**Міністерство освіти і науки України**

**Національний університет "Львівська політехніка"**

**Кафедра ЕОМ**

****

**Звіт з розрахункової роботи**

**з дисципліни “Паралельні та розподілені обчислення ”**

**Виконав: ст. гр. КІ-33**

**Крансоштан О.О.**

**Прийняв: асистент**

**Козак Н.Б.**

**Львів 2020 р.**

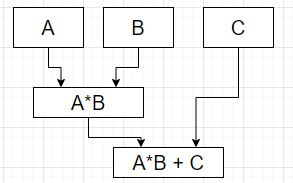
1. **Чи можна організувати паралельні обчислення для розв’язання задачі знаходження найменшого числа з трьох чисел?, з чотирьох чисел? Якщо можна, то яким чином?**

Так, можна.

Для 3 чисел – 1-ше число подати в 1-ий потік, 2-ге число подати в 2-ий потік, паралельно порівняти числа з потоків з 3-ім числом. Якщо знайшлось менше значення – перезаписати число з потоку, інакше порівняти числа в обох потоках і отримати результат.

Для 4 чисел провести паралельне порівняння попарно в 2-х потоках залишивши в кожному потоці менше число. Порівняти залишкові числа між собою і отримати результат.

1. **Розробіть та опишіть варіант структурної схем виконання виразу у =A\*B+C , яка забезпечує паралельну організацію роботи. N= 2**

Вхідні дані: A, B,C

Результатом множення матриць є матриця. Кінцевий результат — матриця.

Спочатку виконуємо обчислення виразу   
А \* B. Далі до результату додаємо значення С і отримуємо y.

Рис. 1. Структурна схема виконання

виразу

1. **Чим відрізняється паралельна організація роботи від конвеєрної?**

Конвеєр передбачає один робочий потік в якому в певний момент часу відбувається виконання однієї команди і підготовка до виконання іншої. В паралельних системах команди виконуються одночасно.

1. **Необхідно перемножити матрицю А (n1(4)\*n2(4)) на матрицю В (n2(4)\*n3(4)). Розробіть функціональну схему пристрою, який забезпечить найшвидше виконання даної операції.**
2. Паралельна програма виконується на MIMD – системі з 100 процесорами, 3% всіх команд при проході програми виконуються послідовно, а решту може виконуватись паралельно на всіх процесорах. Яке значення має показник прискорення цієї програми на даній програмі?
3. Деяка паралельна програма, що має 10% послідовну частину, має виконуватись на MIMD – системі. Чи існує деякий максимально можливий показник прискорення, незалежний від кількості процесорів системи?
4. Паралельна програма має виконуватись на MIMD – системі з *N(10)* процесорами, проте: *M(20)* % всіх команд при проході програми мають виконуватися послідовно; *Р(80)* % всіх команд можуть виконуватись тільки на *N* /*2* процесорах. Яке значення має показник прискорення для цієї програми? *N =10*, *M =20*,*Р =* 80
5. Відомо, що алгоритм добре відображається на топологію ”кільце”. Чи можна гарантувати його хороше відображення на топологію ”гіперкуб”? Відповідь обгрунтуйте.



1. Прискорення S=T1/T100

Час паралельного виконання Т100=3T1+((1-3)T1/100), де 3 – відсоток послідовного коду, 100 – кількість процесорів, Т1 – час послідовного виконання програми. => S=T1/2.98T1 => S=1/2.98=0.3356 або 33,56%.

1. Згідно з законом Амдаля програма з 10% послідовних розрахунків не може отримати більше ніж 10-ти кратне прискорення при паралельному режимі роботи.
2. При наявності 20% послідовного коду, а також за умови виконання решти 80% на половині процесорів(10/2=5) => S=0.05 або 5%.
3. Так, можна.

Найбільш ефективний таке накладання буде для гіперкуба розміру d=2 що являтиме собою аналог топології кільце або d=3, в якому прослідковується звя’зок схожий на комбінація топологій кільця та повного графа, за вийнятком того, що нумерація вершин в топології гіперкуб різнитиметься на 1 біт.

1. Що таке порядок вузла і комутаційний діаметр мережі?

Порядок вузла – величина визначається кількість сусідніх вузлів, де сусідній вузол – той, з яким є пряме з’єднання.

Комунікаційний діаметр мережі – це максимальний шлях між будь-якими двома вузлами.

1. **Допустимо, що перемножуються дві квадратні матриці. Які появляться особливості в організації обчислювальних процесів, якщо взяти матриці максимального розміру і старатися розв’язати задачу максимально швидко? Розгляньте варіанти обчислювальних процесів:**

*а) з спільною пам’яттю і універсальними процесорами;*

Спільна пам’ять передбачає наявність єдиної шини даних для всіх процесорів, тому при роботі з множенням матриць великих розмірів можуть виникати затримки в доступі до пам’яті. Універсальні процесори, тобто ті що готові до будь-яких задач, покажуть нижчі показники швидкодії на відміну від спеціалізованих систем для такого роду обчислень.

