**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи № 1

з дисципліни «Паралельні та розподілені обчислення»

на тему: «Використання функціональної декомпозиції для розв’язку обчислювальних задач»

Виконав:

ст.гр. КІ-34

Мороз О.Ю.

Прийняв:

Козак Н.Б.

**Львів 2020**

Мета роботи: Вивчити методи декомпозицій задач. Набути навиків розв’язування задач з використанням функціональної декомпозиції.

**Завдання:**

Вираз, який слід обрахувати, заданий наступним чином:



При чому елементи  визначаються згідно правил:

**y1=A×b**, де для парних **bi=1/(i2+2+i)** для непарних **bi=1/i**, **і=1,2,...n**

**y2=A1(b1-2c1)**;

**Y3 = A2­(B2-C2)**, де **Cij=1/(i+j)**

Хід роботи:

1.  Аналіз завдання.

Для заданого виразу вхідними даними є:

розмірність матриць – n;

матриці ;

вектори-стовпці .

Ці параметри повинні вводитися з клавіатури, або генеруватися випадковим чином (крім розмірності). При чому, елементи всіх матриць та векторів є цілими додатними числами, більшими за нуль.

Вектор-стовпець  та матриця  обраховуються, виходячи з уведеної розмірності, зауважимо, що значення їх елементів завжди менші одиниці і різко спадають зі збільшенням розмірності.

Наприклад для n=3, значення вектора-стовпця  будуть становити:

b1=1;

b2=1/(i2+2+i)=0.625;

b3=0.33333,

,а значення матриці , відповідно:

C11=1/(1+1)=0.5

C12=1/(1+2)= 0.3333

C13=1/(1+3)=0.25

C21=1/(2+1)=0.3333

C22=1/(2+2)=0.25

C23=1/(2+3)=0.2

C31=1/(3+1)=0.25

C32=1/(3+2)=0.2

C33=1/(3+3)=0.16667

При утворенні  враховуємо, що результатом множення матриці А на вектор-стовпець b є вектор-стовпець, елементи якого будуть раціональними числами(тобто матимуть значущу дробову частину).

При утворенні  враховуємо, що результатом додавання двох векторів-стовпців є вектор-стовпець, елементи якого можуть бути більшими за нуль цілими числами. Далі, при множенні цілочисельної додатної матриці А1 на результат додавання, отримаємо вектор-стовпець з цілочисельними елементами.

При утворенні  враховуємо, що присутні лише операції віднімання та множення, а тому вихідний результат може бути додатнім або відємним і завжди матиме значущу дробову частину.

Таким чином, згідно поставленої задачі, в обчисленні загального виразу приймають участь три різні елементи – два вектори стовпці  та матриця .

Перший множник загального виразу містить два доданка – добуток транспонованого вектора-стовпця , матриці  в квадраті, вектора стовпця y1 та транспонованого вектора-стовпця y1.Оскільки, згідно правил матричних обчислень, добуток не є комутативною операцією, всі множення слід виконувати в тій послідовності, яка задана. Результатом множення матриці в квадраті на рядок є рядок, а рядок на ствопець - число, і число додати рядок це рядок. Тому, в загальному, перший доданок рядком.

Другий множник складається з суми добутка матриці Y3 на вектор стовпець y1 та вектора стовпця y2. Результатом множення стовпця на матрицю є стовпець і додавши стовпець вийде стовпець.

Таким чином, добутком стовпця і рядка є число, отже остаточний результат є числом.

2.  Декомпозиція задачі.

Однозначно, всі обчислення безпосередньо залежать від розмірності даних, тому найперше, слід забезпечити ввід змінної n, що визначає цю розмірність. Далі, можна паралельно виконувати обчислення значень вектора b та матриці С2, оскільки вони незалежні від інших параметрів. Крім того, на тому ж рівні декомпозиції слід визначати вхідні дані, тобто вводити з клавіатури, або генерувати випадковим чином матриці  та вектори-стовпці . Наступний рівень декомпозиції – це знаходження елементів виразу. Значення  залежить від введеної матриці А та обрахованого вектора b. Значення  залежить від введеної А1 та різниці векторів b1 і c1, тому знайти його можна лише після обчислення (b1 - c1). Зауважимо, що множення на константу не є окремою операцією, як і транспонування векторів. Аналогічно, знаходимо . Подальша декомпозиція відбувається згідно заданої послідовності операцій та врахування залежностей отриманих на кожному рівні даних. Повна схема декомпозиції обчислення заданого виразу приведена нижче.

3. Об’єднання частин виразу проведено безпосередньо у схемі декомпозиції, оскільки воно однозначно визначається порядком обчислень.

4. Для написання програми, що ілюструє процес обчислення виразу згідно розробленої схеми декомпозиції, слід врахувати наступні особливості:

Дані, що вводяться з клавіатури, або генеруються випадковим чином повинні бути цілими числами, більшими за нуль. Результати проміжних обрахунків будуть містити дробову частину, виняток складає лише елемент y2. Тому слід обирати відповідний тип даних(багатобайтний).

Оскільки на окремо взятому рівні декомпозиції обраховуються незалежні частини підвиразів, функції що їх програмно реалізують повинні викликатися псевдо-одночасно.(Тобто перш ніж виконувати будь яку функцію третього рівня, слід виконати всі функції другого рівня і т.д.)



