Lab 1 – Farcas Tudor

**1. Analiza cerințelor**

Tema de laborator presupune calculul matricei V rezultată în urma aplicării convoluției matricei de convoluție C pe matricea F. Matricea V are dimensiunile n x m, iar matricea de convoluție C are dimensiunea k x k, unde k este un număr impar. Se consideră că matricea F este bordată virtual cu elemente de pe frontiera. Scopul cerintei este să implementam aceasta operație, atât într-un program secvențial cât și într-un program paralel folosind p threaduri.

**2. Proiectare**

**Structuri de date:**

* **Matrix\_V(n, m)**: Matricea rezultată în urma convoluției.
* **Matrix\_C(k, k)**: Matricea de convoluție.
* **Matrix\_V(n,m)**: Matricea rezultata in urma convolutiei.

**Partitionare pe threaduri:**

* Împărțirea matricei V între cele p threaduri va fi realizată astfel încât fiecare thread să proceseze o anumită parte a matricei V. Se va evita suprapunerea datelor procesate de threaduri. Aloc fiecarui thread un anumit numar de linii sau coloane, in functie daca realizam parcurgerea orizontala sau verticala.

**Clase și funcții:**

* **ThreadMatrix**: Clasa pentru crearea threadurilor, aplicand algoritmul convolutional in interiorul functiei run()
* Functii precum “convolution“ care au rolul de a calcula prin intermediul algoritmului convolutional, matricea rezultata in c++, deoarece in c++ nu folosim clase pentru threaduri.
* **Relații:**
* Funcțiile și thread-urile vor colabora pentru a efectua convoluția și pentru a asigura o gestionare corecta a datelor. Threadurile gestionand fiecare un anumit numar de linii, daca e parcurgere orizontala, iar daca e verticala, un anumit numar de coloane.
* Fiecare thread se ocupa de un anumit numar de linii din matrice.

**3. Detalii de implementare**

**Algoritm:**

Pentru fiecare element V(i, j), se va aplica formula de convoluție specificată în cerință. Se va itera prin matricea V, calculând suma ponderată a elementelor vecine cu matricea de convoluție C.

**Gestionarea datelor:**

* Datele vor fi citite de la intrare și vor fi stocate în structurile de date corespunzătoare.
* Rezultatele vor fi stocate într-o matrice V noua.
* Grija la respectarea regulilor de bordare virtuală.
* La alocare statica, am ales valoarea maxima de 10000 pentru matrici.

**4. Cazuri de testare:**

**a) C++**

|  |  |  |  |
| --- | --- | --- | --- |
| Tip Matrice | Tip alocare | Nr threads | Timp executie |
| N=10 M=10 K=3 | dinamic | secvential | 4390 |
| N=10 M=10 K=3 | dinamic | 4 | 726590 |
| N=10 M=10 K=3 | dinamic | 4 | 806580 |
| N=10 M=10 K=3 | dinamic | 4 | 788590 |
| N=10 M=10 K=3 | static | 4 | 782960 |
| N=10 M=10 K=3 | static | 4 | 770170 |
| N=10 M=10 K=3 | static | secvential | 23540 |
| N=1000 M=10 K=5 | dinamic | secvential | 852470 |
| N=1000 M=10 K=5 | static | 2 | 4688730 |
| N=1000 M=10 K=5 | static | 2 | 2364000 |
| N=1000 M=10 K=5 | static | secvential | 2653450 |
| N=1000 M=10 K=5 | dinamic | secvential | 862330 |
| N=1000 M=10 K=5 | static | 4 | 4357300 |
| N=1000 M=10 K=5 | static | 4 | 2286030 |
| N=1000 M=10 K=5 | static | secvential | 2600830 |
| N=1000 M=10 K=5 | dinamic | secvential | 851210 |
| N=1000 M=10 K=5 | static | 8 | 4285940 |
| N=1000 M=10 K=5 | static | 8 | 2303290 |
| N=1000 M=10 K=5 | static | secvential | 2615580 |
| N=1000 M=10 K=5 | dinamic | secvential | 878090 |
| N=1000 M=10 K=5 | static | secvential | 2707280 |
| N=1000 M=10 K=5 | static | 16 | 2807470 |
| N=1000 M=10 K=5 | static | 16 | 5047140 |
| N=10 M=1000 K=5 | dinamic | secvential | 825500 |
| N=10 M=1000 K=5 | static | 2 | 1431750 |
| N=10 M=1000 K=5 | static | 2 | 1285440 |
| N=10 M=1000 K=5 | static | secvential | 831190 |
| N=10 M=1000 K=5 | static | 2 | 4097410 |
| N=10 M=1000 K=5 | static | 4 | 3981120 |
| N=10 M=1000 K=5 | static | 4 | 1347780 |
| N=10 M=1000 K=5 | static | secvential | 819930 |
| N=10 M=1000 K=5 | dinamic | secvential | 827940 |
| N=10 M=1000 K=5 | static | 8 | 1590030 |
| N=10 M=1000 K=5 | static | 8 | 4380420 |
| N=10 M=1000 K=5 | static | secvential | 821220 |
| N=10 M=1000 K=5 | dinamic | secvential | 859600 |
| N=10 M=1000 K=5 | dinamic | secvential | 846660 |
| N=10 M=1000 K=5 | static | secvential | 872700 |
| N=10 M=1000 K=5 | static | 16 | 2116780 |
| N=10 M=1000 K=5 | static | 16 | 5261520 |
| N=1000 M=1000 K=5 | dinamic | secvential | 84362120 |
|  |  |  |  |
| N=1000 M=1000 K=5 | static | secvential | 83842890 |
| N=1000 M=1000 K=5 | static | 16 | 1.07E+08 |
| N=1000 M=1000 K=5 | static | 16 | 24268280 |
| N=1000 M=1000 K=5 | static | secvential | 83366750 |

