Laborator 1 (PPD)

Pater Beniamin

Implementati in Java

A) adunarea a doua matrice de dimensiune (nxm)

B) inmultirea a doua matrice de dimensiune (nxk), respectiv (k,m) folosind multithreading

-Numarul de threduri p trebuie sa fie un parametru care poate fi citit (modificat) inainte de inceperea executiei

Datele de intrare corespunzatoare elementelor matricilor se vor citi din fisiere (care au fost anterior create folosind generare aleatoare de numere)!

Programul va afisa la sfarsit timpul global(total) T de executie corespunzator operatiei de adunare(respective operatiei de inmultire) de matrice.

Sistem:

iMac (Retina 4K, 21.5-inch, Late 2015)

3,1 GHz Intel Core i5

16 GB 1867 MHz DDR3

Clase:

MatrixUtils, PairUtils, ThreadUtils, MultiplicationThread, SumThread

• Functia performSum() se ocupa de calcularea sumei a 2 matrice:

Aceasta foloste metoda getRandomArray() care scrie in fisier o matrice de foma row x column. Cu ajutorul acesteia se creeaza 2 matrici care vori fi scrise in fisier. firstMatrix.txt, secondMatrix.txt. Folosind readFile() se vor citi cele 2 matrici din fisiere.

Suma acestor 2 matrici se calculeaza folosind metoda addTwoMatrices(int[][] a, int[][]b, int thredsNo). Pt. a gestiona calulele pe threaduri folosim getPositionsList(List<Int> splits) cu parametrul reurnSplits(List<Int> portions, int number, int division).

reurnSplits(List<Int> portions, int number, int division) returneaza nr de operatii pt fieare thread

folosim getPositionsList(List<Int> splits) precizeaza indicia valorilor ce le va calcula fiecare thread.

Pt. calcularea pe threaduri a acestei operatii folosim clasaSumThread care extinde clasa Thread.

In functia run luam fiecare pereche de rand-coloana si calculam mcatricea suma cu instructiunea sumMatrix[row][column] = firstMatrix[row][column] + secondMatrix[row][column]

In caz de eroare se va afisa logul dat de linia de mai jos:

System.*out*.println("Row - " + row + " Column - " + column);

Testare:

Dimensiune

Threads

Timp

1000 x 1000

2

0.05308909e+10 nanosec

1000 x 1000

3

0.04926558e+10 nanosec

1000 x 1000

4

0.04926558e+10 nanosec

1000 x 1000 5 0.053470565e+10 nanosec

• Produsul a 2 matrici

Se apeleaza metoda multiplyTwoMatrices cu parametrii cele 2 matrice si cu nr de threaduri care vor executa operatiile. Apelam metoda getPairs pt. ambele matrici. Parcurgem in adancime ambele listesi pt. fiecare element calculam indicia pt matricea finala. Pt. fiecare element din lista precedent calculam lista de operatii ce vor fi effectuate. Folosind metoda returnSplits caluclam cate operatii va efectua fiecare thread. Vom creea un obiect MultiplicationThread care va primii ca parametrii lista de perechi, nr, de thread si matricea rezultat iar acesta va calcula produsul celor 2 matrici.

Testare:

Dimensiune

Threads

Timp

1000 x 1000

2

14.13752834

1000 x 1000

3

13.575948358

1000 x 1000

4

13.231245250

1000 x 1000 5 13.120248019