	S+M	S	S+M	S
500	244	161	175	85	1.39	1.89
1000	667	337	560	224	1.19	1.5
1500	2846	683	2469	425	1.15	1.6
2000	3105	1107	2743	793	1.13	1.4

Продовження таблиці 1:

Size	C2S, ms		S2S, ms		Acc, t	
	S+M	S	S+M	S	S+M	S
3000	16281	2685	14227	1502	1.14	1.78

За результатами вимірювань бачимо, що час пересилання матриць найбільш суттєво впливає при найменших та найбільших об'ємах. В цілому було доведено, що множення матриць із вхідними даними на сервері є швидшим через уникання одного з пересилань.

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

Порівнюючи дану реалізацію та множення матриць з використанням MPI, можна провести наступну паралель: клієнт – це Майстер, сервер – воркер. Майстер здійснює пересилання матриць до воркера, множить матриці та отримує результат. Схожим випадком є метод обміну «один-до-одного», де Майстер не бере участі у обробці, а тільки пересилає вхідні та отримує результуючі.

В цілому, обидві реалізації мають схожі аспекти: зв'язок за допомогою сокету, вказуючи порт та хост, нагадує зв'язок методів MPI_Send() та MPI_Recv() за допомогою індексів source, dest та tag.

Висновок

Під час виконання даного комп'ютерного практикуму я здобув знання та навички розробки алгоритмів для клієнт-серверної архітектури. Було використано сокети як gateway між Клієнтом та Сервером, а за основу паралельних обчислень на Сервері було взято алгоритм множення матриць Фокса. Було опрацьовано два випадки: пересилання вхідних матриць із Клієнта та ініціалізація матриць на Сервері.

Код програми доступний на [Github](#).