b) Java

|  |  |  |
| --- | --- | --- |
| Tip Matrice | Nr threads | Timp executie |
| N=10 M=10 K=3 | both | 0.04774 |
| N=10 M=10 K=3 | both | 0.04946 |
| N=10 M=10 K=3 | 4 | 0.80748 |
| N=5 M=5 K=3 | 4 | 0.74561 |
| N=5 M=5 K=3 | 4 | 0.67829 |
| N=5 M=5 K=3 | 4 | 0.72172 |
| N=5 M=5 K=3 | 4 | 0.59285 |
| N=5 M=5 K=3 | 4 | 0.6047 |
| N=5 M=5 K=3 | 4 | 1.05817 |
| N=5 M=5 K=3 | paralel | 0.93796 |
| N=5 M=5 K=3 | both | 0.9208 |
| N=5 M=5 K=3 | both | 0.79463 |
| N=5 M=5 K=3 | both | 0.81819 |
| N=5 M=5 K=3 | both | 0.84711 |
| N=5 M=5 K=3 | both | 1.01689 |
| N=5 M=5 K=3 | both | 0.88959 |
| N=5 M=5 K=3 | both | 0.81592 |
| N=5 M=5 K=3 | both | 0.78051 |
| N=5 M=5 K=3 | both | 0.76303 |
| N=5 M=5 K=3 | both | 0.78333 |
| N=5 M=5 K=3 | both | 0.79576 |
| N=5 M=5 K=3 | secvential | 0.01656 |
| N=5 M=5 K=3 | paralel | 2.59966 |
| N=5 M=5 K=3 | both | 1.61405 |
| N=5 M=5 K=3 | paralel | 0.85435 |
| N=5 M=5 K=3 | both | 1.51109 |
| N=10 M=10 K=3 | secvential | 0.04664 |
| N=1000 M=1000 K=5 | secvential | 59.89453 |
| N=1000 M=1000 K=5 | paralel | 116.152 |
| N=1000 M=1000 K=5 | both | 182.4656 |
| N=1000 M=1000 K=5 | both | 206.3115 |
| N=1000 M=1000 K=5 | paralel | 139.6988 |
| N=1000 M=1000 K=5 | secvential | 67.7311 |
| N=1000 M=1000 K=5 | secvential | 67.17222 |
| N=1000 M=1000 K=5 | paralel | 139.622 |
| N=1000 M=1000 K=5 | both | 208.6609 |
| N=1000 M=10 K=5 | both | 19.57819 |
| N=1000 M=10 K=5 | paralel | 13.44314 |
| N=1000 M=10 K=5 | secvential | 4.57422 |
| N=1000 M=10 K=5 | secvential | 4.99324 |
| N=1000 M=10 K=5 | paralel | 12.82876 |
| N=1000 M=10 K=5 | both | 18.79998 |
| N=10 M=1000 K=5 | both | 7.85342 |
| N=10 M=1000 K=5 | both | 8.67788 |
| N=10 M=1000 K=5 | both | 6.73391 |
| N=10 M=1000 K=5 | both | 6.25126 |
| N=10 M=1000 K=5 | paralel | 1.17446 |
| N=10 M=1000 K=5 | paralel | 1.40956 |
| N=10 M=1000 K=5 | paralel | 2.33081 |
| N=10 M=1000 K=5 | paralel | 3.46435 |