

Arhitectura sistemelor de calcul paralel - examen

20 iunie 2022

Durata examenului este de două ore. Tratați următoarele subiecte (soluția dumneavoastră va fi sub forma unor coduri sursă C MPI).

1. Scrieți un program C MPI care să asigure următoarea funcționalitate:

- (a) fiecare proces definește două șiruri de întregi *send_data[10]*, *recv_data[10]*, primul fiind inițializat cu valori succesive, începând cu rangul procesului, al doilea inițializat cu 0;
- (b) pe o topologie inelară, fiecare proces trimite către procesul de rang imediat superior valorile șirului *send_data* și recepționează în *recv_data* valorile primite de la procesul cu rang imediat inferior (veți utiliza *MPI_Sendrecv()*);
- (c) procesul cu rang 0 afișează la final conținutul șirurilor *send_data[10]*, *recv_data[10]* pe care le deține.
- (d) Funcții MPI sugerate: *MPI_Init()*, *MPI_Comm_size()*, *MPI_Comm_rank()*, *MPI_Sendrecv()*, *MPI_Finalize()* **(4p)**

2. Se dă fragmentul de cod C MPI *exam_comm_mpi.c* atașat. Programul trebuie să calculeze într-o implementare paralelizată valoarea medie

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

și varianța

$$\sigma = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2 \quad (2)$$

unui șir de N valori *double*, disponibil în fisierul text *input_vec.dat*, atașat de asemenea. Veți construi programul, introducând în locurile corespunzătoare, marcate cu simbolurile

/*****

codul dumneavoastră, astfel încât să asigurați următoarea funcționalitate:

- (a) Veți grupa procesele lansate în execuție (se va presupune ca numărul lor *nproc* este par și, în plus, că $\frac{nproc}{2}$ este divizor al lui N) în două subgrupuri distincte, respectiv subgrupul *group1* al proceselor cu ranguri pare în comunicatorul global *MPI_COMM_WORLD* și subgrupul *group2* al proceselor cu ranguri impare. Primul subgrup va calcula media șirului, iar cel de-al doilea varianța lui;
- (b) Veți crea doi noi comunicatori, *comm1* și *comm2*, pe baza comunicatorului standard *MPI_COMM_WORLD*, subgrupurile asociate fiind *group1* și *group2*;
- (c) Procesele cu rangul nou *new_rank = 0* din noii comunicatori citesc fișierul de date în blocurile *data* (pe care le alocă numai ele!) pe care le redistribuie în mod egal în interiorul acestor noi comunicatori cu *MPI_Scatter()*.
- (d) Procesele din primul grup calculează sumele parțiale $\sum_i^{\frac{2N}{nproc}} local_data[i]$, acumulate de procesul cu rangul *new_rank = 0* din *comm1*, care calculează valoarea medie μ , distribuită apoi cu *MPI_Bcast()* **tuturor proceselor în comunicatorul global *MPI_COMM_WORLD***;
- (e) Procesele din al doilea grup calculează (după finalizarea *MPI_Bcast()*) sumele parțiale $\sum_i^{\frac{2N}{nproc}} (local_data[i] - \mu)^2$, acumulate de procesul cu rangul *new_rank = 0* din *comm2*, care calculează varianța σ ;
- (f) Cele două procese cu rangul *new_rank = 0* afișează valorile calculate.
- (g) Funcții MPI sugerate: *MPI_Init()*, *MPI_Comm_size()*, *MPI_Comm_rank()*, *MPI_Comm_group()*, *MPI_Group_incl()*, *MPI_Comm_create()*, *MPI_Scatter()*, *MPI_Bcast()*, *MPI_Reduce()*, *MPI_Finalize()* **(5p)**.

Se acordă **1p** din oficiu. Funcțiile MPI indicate mai sus sunt doar sugestii de lucru. Orice soluție corectă, care asigură funcționalitatea cerută, va fi luată în considerare.

Succes!