

Arhitectura sistemelor de calcul paralel - examen

15 iunie 2021

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 12:00 - 12:30.

Mesajele electronice trimise după ora 12:30 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 declară un sir `int ranks[4];`;
 - (b) Procesul cu rangul 0 initializează acest sir cu 0, apoi trimit valorile către procesul de rang 1, care își scrie valoarea rangului în `ranks[1]`, apoi trimit valorile către procesul cu rangul 2 care își scrie valoarea rangului în `ranks[2]`, s.a.m.d., pe o topologie inelară;
 - (c) la ultima etapă, procesul cu rangul 3 trimit sirul completat cu toate valorile către procesul cu rangul 0, care îl afișează, în formatul `"ranks[%d]=%d\n"`.
 - (d) Funcții MPI sugerate: `MPI_Init()`, `MPI_Comm_size()`, `MPI_Comm_rank()`, `MPI_Send()`, `MPI_Recv()`, `MPI_Finalize()` (4p).
2. Algoritmul QAGS este o procedură adaptativă de integrare numerică a funcțiilor cu o singularitate integrabilă în domeniul de integrare. Domeniul de integrare este divizat în subintervale și la fiecare iterare subintervalul care dă cea mai mare eroare estimată este subdivizat în două părți egale. În felul acesta eroarea este redusă rapid, pe măsură ce subintervalele se concentrează în jurul punctelor dificile din domeniu (aici singularitatea, aproximările succesive asigurând convergența rapidă a integralei). Programul secvențial `int_QAG.c`, atașat, calculează numeric integrala:

$$\int_0^1 \frac{\ln(x)}{\sqrt{x}} dx, \quad (1)$$

care prezintă o singularitate logaritmică în 0. Vi se cere să-l paralelizați, astfel încât să asigure următoarea funcționalitate:

- (a) Vor fi lansate în execuție `nproc = 4` procese, procesul cu rangul 0 (procesul master) definind valorile a și b și distribuind inițial o partitie uniformă a intervalului de integrare (a, b) (patru subintervale egale, fiecare de lungime $\frac{b-a}{nproc}$);
- (b) O soluție posibilă ar fi ca fiecare proces să definească un sir `double local_lims[2]`; în care să recepționeze limitele de integrare inferioară și superioară de la procesul cu rang 0 (evident, acesta își va defini local limitele de integrare);
- (c) Partitia inițială va fi: procesul cu rangul 0 tratează primul subinterval, procesul cu rangul 1 pe al doilea, etc.;
- (d) Toate procesele vor defini suplimentar variabilele `double local_result`, `local_error`; care vor stoca, respectiv, rezultatul și eroarea estimată pentru calculul local;
- (e) 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 `local_result` și `local_error`;
- (f) Valorile locale vor fi acumulate de procesul cu rangul 0 în variabilele `result` și `error` (o soluție elegantă constă în utilizarea funcției `MPI_Reduce()`);
- (g) Procesul cu rangul 0 afișează rezultatul final și estimarea marginii superioare a erorii;
- (h) Funcții MPI sugerate: `MPI_Init()`, `MPI_Comm_size()`, `MPI_Comm_rank()`, `MPI_Send()`, `MPI_Recv()`, `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. Întrucât sunt utilizate implementări ale algoritmului QAG incluse în biblioteca GSL, compilarea se face cu:

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

Dacă sistemul dumneavoastră de operare este Linux Ubuntu, compilați cu:

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

Succes!