

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Звіт до лабораторної роботи
з дисципліни "Розподілене та паралельне програмування" на тему
"Розв'язок задачі 2-SUM за допомогою послідовних та паралельних
обчислень"

Виконав студент 3 курсу факультету
комп'ютерних наук та кібернетики
спеціальності "Інформатика" групи ТК-31

Некряч Владислав Вадимович

КИЇВ-2022

Постановка задачі

Реалізувати послідовний та паралельні (MPI та OpenMP) алгоритми знаходження розв'язку задачі 2-SUM, порівняти час їхнього виконання та пришвидшення, яке дають паралельні алгоритми.

Алгоритм

Фіксуємо елемент x_i .

Знаходимо різницю $Total - x_i$, перебором шукаємо в масиві отримане число.

У паралельних алгоритмах здійснюємо те саме, тільки у кожному процесі внутрішній цикл(перебір) розбиваємо на частини, і якщо на вході маємо N процесів, то для конкретний процес з номером P буде опрацьовувати підмасив $[count * P / N, x_i * (P+1) / N]$.

Результати виконання

1) Залежність часу виконання (с) від розміру вхідних даних N

Кількість процесів для паралельних алгоритмів – 4

N	Послідовний	MPI	OpenMP
10^3	0.0023	0.0001	0.0009
10^4	0.2132	0.0096	0.0919
$5 \cdot 10^4$	5.3358	0.4686	1.9779
10^5	21.5348	1.8756	8.1044

2) Залежність часу виконання (с) від кількості процесів M

Розмір вхідних даних – 10^5

M	Послідовний	MPI	OpenMP
1	21.5348	–	–
2	–	4.2886	10.8109
4		2.1950	8.1044
8		1.8325	7.8399
16		1.7951	7.9568
32		1.7800	8.0660

3) Залежність прискорення від кількості процесів M

M	MPI	OpenMP
2	5.0214	1.9921
4	9.8108	2.6571
8	11.7515	2.7468
16	11.9964	2.7064
32	12.0982	2.6698

Висновки

Зі збільшенням розміру вхідних даних в усіх трьох випадках час виконання зростає квадратично.

Розпаралелювання алгоритму дає помітне пришвидшення – час виконання в обох випадках зі зростанням кількості процесів зменшується, асимптотично наближаючись до деякої величини, нижче якої він надалі після досягнення певної кількості процесів практично не падатиме через обмеженість зверху прискорення.

При цьому MPI-алгоритм працює набагато швидше за OpenMP-алгоритм і дає прискорення, що дуже швидко зростає зі збільшенням кількості процесів (але зупиняється на рівні 12х), тоді як OpenMP дає не дуже велике та практично незмінне значення прискорення близько 2.7х навіть зі зростанням кількості процесів.

Отже, MPI-версія алгоритму краще підходить для даної задачі і дає величезний виграш у часі, що й робить її використання найбільш доцільним для розв'язання.