

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт до комп'ютерного практикуму №6

З дисципліни «Технології паралельних обчислень»

Прийняв:

Викладач

Дифучина О. Ю.

Оцінка: _____

«___» _____ 2025 р.

Виконав:

Студент 3 курсу, гр. ПІ-24

Кондратюк Вадим Русланович

Комп'ютерний практикум №6

1) Текст завдання

5.6 Завдання до комп'ютерного практикуму 6 «Розробка паралельного алгоритму множення матриць з використанням MPI-методів обміну повідомленнями «один-до-одного» та дослідження його ефективності»

1. Ознайомитись з методами блокуючого та неблокуючого обміну повідомленнями типу point-to-point (див. лекцію та документацію стандарту MPI).
2. Реалізувати алгоритм паралельного множення матриць з використанням розподілених обчислень в MPI з використанням методів блокуючого обміну повідомленнями (лістинг 1). 30 балів.
3. Реалізувати алгоритм паралельного множення матриць з використанням розподілених обчислень в MPI з використанням методів неблокуючого обміну повідомленнями. 30 балів
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.

2) Лістинг програмного коду

Увесь програмний код було завантажено на гітхаб за посиланням – <https://github.com/VadymKondratiuk/tpo/tree/main/LAB6>

3) Скріншоти запуску програми

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
• $ mpjrun.sh -np 4 blocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 100x100, Workers: 3
Execution time: 12 ms
```

Рис.1 – множення матриць з використанням **блокуючих** методів обміну повідомленнями (100x100, 4 процеси)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
• $ mpjrun.sh -np 6 blocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 500x500, Workers: 5
Execution time: 88 ms
```

Рис.2 – множення матриць з використанням **блокуючих** методів обміну повідомленнями (500x500, 6 процесів)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
• $ mpjrun.sh -np 8 blocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 1000x1000, Workers: 7
Execution time: 383 ms
```

Рис.3 – множення матриць з використанням **блокуючих** методів обміну повідомленнями (1000x1000, 8 процесів)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
• $ mpjrun.sh -np 4 blocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 2000x2000, Workers: 3
Execution time: 13867 ms
```

Рис.4 – множення матриць з використанням **блокуючих** методів обміну повідомленнями (2000x2000, 4 процеси)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
• $ mpjrun.sh -np 6 blocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 4000x4000, Workers: 5
Execution time: 127805 ms
```

Рис.5 – множення матриць з використанням **блокуючих** методів обміну повідомленнями (4000x4000, 6 процесів)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
$ mpjrun.sh -np 4 nonblocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 100x100, Workers: 3
Execution time: 15 ms
```

Рис.6 – множення матриць з використанням **неблокуючих** методів обміну повідомленнями (100x100, 4 процеси)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
$ mpjrun.sh -np 6 nonblocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 500x500, Workers: 5
Execution time: 94 ms
```

Рис.7 – множення матриць з використанням **неблокуючих** методів обміну повідомленнями (500x500, 6 процесів)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
$ mpjrun.sh -np 8 nonblocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 1000x1000, Workers: 7
Execution time: 430 ms
```

Рис.8 – множення матриць з використанням **неблокуючих** методів обміну повідомленнями (1000x1000, 8 процесів)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
$ mpjrun.sh -np 4 nonblocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 2000x2000, Workers: 3
Execution time: 15863 ms
```

Рис.9 – множення матриць з використанням **неблокуючих** методів обміну повідомленнями (2000x2000, 4 процеси)

```
vadik@VadymKondratiuk MINGW64 /d/KPI-2025/TPO/LAB6
$ mpjrun.sh -np 6 nonblocking
MPJ Express (0.44) is started in the multicore configuration
Matrix Size: 4000x4000, Workers: 5
Execution time: 115127 ms
```

Рис.10 – множення матриць з використанням **неблокуючих** методів обміну повідомленнями (4000x4000, 6 процесів)

4) Результати

Для дослідження ефективності розподіленого обчислення алгоритму множення матриць при збільшенні розміру матриць та при збільшенні кількості вузлів, на яких здійснюється запуск програми було проведено три спроби тестування для блокуючого та неблокуючого обміну повідомленнями.

Тестування здійснювалося для квадратних матриць різного розміру (100, 500, 1000, 2000, 4000) із використанням 4, 6 та 8 процесів.

Результати кожної з трьох спроб наведено у Таблицях 1 відповідно. У Таблиці 2 подано порівняльну таблицю, що містить середній час виконання.

Таблиця 1 – три спроби тестування

Matrix Size	Threads	Blocking (ms)	Nonblocking (ms)
100	4	13, 11, 12	10, 14, 13
100	6	16, 17, 15	18, 16, 16
100	8	15, 14, 18	15, 15, 18
500	4	96, 87, 98	99, 94, 93
500	6	82, 79, 87	81, 85, 91
500	8	90, 94, 99	104, 95, 101
1000	4	593, 502, 510	488, 504, 538
1000	6	441, 409, 390	386, 383, 459
1000	8	391, 391, 376	444, 394, 437
2000	4	19727, 18228, 14291	15026, 15670, 18970
2000	6	8861, 8790, 8899	11025, 11765, 12599
2000	8	12186, 12179, 13626	8872, 9228, 10234
4000	4	165415, 143262, 146507	173682, 175859, 173947
4000	6	116656, 134676, 143317	127515, 144060, 129884
4000	8	96205, 97899, 108754	106849, 110973, 109165

Таблиця 2 – середні значення

Matrix Size	Threads	Blocking (ms)	Nonblocking (ms)
100	4	12	12.33
100	6	16	16.67
100	8	15.67	16
500	4	93.67	95.33
500	6	82.67	85.67
500	8	94.33	100
1000	4	535	510
1000	6	413.33	409.33
1000	8	386	425
2000	4	17482	16568.67
2000	6	8826.67	11496.33
2000	8	12694	9378.33
4000	4	151590.67	176274.67
4000	6	131503.67	134446.33
4000	8	101614.67	108343.67

На основі експериментальних результатів можна зробити висновок, що ефективність блокуючих і неблокуючих обмінів залежить від розміру матриць і кількості процесів. Для невеликих розмірів матриць різниця в часі виконання між двома підходами незначна, а іноді блокуючий варіант навіть швидший завдяки простішій реалізації та меншому накладному часу.

Висновок:

У ході виконання завдання було реалізовано два варіанти алгоритму паралельного множення матриць у середовищі MPI — з використанням блокуючого та неблокуючого обміну повідомленнями. Ознайомлення з принципами point-to-point комунікацій дозволило глибше зрозуміти відмінності між методами Send/Recv і Isend/Irecv, а також їхній вплив на організацію взаємодії між процесами в розподіленій системі.

Після реалізації обох підходів було проведено експериментальне дослідження ефективності виконання при різних розмірах матриць і кількості процесів. Результати показали, що блокуючий обмін повідомленнями часто демонструє стабільніший час виконання, тоді як неблокуючий у деяких випадках дозволяє досягнути кращої продуктивності завдяки перекриттю обчислень і передавання даних.

