

Laborator MPI

(Problemele se distribuie folosind formula "nr_crt%4+1", unde "nr_crt" este numarul in lista studentilor din fiecare grupa.)

Observatii: se pot folosi functii de comunicare globala acolo unde este cazul: MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Reduce.

Predare: saptamana 13/14

I.

Se considera un fisier "input.txt" care contine numere intregi.

Scrieti un program MPI prin care se citeste un numar si verifica daca acel numar se gaseste in lista numerelor citite si daca da pe ce pozitii.

Output:

Var 1) DA pozitia "?" [pozitia "?" pozitia"?"] ...]

Var 2) NU

Cerinte:

Etapa 1:

- procesul 0 citeste numarul cautat si apoi citeste toate numerele din fisierul "input.txt" si le stocheaza intr-un vector cu nume "date".;
- procesul 0 distribuie parti (aproximativ) egale din vectorul "date" astfel incat fiecare proces va avea un vector "segment" populat cu numere din vectorul "date".;
- procesul 0 distribuie prin broadcast valoarea cautata.

Etapa 2 :

- fiecare proces cauta in propriul "segment" valoarea cautata;
- procesul 0 primeste de la fiecare proces rezultatul cautarii care va fi reprezentat printr-un numar cu valoarea 0 daca nu a fost gasit, sau egal cu numarul de pozitii pe care a fost gasit;
- procesul 0 primeste de la fiecare proces pozitiile pe care a fost gasit numarul.

Etapa 3:

- procesul 0 afiseaza rezultatul (Output).

II. Se considera un fisier "input.txt" care contine numere intregi.

Scrieti un program MPI prin care se citeste un numar natural - a - si modifica fiecare numar din fisierul dat conform functiei $f(x, a) = (x*x*x*a - x*x*a*a - a)$. Rezultatul va fi salvat in fisierul "output.txt". Se va afisa si cate dintre numerele calculate sunt negative.

Cerinte:

Etapa 1:

- procesul 0 citeste numarul a si apoi citeste toate numerele din fisierul "input.txt" si le stocheaza intr-un vector cu nume "date".;

- procesul 0 distribuie parti (aproximativ) egale din vectorul "date" astfel incat fiecare proces va avea un vector "segment" populat cu numere din vectorul "date";

-procesul 0 distribuie prin broadcast valoarea a.

Etapă 2 :

- fiecare proces modifica numerele din propriul "segment" conform functiei si numara cate dintre acestea sunt negative;
- procesul 0 primeste de la fiecare proces segmentul corespunzator modificat;
- procesul 1 primeste de la fiecare proces cate valori negative a calculat.

Etapă 3:

- procesul 0 scrie rezultatul in fisierul "output.txt".
- procesul 1 calculeaza numarul total de valori negative si il afiseaza.

III

Se considera un fisier "input.txt" care contine n numere reale ($n > 10000$).

Se da polinomul P cu coeficientii $a_0, a_1, a_2, a_3, \dots, a_N$, in fisierul "polinom.txt". Se cere sa se calculeze valorile polinomului in punctele citite din fisierul "input.txt". Valorile vor fi salvate in fisierul "output.txt".

Cerinte:

Etapă 1:

- procesul 0 citeste coeficientii polinomului din fisierul "polinom.txt";
- procesul 0 distribuie prin broadcast coeficientii polinomului.
- procesul 0 citeste toate numerele din fisierul "input.txt" si le stocheaza intr-un vector cu nume "date".;
- procesul 0 distribuie parti (aproximativ) egale din vectorul "date" astfel incat fiecare proces contine un vector "segment" cu parti din vectorul "date";

Etapă 2 :

- fiecare proces calculeaza valorile polinomului in punctele din propriul vector "segment";
- procesul 0 primeste de la fiecare proces valorile calculate;

Etapă 3:

- procesul 0 scrie rezultatul in fisierul "output.txt".

IV

Se considera un fisier "input.txt" care contine n numere reale ($n > 10000$).

Se da polinomul P cu coeficientii $a_0, a_1, a_2, a_3, \dots, a_N$, in fisierul "polinom.txt" (pentru testare se vor considera cazurile $N=4, 8, 12$). Se cere sa se calculeze valorile polinomului in punctele citite din fisierul "input.txt". Valorile vor fi salvate in fisierul "output.txt".

Cerinte:

- Se creeaza N procese!
- Se foloseste schema lui Horner de calculare a valorii polinomului intr-un punct: $P(x) = ((a_0 \cdot x + a_1) \cdot x + \dots + a_{(N-1)} \cdot x) + a_N$

Etapă 1:

- procesul 0 citește coeficienții polinomului din fișierul "polinom.txt";
- procesul 0 distribuie procesului 1 valoarea a_1 , procesului 2 valoarea a_2 , etc.

- va avea un vector "segment" populat cu numere din vectorul "date";

Etapă 2 : se repetă următoarele operații până când se termină valorile din fișierul "input.txt"

- a) procesul 0 citește următoarea valoare (x) din fișier și o transmite prin broadcast tuturor celorlalte procese;
- b) fiecare proces cu $id > 0$ primește o valoare v de la procesul precedent; dacă $id=0$ atunci $v=a_0$;
- c) fiecare proces înmulțește valoarea sa v cu x și adună coeficientul distribuit lui la rezultat;
- d) fiecare proces trimite valoarea calculată la procesul următor (procesul cu id cu o unitate mai mare) dacă acesta există, sau scrie valoarea în fișierul "output.txt" dacă este ultimul proces.