

Laborator 1 (PPD)

Implementati in Java si C++ (11 sau >11)

- A) adunarea a doua 'numere mari'
- B) inmultirea a doua 'numere mari'

fiecare in 2 variante:

1) sequential

2) paralel - folosind multithreading

- 2.1 varianta simpla
- 2.2 varianta optimizata

2.1 Varianta paralela simpla – considera rezolvarea problemei prin executia a 2 etape:

- prima etapa imparte cifrele numerelor echilibrat pe threaduri si face sumele cifrelor corespunzatoare calculand un 'carry'
- a doua etapa actualizeaza rezultatul folosind carry-urile calculate in prima etapa (aceasta actualizare se poate face succesiv de la un thread la altul fara sa se faca efectiv calcul in paralel)

2.2 Varianta paralela optimizata – fie optimizeaza varianta anterioara prin paralelizarea etapei a doua, fie considera un alt algoritm de calcul care permite paralelizari mai eficiente (evidentiat print imp de executie mai bun).

'numar mare' = numar cu mai mult de 7 cifre reprezentat ca un tablou de cifre (numere intregi fara semn - byte) in care cifra cea mai nesemnificativa este pe prima pozitie.

-Numarul de threaduri p trebuie sa fie un parametru care poate fi citit (modificat) inainte de inceperea executiei.

Datele de intrare corespunzatoare numerelor *se vor citi din fisiere* (care au fost anterior create folosind generare aleatoare de numere)!

Programul va afisa la sfarsit timpul global (total) T de executie corespunzator

- A) operatiei de adunare
- B) operatiei de inmultire

($T = \text{timp_evaluat_dupa_terminarea_executiei_tuturor_threadurilor} - \text{timp_evaluat_inainte_de_pornirea_threadurilor}$)

Important: folositi o incarcare echilibrata de calcul pe fiecare thread.

Balanced Distribution (dimensiunea datelor pe care lucreaza fiecare thread este aproximativ egala).

Documentatia include explicatii referitoare la proiectare si testare:

Proiectare:

Explicatia algoritmului de calcul.

Optimizari realizate.

Testarea va include:

-variante orientate pe testarea corectitudinii (dimensiuni relativ mici; de exemplu ~ 7 cifre)

-variante orientate pe masurarea performantei (dimensiuni relativ mari; de exemplu > 1000 cifre);

- verificarea egalitatii dintre rezultatul executiei secventiale si a celei paralele (folositi functii de la L0) – obligatorie pentru fiecare test cu numere cu mai mult de 7 cifre

Pentru fiecare testare care trece testul de corectitudine includeti in documentatie informatii despre:

Dim_numere(nr cifre) | Obs- referitoare la limbaj si la sistemul pe care s-a facut testarea

Timp executie secventiala

Nr_threaduri | Timp de executie paralela |

**1
2
4
6
8**

Deadline:

A) Saptamana 4 (Java si C++11)

B) Saptamana 5 (Java si C++11)