

Arhitectura sistemelor de calcul paralel - examen

14 septembrie 2020

Durata examenului este de două ore. Tratați următoarele subiecte și transmiteți soluția dumneavoastră (coduri sursă C MPI) prin e-mail, la adresele lucian@solid.fizica.unibuc.ro, lucian.ion@g.unibuc.ro în intervalul 15:00 - 15:30.

Mesajele electronice trimise după ora 15:30 nu vor fi luate în considerare.

1. Scrieți un program C MPI care să poată fi rulat cu patru procese și să asigure următoarea funcționalitate:

- (a) Toate procesele își stochează rangul în variabila *my_rank* și declară un șir *int data[4]*;
- (b) Procesul cu rangul 0 inițializează fiecare element al acestui șir cu valoarea 100, apoi trimite șirul către procesul de rang 1, care scrie valoarea $data[1] * (my_rank + 1)$ în $data[1]$, apoi trimite șirul către procesul cu rangul 2 care scrie valoarea $data[2] * (my_rank + 1)$ în $data[2]$, ș.a.m.d., pe o topologie inelară;
- (c) la ultima etapă, procesul cu rangul 3 trimite șirul completat cu toate valorile către procesul cu rangul 0, care îl afișează, în formatul $data[\%d]=\%d\backslash n$.
- (d) Funcții MPI sugerate: *MPI_Init()*, *MPI_Comm_size()*, *MPI_Comm_rank()*, *MPI_Send()*, *MPI_Recv()*, *MPI_Finalize()* (**4p**).

2. Programul secvențial *exam_int_T1_serial.c*, atașat, calculează numeric integrala

$$\int_a^b e^{-x^2} \sin\left(\pi \frac{x-a}{b-a}\right) dx,$$

în care $a = 0$, $b = 10$. Vi se cere să paralelizați acest cod, distribuind celor $nproc$ procese lansate în paralel o partiție a intervalului de integrare în subintervale de lungime tipică $\frac{b-a}{nproc}$. Ultimul subinterval va avea o lungime diferită dacă $nproc$ nu este divizor al lui $(b-a)$! Veți proceda astfel:

- (a) Fiecare proces (inclusiv procesul cu rangul 0) își va defini limitele locale de integrare în funcție de rang;
- (b) Partiția inițială va fi: procesul cu rangul 0 tratează primul subinterval $(a, a + \frac{b-a}{nproc})$, procesul cu rangul 1 pe al doilea, etc.;
- (c) Urmează partea secvențială, în care va fi aplicat algoritmul de integrare exact la fel ca în programul secvențial, însă valorile calculate vor fi stocate în variabila *local.IT*;
- (d) Valorile locale vor fi acumulate de procesul cu rangul 0 în variabila *IT* (o soluție elegantă constă în utilizarea funcției *MPI_Reduce()*);
- (e) Procesul cu rangul 0 afișează rezultatul final;
- (f) Funcții MPI sugerate: *MPI_Init()*, *MPI_Comm_size()*, *MPI_Comm_rank()*, *MPI_Reduce()*, *MPI_Finalize()* (**5p**)

Întrucât sunt utilizate funcții și variabile definite în biblioteca matematică, compilarea se face cu:

```
mpicc -o nume_program fisier_sursa.c -lm
```

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!