

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 două procese și să asigure următoarea funcționalitate:
 - (a) Procesul cu rangul 1 trimite către procesul cu rangul 0 mesajul (stringul): "Procesul cu rangul 1 este activ";
 - (b) Procesul cu rangul 0 recepționează mesajul și îl afișează;
 - (c) Funcții MPI sugerate: *MPI_Init()*, *MPI_Comm_size()*, *MPI_Comm_rank()*, *MPI_Send()*, *MPI_Recv()*, *MPI_Finalize()* (4p).
2. Un *wavelet* este un pachet de unde (Fig. 1) utilizat în procesarea semnalelor. Transformata wavelet continuă a semnalului $f(t)$ este definită de:

$$w(s, \tau) = \int_{-\infty}^{\infty} f(t) \psi_{s,\tau}^*(t) dt, \quad (1)$$

în care $\psi_{s,\tau}(t)$ sunt funcțiile wavelet de bază. Transformata inversă este:

$$f(t) = \int_0^{\infty} ds \int_{-\infty}^{\infty} w(s, \tau) \psi_{s,\tau}(t) d\tau. \quad (2)$$

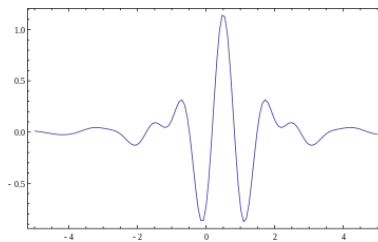


Figura 1: Wavelet - semnal cu amplitudine variabilă, care crește de la zero și scade înapoi la zero, utilizat în procesarea semnalelor.

Varianta discretă utilizată în analiza numerică a semnalelor este construită pe baza setului ortonormat $\{\psi_{m,n}(t) = \frac{1}{\sqrt{a^m}} \psi\left(\frac{t-nb}{a^m}\right)\}$ cu $m, n \in \mathbb{Z}$, $a > 1$, $b > 0$ ($\psi(t)$ se numește *pachetul mamă* - *mother wavelet*).¹ Codul sursă C *wavelet.c* disponibil alăturat implementează o procedură de filtrare a unui semnal disponibil în fișierul *input.dat* folosind primele 20 cele mai mari componente ale transformatei wavelet. Semnalul procesat este scris apoi în fișierul *wavelet.signal.dat*. Vi se cere să paralelizați acest program, astfel încât să funcționeze cu 4 procese, cu următoarea procedură:

- (a) toate procesele declară *int nc=5* și suplimentar variabila *int nl* și blocul *double * ldata* cărui i se alocă spațiu în memoria liberă pentru $nl = n/nproc$ valori *double*;
- (b) toate procesele alocă pentru *abscoeff* spațiu în memoria liberă pentru nl valori *double*;
- (c) toate procesele alocă pentru *p* spațiu în memoria liberă pentru nl valori *size_t*;

¹Textul referitor la transformata wavelet are un caracter pur informativ. Nu este necesar să cunoașteți mai multe detalii despre subiect pentru a rezolva problema propusă.

- (d) procesul cu rangul 0 este cel care efectuează operațiile I/O (citește fișierul de intrare în *data* pentru care alocă spațiu pentru n valori *double*, scrie fișierul de ieșire)
- (e) procesul cu rangul 0 trimite segmente egale din buffer-ul *data* către toate procesele (*MPI_Scatter()* este o bună alegere);
- (f) urmează partea secvențială, în care fiecare proces aplică transformata wavelet directă asupra valorilor locale *ldata* (atenție că acest bloc conține nl valori!), filtrează primele $nc = 5$ componente și aplică transformata wavelet inversă, la fel ca în codul secvențial;
- (g) rezultatul este colectat de procesul cu rangul 0 înapoi în blocul *data* (funcția *MPI_Gather()* este o bună alegere), care îl scrie apoi în fișierul de ieșire;
- (h) datele de intrare și de ieșire pot fi vizualizate cu *gnuplot* direct sau cu funcțiile de bibliotecă de interfață cu *gnuplot* create la începutul semestrului;
- (i) Funcții MPI sugerate: *MPI_Init()*, *MPI_Comm_size()*, *MPI_Comm_rank()*, *MPI_Scatter()*, *MPI_Gather()*, *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 transformatei wavelet 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!