

Proiect APD

Dominte Mihai

CR3.1B

Introducere

In cadrul acestui proiect am realizat un algoritm de sortare a unui sir de numere folosind metoda Bubble Sort si am studiat performanta de rulare a acestuia pe diferite esantioane de numere.

Algoritmul foloseste o functie de generare a numerelor pentru a constitui vectorul ce trebuie sortat. A fost folosita functia rand() pentru generarea de numere aleatoare.

Functia de sortare prin metoda bubble sort este cronometrata folosind biblioteca <chrono> rezultatul obtinut fiind afisat in microsecunde.

Functia de afisare este optionala si afiseaza numerele sortate din vector.

Algoritm, Limbaj, Masina:

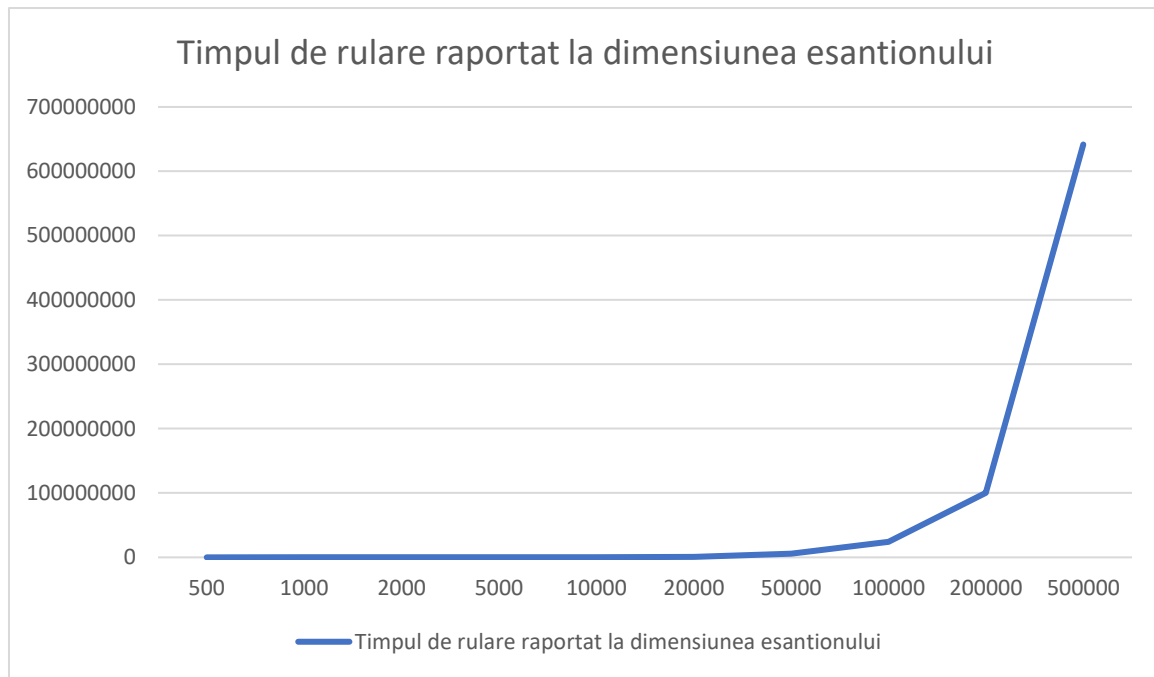
-Ca metoda de sortare a fost folosit Bubble Sort, algoritmul complet fiind prezentat mai jos;

-Limbajul folosit a fost C++, codul a fost scris folosind Visual Studio 2019

-Masina pe care a fost rulat foloseste Windows 10, 32GB RAM, procesor AMD Ryzen 2700X cu 8 nuclee

Teste:

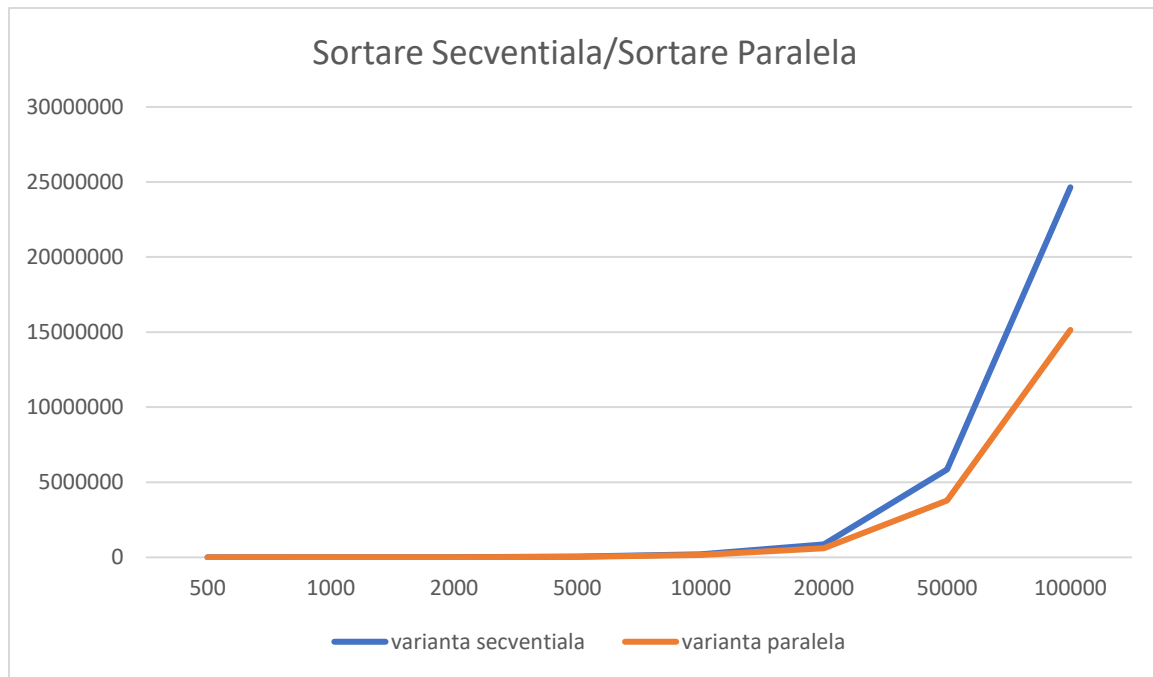
Au fost efectuate 10 teste de sortare cu urmatoarele esantioane de numere: 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000, 500000 de numere cu rezultatele obtinute prezentate in graficul de mai jos:



