Laborator 3

# Analiza cerintei

Programul construit realizeaza suma pe cifre a unor numere mari retinute ca liste. Executia este realizata atat secvential, cat si paralelizat in limbajul de programare C++. Scopul urmarit este de a compara eficienta ca timp a diferitelor metode de executie si realizare, documentate mai jos.

# Proiectare

Pentru varianta secventiala am facut o parcurgere completa a celor 2 numere citite din fisier, modificand pe parcurs rezultatul sumei cifrelor intr-o lista rezultat. In cazul variantei paralelizate, am adoptat varianta de scatter a listelor in p procese folosind MPI. Fiecare proces face calculele pe sublista alocata ei prin metoda MPI\_Scatterv (care permite subliste de lungimi variabile) si trimite procesului urmator un carry flag. Am evaluat atat varianta in care se trimite carry flag la inceputul calculelor, cat si la final, si am comparat rezultatele. La final, programul verifica validitatea calculelor efectuate, comparand cu varianta secventiala si returneaza un calcul al timpului necesar executiei in milisecunde.

# Cazuri de testare

1. Numar 1 = “123456789123456789” = Numar2
2. N\_1=1000 si N\_2=1000 (random digits)
3. N\_1=100 si N\_2=100000 (random digits)

Fiecare caz a fost rulat secvential, pe 4 si respectiv 8 threaduri.

# Analiza rezultatelor

In urma rezultatelor obtinute, putem constata ca varianta cu trimiterea carry flag-ului la inceput nu utilizeaza resursele paralelizate disponibile, deoarece fiecare proces este nevoit sa astepte incheierea celui anterior (pentru a primi carry flag). Din acest motiv, performanta sufera la cresterea numarului de procese. O imbunatatire a rezultatelor se observa daca trimiterea carry flag-ului se efectueaza doar la finalul calculelor, fiind aplicate modificarile necesare. In aceasta situatie, un numar echilibrat de procese pentru volumul de date ofera o imbunatatire a timpului de executie.

|  |  |  |
| --- | --- | --- |
| Lungime nr | Nr threads | Timp executie |
| N1=18 N2=18 | 1 | 18.29622 |
| N1=18 N2=18 | 4 | 23.74237 |
| N1=18 N2=18 | 8 | 35.76878 |
|  |  |  |
| N1=1000 N2=1000 | 1 | 28.7606 |
| N1=1000 N2=1000 | 4 | 33.41518 |
| N1=1000 N2=1000 | 8 | 44.08617 |
|  |  |  |
| N1=100 N2=100000 | 1 | 601.8528 |
| N1=100 N2=100000 | 4 | 607.2569 |
| N1=100 N2=100000 | 8 | 614.8958 |

## Rezultate Varianta 2 (fiecare thread asteapta carry la inceput)

## Rezultate Varianta 2 (fiecare thread primeste carry la final)

|  |  |  |
| --- | --- | --- |
| Lungime nr | Nr threads | Timp executie |
| N1=18 N2=18 | 1 | 17.59998 |
| N1=18 N2=18 | 4 | 22.23501 |
| N1=18 N2=18 | 8 | 30.32072 |
|  |  |  |
| N1=1000 N2=1000 | 1 | 26.14945 |
| N1=1000 N2=1000 | 4 | 31.11414 |
| N1=1000 N2=1000 | 8 | 36.73844 |
|  |  |  |
| N1=100 N2=100000 | 1 | 616.111 |
| N1=100 N2=100000 | 4 | 677.1834 |
| N1=100 N2=100000 | 8 | 645.9803 |

## Rezultate Varianta 1

|  |  |  |
| --- | --- | --- |
| Lungime nr | Nr threads | Timp executie |
| N1=18 N2=18 | 1 | 17.99368 |
| N1=18 N2=18 | 4 | 20.39831 |
| N1=18 N2=18 | 8 | 32.32973 |
| N1=1000 N2=1000 | 1 | 24.63315 |
| N1=1000 N2=1000 | 4 | 27.81015 |
| N1=1000 N2=1000 | 8 | 36.31948 |
| N1=100 N2=100000 | 1 | 565.3457 |
| N1=100 N2=100000 | 4 | 558.731 |
| N1=100 N2=100000 | 8 | 571.4189 |