ЗВІТ

по Лабораторній роботі №2

на тему

"Parallel Algorithms of Matrix-Vector Multiplication"

студента 2-го курсу магістратури

групи ТТП-2

Каплюка Владислава

Теоретична частина

Мета лабораторної роботи полягає у дослідженні паралельного програмування на прикладі виконання програми множення матриці на матрицю.

Етапи дослідження:

1. Постановка задачі множення матриці на матрицю
2. Написання програми, яка виконує множення матриці на матрицю для однопроцесорного випадку
3. Дослідити алгоритм паралельного виконання множення матриці на матрицю
4. Написання програми, яка виконує множення матриці на матрицю для паралельного випадку

Формула отримання значення для результоючої матриці:

$$c_{ij} = \left(a_i, b_j^T\right), a_i = \left(a_{i0,} a_{i1}, ..., a_{in-1}\right), b_j^T = \left(b_{0j}, b_{1j}, ..., b_{n-1j}\right)^T .$$

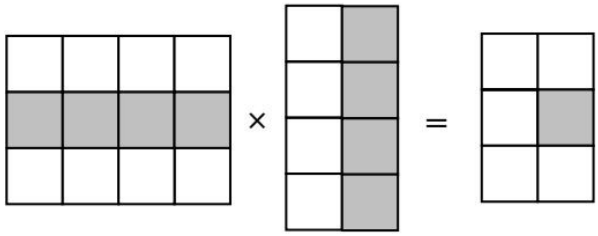


Figure. 2.1. The Element of the Result Matrix *C* is the Result of the Scalar Multiplication of the Corresponding Matrix *A* Row of the Matrix *A* and the Column of the Matrix *B*

Приклад:

$$\begin{pmatrix} 3 & 2 & 0 & -1 \\ 5 & -2 & 1 & 1 \\ 1 & 0 & -1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 & -1 \\ 2 & 5 \\ -3 & 2 \\ 7 & 4 \end{pmatrix} = \begin{pmatrix} 0 & 3 \\ 5 & -9 \\ -3 & -7 \end{pmatrix}$$

Принцип паралелізму у рамках **MPI** полягає у розбитті задачі на підзадачі, які виконуватимуть інші процесори. Одночасно на одному процесорі може розташовуватись кілька процесів. Кількість процесів і кількість використовуваних процесорів визначаються в момент запуску паралельної програми за допомогою середовища виконання програми MPI. Номер процесу називається **рангом** процесу.

Паралельні програмні процеси об'єднуються в групи. Комунікатор в MPI – спеціально розроблений об’єкти управління, який об’єднує в собі групу процесів і ряд додаткових параметрів, які використовуються при виконанні операцій передачі даних. У MPI необхідно обов’язково вказувати комунікатор, який використовується для операцій передачі даних. Усі процеси, доступні в паралельній програмі, належать комунікатеру з ідентифікатором MPI_COLUMN_WORLD, який створюється за замовчуванням.

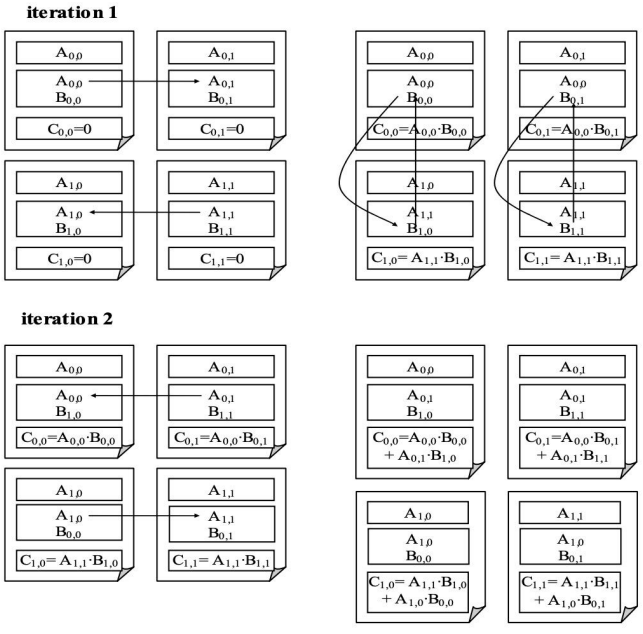
При використанні методу паралельного програмування, одна об’ємна задача розбивається на окремі блоки. В нашому випадку, окремими блоками слугуватимуть підматриці матриць A, B та C (як на малюнку далі).