Varianta Paralela 1:

Pentru varianta paralela am folosit multiprocesarea cu ajutorul MPI, iar numarul de procese folosite pentru testarea esantioanelor de numere a fost 10.

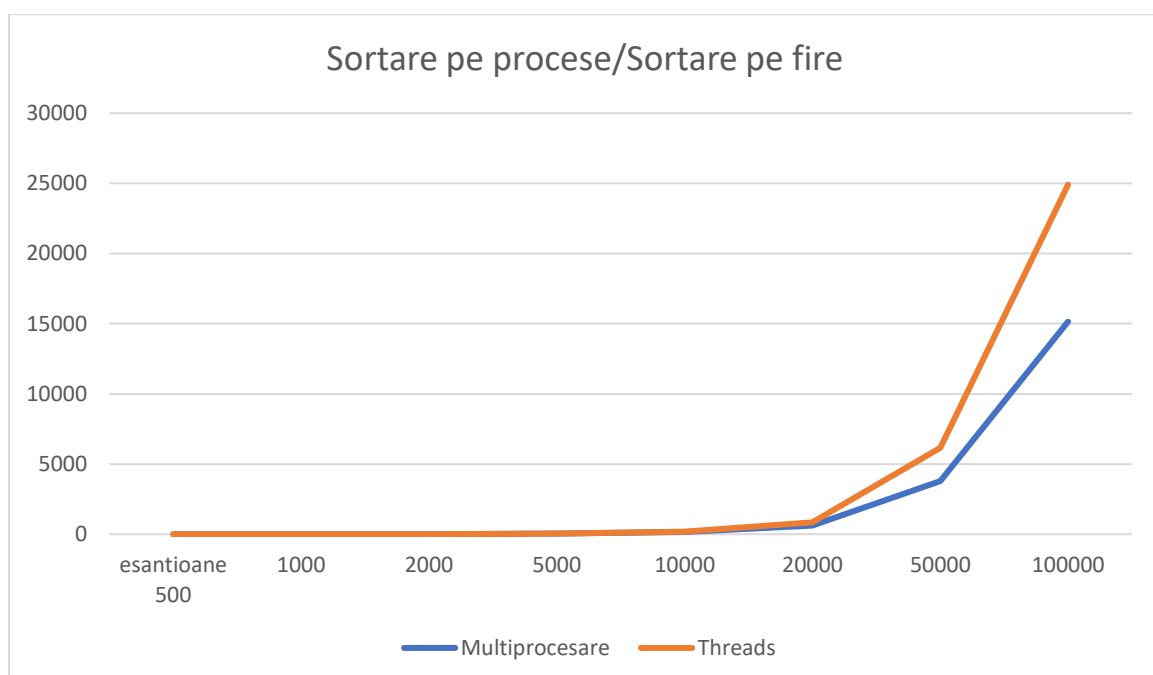
Mai jos este afisat un grafic cu rezultatele obtinute in varianta paralela comparativ cu cea secventiala:



Varianta Paralela 2:

A doua varianta secventiala a avut ca baza folosirea firelor de executie(Threads), acestea fiind in numar tot de 10 pentru testarea esantioanelor specificate mai sus si compararea cu sortarea bazata pe multiprocessing MPI.

Mai jos este o diagrama comparativa a celor doua sortari paralele(timpul este exprimat in milisecunde).



Concluzie:

Algoritmul Bubble Sort are un timp de rulare $O(n^2)$ lucru care poate fi observat si prin testele de mai sus unde timpul de rulare creste de aproximativ 4 ori la fiecare dublare a dimensiunii esantionului.

In ceea ce priveste varianta paralela aceasta devine mai eficienta din punct de vedere al timpului decat varianta secventiala dupa un esantion mai mare de 2000 de numere.