#### Рис.1. Схема декомпозиції обчислення виразу

5. Код програми

n = input('n = '); %n - створюється змінна,її тим визначається згідно з тим що їй присвоюється, може бути число, вектор, матриця, рядок

r = input('Щоб задати масиви вручну введіть 1, випадково 2: '); %input() функція для зчитування значення з клавіатури, параметром є рядок який буде виводитись

if (r == 2) %if - оператор if закінчується оператором end

A = randi(10,n,n); %A - змінна, randint - генерація масиву цілих чисел, параметри n - к-сть рядків, n - к-сть стовпців, 10 - діапазон чисел що генеруються 0..10

mess = 'A = '; %mess - рядкова змінна для запису в файл, всі рядки заключаються в одинарні лапки

dlmwrite('1.txt', mess, 'delimiter', ''); %dlmwrite - функція запису в файл

dlmwrite('1.txt', A, '-append','delimiter', '\t', 'precision',15); %вона має наступні параметри '1.txt' - файл, A - Змінна яку треба записати,'-append' - дописувати в файл, 'delimiter' - задати явно розділювач для елементів матриці, '\t' - розділювач, 'precision' - задати точність типу double, 15 - число має складатись з 15 знаків

A1 = randi(10,n,n);

mess = 'A1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', A1, '-append','delimiter', '\t', 'precision',15);

b1 = randi(10,n,1);

mess = 'b1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', b1, '-append','delimiter', '\t', 'precision',15);

c1 = randi(10,n,1);

mess = 'c1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', c1, '-append','delimiter', '\t', 'precision',15);

A2 = randi(10,n,n);

mess = 'A2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', A2, '-append','delimiter', '\t', 'precision',15);

B2 = randi(10,n,n);

mess = 'B2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', B2, '-append','delimiter', '\t', 'precision',15);

else

A = randi(10,n,n);

A1 = randi(10,n,n);

b = randi(10,n,1);

b1 = randi(10,n,1);

c1 = randi(10,n,1);

A2 = randi(10,n,n);

B2 = randi(10,n,n);

C2 = randi(10,n,n);

for i = 1:n %Вкладений цикл for змынна і набуває значень від 1 до n

for j = 1:n %Внутрішній цикл for змінна j набуває значень від 1 до n

A(i,j) = input('A[i,j] = '); %

end %Завершує внутрішній цикл

end %Завершує зовнішній цикл

for i = 1:n

for j = 1:n

A1(i,j) = input('A1[i,j] = ');

end

end

for i = 1:n

b1(i) = input('b1[i] = ');

end

b1 = b1'; % b = b' - операція транспонування оператор ' транспонує матрицю.

for i = 1:n

c1(i) = input('c1[i] = ');

end

c1 = c1';

for i = 1:n

for j = 1:n

A2(i,j) = input('A2[i,j] = ');

end

end

for i = 1:n

for j = 1:n

B2(i,j) = input('B2[i,j] = ');

end

end

end

for i = 1:n

if(mod(i,2)==0)

b(i) = 1/(i^2+2+i);

else

b(i) = 1/i;

end

end

mess = 'b = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', b', '-append','delimiter', '\t', 'precision',15);

y1 = A\*b'; %y1 = A\*b' - множення матриці на стовбець, b - було рядком, після транспонації стало стовпцем

mess = 'y1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', y1, '-append');

temp0 = 2\*c1;

temp1 = b1-temp0;

mess = 'b1-2c1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp1, '-append');

y2 = A1\*temp1;

mess = 'y2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', y2, '-append');

for i = 1:n

for j = 1:n

C2(i,j) = 1/(i+j);

end

end

mess = 'C2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', C2, '-append','delimiter', '\t', 'precision',15);

temp21 = C2;

temp2 = B2-temp21;

mess = 'B2-C2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp2, '-append','delimiter', '\t', 'precision',15);

Y3 = A2\*temp2;

mess = 'Y3 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', Y3, '-append','delimiter','\t');

temp3 = Y3^2;

mess = 'Y3^2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp3, '-append','delimiter', '\t', 'precision',15);

temp4 = y2'\*temp3;

mess = 'y2\*Y3^2 =';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp4, '-append','delimiter', '\t', 'precision',15);

temp5 = temp3\*y1;

mess = 'Y3^2\*y1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp5, '-append','delimiter', '\t', 'precision',15);

temp6 = y2'\*y2\*y1;

mess = 'y2"\*y2\*y1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp6, '-append','delimiter', '\t', 'precision',15);

temp7 = y1+y2;

mess = 'y1+y2 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp7, '-append','delimiter', '\t', 'precision',15);

temp71 = Y3\*(y1+y2);

mess = 'Y3\*(y1+y2) = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', temp71, '-append','delimiter', '\t', 'precision',15);

temp111=temp6+temp5;

temp112=temp4'+temp71;

x = temp111+temp112;

mess = 'x = Y3^2\*y2+Y3\*(y1+y2)+y1\*y2\*y2"+Y3^2\*y1 = ';

dlmwrite('1.txt', mess,'-append', 'delimiter', '');

dlmwrite('1.txt', x, '-append','delimiter', '\t', 'precision',15);

6. Результати виконання програми

x = (y2"\*Y3^2\*y1+y1")\*(Y3\*y1+y2) =

(-8609716.51094907,

-6560922.29844907,

-14248736.7097917)

**Висновок:** В роботі використано паралелізм на рівні підзадач, оскільки передбачається, що кожен блок зі схеми декомпозиції є реалізований у виді функції. Це є середньоблоковий паралелізм. Обмін даними відбувається через використання спільних змінних. Присутня залежність даних між різними рівнями декомпозиції, але в межах одного рівня її немає. Є залежність за керуванням, оскільки послідовність обчислювального процесу наперед однозначно відома. Залежність за ресурсами та вводом/виводом може бути визначена лише у відношенні до певної обчислювальної системи.