*б) з спільною пам’яттю і конвеєрними суматорами, перемножувачами і пристроями ділення;*

Спільна пам’ять має ту ж проблему, що й у пункті 1, проте конвеєрна система суматорів, помножувачів та пристроїв ділення дасть значне прискорення при великих об’ємах розрахунків.(найшвидший варіант)

*в)  з розподіленою пам’яттю і універсальними процесорами;*

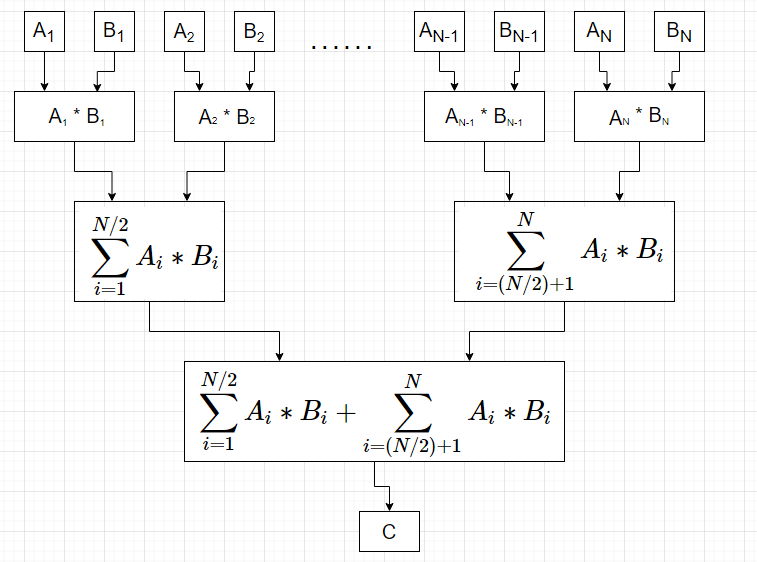
Розділена пам’ять додасть затримки так, як для отримання часткових результатів чи інших даних процесорам доведеться обмінюватись повідомленнями з запитами цих даних. Нюанси універсальних процесорів ті ж, що й у пункті 1.(найповільніший варіант)

*г) з розподіленою пам’яттю і конвеєрними суматорами, перемножувачами і пристроями ділення.*

Як і в пункті 3 розділена пам’ять додасть затримки в між-процесорний обмін даними, проте спеціалізована конвеєрна система арифметико-логічних пристроїв покаже свою ефективність при обчисленнях.

***Тому найшвидший варіант обчислювальних процесів — це з спільною пам’яттю і конверними суматорами, перемножувачами, і пристроями ділення, а найповільніший — з розподіленою пам’яттю і універсальними процесорами***

1. **Розробіть паралельний алгоритм обчислення величини** **, де А і В – одновимірні масиви.**

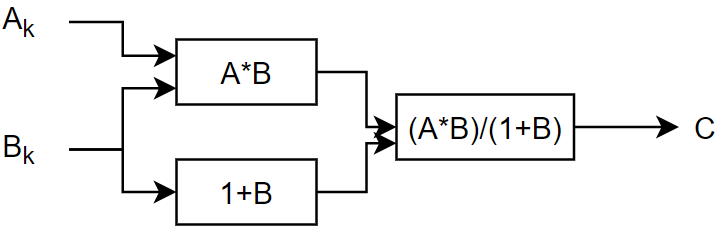


Як видно зі схеми вище – програма ділиться на потоки де знаходяться часткові добутки, після чого вони сумуються у результуючу матрицю/число(залежно від множення стовпець на рядок чи навпаки).

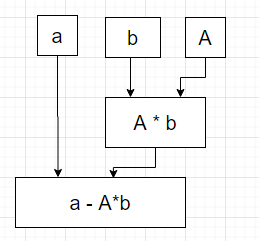
1. **Дані матриці А і В. Розробіть алгоритм обчислення матриці С = А\*В – В\*С.**

Перетворимо вираз:

A\*B=C+(B\*C) => A\*B=C(1+B) => C = (A\*B)/(1+B)



1. **Дана матриця А і вектори *а* і *b*. Розробіть алгоритм обчислення матриці С = *а* - А\* *b.***



Як видно зі схеми вище – спершу ми обраховуємо результат множення матриці А на вектор b, де результатом є вектор. Тоді від нашого вектора а ми віднімаємо новоутворений і отримуємо результат.