$$\begin{pmatrix} A_{00}A_{01}\ldots A_{0q-1} \\ \cdots \\ A_{q-10}A_{q-11}\ldots A_{q-1q-1} \end{pmatrix} \times \begin{pmatrix} B_{00}B_{01}\ldots B_{0q-1} \\ \cdots \\ B_{q-10}B_{q-11}\ldots B_{q-1q-1} \end{pmatrix} = \begin{pmatrix} C_{00}C_{01}\ldots C_{0q-1} \\ \cdots \\ c_{q-10}C_{q-11}\ldots C_{q-1q-1} \end{pmatrix},$$

where each block C_{ij} of matrix C is defined in accordance with the expression:

$$C_{ij} = \sum_{s=0}^{q-1} A_{is} B_{sj} \ .$$

Ітерації виконуватимуться наступним чином:



Практична частина

Test Number	Matrix size	Serial algorithm	Parallel algorithm			
			4 processors		9 processors	
			Time	Speed up	Time	Speed up
1	12	0.000000	0.000217	**0.000000**	0.005421	**0.000000**
2	102	0.001	0.001616	**0.618811**	0.027023	**0.037005**
3	522	0.207	0.072176	**2.867983**	0.112751	**1.835903**
4	1200	4.960	1.839645	**2.696172**	1.636215	**3.031386**
5	1500	14.340	4.487795	**3.195331**	3.769482	**3.804236**
6	2100	95.848	21.504972	**4.457015**	11.845334	**8.091624**
7	2502	170.877	51.123722	**3.34242**	29.781807	**5.737630**
8	3000	310.243	128.923538	**2.406411**	76.366273	**4.062565**

Serial Log:

serial.log

```
----------
Serial.exe
----------
--------------------------------------------------

Enter the size of matrices=Test data = 12; Time of execution = 0.000000

Enter the size of matrices=Test data = 102; Time of execution = 0.001000

Enter the size of matrices=Test data = 522; Time of execution = 0.207000

Enter the size of matrices=Test data = 1200; Time of execution = 4.960000

Enter the size of matrices=Test data = 1500; Time of execution = 14.340000

Enter the size of matrices=Test data = 2100; Time of execution = 95.848000

Enter the size of matrices=Test data = 2502; Time of execution = 170.877000

Enter the size of matrices=Test data = 3000; Time of execution = 310.243000
```

Parallel log 4 processes:

```
----------
FoxMethod.exe
----------
-------------------------------------------------
Size: 12  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 0.000217
Size: 102  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 0.001616
Size: 522  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 0.072176
Size: 1200  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 1.839645
Size: 1500  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 4.487795
Size: 2100  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 21.504972
Size: 2502  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 51.123722
Size: 3000  Number of processes:4
The results of serial and parallel algorithms are identical. Time of execution = 128.923538
```

Parallel log 9 processes :

```
Size: 12  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 0.005421
Size: 102  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 0.027023
Size: 522  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 0.112751
Size: 1200  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 1.636215
Size: 1500  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 3.769482
Size: 2100  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 11.845334
Size: 2502  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 29.781807
Size: 3000  Number of processes:9
The results of serial and parallel algorithms are identical. Time of execution = 76.366273
```

Значення $\omega = 0.00000000195132$ – швидкість виконання 1 арифметичної операції на персональному комп’ютері.

Заповення наступної таблиці здійснюється за допомогою наступної формулі

$T_p = q[(n^2 / p) \cdot (2n / q - 1) + (n^2 / p)] \cdot \tau$, де тау = ω, n – розмірність матриці, p – кількість процесорів, q – розмір “сітки” процесора (sqrt(p)). Маємо:

Test Number	Matrix size	Parallel algorithm			
		4 processors		9 processors	
		Model	Experiment	Model	Experiment
1	12	0.000001	0.000217	0.000000	0.005421
2	102	0.000517	0.001616	0.0001537	0.027023
3	522	0.0693876	0.072176	0.020552	0.112751
4	1200	3.79922	1.839645	1.727894	1.636215
5	1500	10.515175	4.487795	6.400329	3.769482
6	2100	34.08028	21.504972	22.032354	11.845334
7	2502	51.96608	51.123722	42.173270	29.781807
8	3000	101.858034	128.923538	77.808885	76.366273