**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 № 7 з дисципліни «Технології паралельних обчислень»

**«Розробка паралельного алгоритму множення матриць з використанням МРІ-методів колективного обміну повідомленнями («один-до-багатьох», «багато-до-одного», «багато-до-багатьох») та дослідження його ефективності»**

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-02, Зусько Владислав Юрійович*

Київ 2023

**Завдання #1**

Реалізувати алгоритм паралельного множення матриць з використанням розподілених обчислень в MPI з використанням методів колективного обміну повідомленнями. **40 балів.**

Для виконання поставленої задачі ми можемо використати напрацювання попередніх робіт. Там нами було реалізовано множення матриць методом “один-до-одного”, що дещо сповільнювало процес.

У межах даної роботи ми змінимо метод обміну повідомленнями. Перш за все, оновимо метод надсилання рядків матриці А. В цьому нам допоможе метод спілкування “один-до-багатьох”, а саме Scatterv, який надає змогу надсилати різні кількість рядків буферу, що ділиться згідно з кількістю процесів.

Відправка матриці В теж є малоефективною. Логічно буде використати метод надсилання “один-до-багатьох” без розділення – Bcast.

Фінальним етапом роботи програми є збір даних з субпроцесів у результуючу матрицю. Для такої операції вдало можна використати метод Gatherv, що збере дані у буфер С. На цьому етапі також можна розглянути метод Allgatherv, що додатково зібрані дані надає у процеси-воркери. Проте, цей варіант не є оптимальним через відсутність потреби у використанні фінальної матриці в субпроцесах.

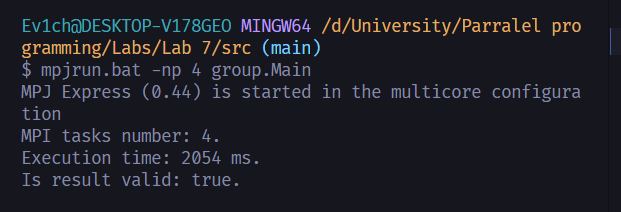


Рисунок – Відображення процесу роботи програми на прикладі матриці розміром 1000х1000

**Завдання #4**

Дослідити ефективність розподіленого обчислення алгоритму множення матриць при збільшенні розміру матриць та при збільшенні кількості вузлів, на яких здійснюється запуск програми. Порівняйте ефективність алгоритму при використанні блокуючих та неблокуючих методів обміну повідомленнями. **40 балів.**

|  |  |  |  |
| --- | --- | --- | --- |
| Розмір | Синхронізований (8 процесів) | Асинхронний (8 процесів) | C/A |
| 1000 | 1475 | 1224 | 1.205065 |
| 1500 | 7481 | 7035 | 1.063397 |
| 2000 | 18904 | 17243 | 1.096329 |

|  |  |  |  |
| --- | --- | --- | --- |
| Розмір | Асинхронний (4 процеси) | Асинхронний (8 процесів) | Асинхронний (16 процесів) |
| 1000 | 1879 | 1157 | 1188 |
| 1500 | 13284 | 7675 | 7810 |
| 2000 | 35369 | 21119 | 21462 |