

Laborator 1

Se considera o imagine reprezentata printr-o matrice de pixeli, F , de dimensiune $(N \times M)$.

Se cere transformarea ei aplicand o filtrare cu o fereastră definită de multimea de indici W cu coeficientii w_{kl} (reprezentati prin matricea $W[k,l]$, unde $0 \leq k < n$, $0 \leq l < m$; si $n < N$, $m < M$).

Transformarea unui pixel:

$$v(m, n) = \sum_{(k,l) \in W} w_{kl} f(m - k, n - l)$$

De exemplu:

$$W = \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & \boxed{1/9} & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix}$$

Se cere asigurarea următoarei postcondiție:

Postcondiție: Matricea rezultat V conține imaginea filtrată a imaginii inițiale F ($V \leq F$)

- A) Program secvențial
- B) Program paralel: folosiți **p** threaduri pentru calcul..

Obiectiv: Impartire cat mai echilibrata a calculul pe threaduri!

Datele de intrare se citesc dintr-un fisier de intrare "date.txt".

(Fisierul trebuie creat anterior prin adaugare de numere generate aleator.)

Implementare:

- a) Java
- b) C++ (cel puțin C++11)
 - i. matricile sunt alocate static
 - ii. matricile sunt alocate dinamic

Testare: masurati timpul de executie pentru

- 1) $N=M=10$ si $n=m=3$; $p=4$;
- 2) $N=M=1000$ si $n=m=5$; $p=2,4,8,16$
- 3) $N=10$ $M=10000$ si $n=m=5$; $p=2,4,8,16$
- 4) $N=10000$ $M=10$ si $n=m=5$; $p=2,4,8,16$

Observatii:

- Fiecare test trebuie repetat de 5 ori si pentru evaluarea timpului de executie se considera media aritmetica a celor 5 rulari.
- Pentru fiecare varianta a cazului de testare 1) folosiți același fisier "date.txt";
- similar pentru cazurile 2), 3) si 4)
- Pentru programul C++ testati si comparati:

Analiza

Comparati performanta pentru fiecare caz – secvențial versus paralel si variantele paralele intre ele.

Comparati timpii obtinuti cu implementarea Java versus implementarea C++.

Comparati cele doua variante pentru implementarea C++

Deadline: saptamana 4