Laborator 1 - Nichita Utiu 233

Cerinta

Se vor implementa in java operatiile de adunare si de inmultire a doua matrici intr-un mod paralelizat folosind multithreading(pe un numar de thread-uri parametrizabil). Datele vor fi distribuite pe threaduri intr-un mod echilibrat(*balanced distribution*).

Se vor face benchmark-uri pentru testarea performantei operatiilor pe matrici aleatoare de dimensiuni mari(1000 x 1000). Pentru fiecare benchmark se vor specifica:

- dimensiunile matricei
- numarul de thread-uri
- timpul de executie
- specificatiile sistemului

Proiectare

Proiectul e impartit in urmatoarele clase:

- **MatrixAdder** - Clasa care contine metoda statica de adunare a doua matrici
  - `T[][] add(T[][] first, T[][] second, int numThreads, BinaryOperator<T,T,T>)` - metoda statica care primeste doua matrici si returneaza matricea rezultata aplicarii operatorului specificat pe fiecare perche corespunaztoare de elemente din cele 2 matrici. Operatia va rula pe numarul specificat de thread-uri.
- **MatrixMultiplier** - Clasa care contine metoda statica de inmultire a doua matrici
  - `T[][] multiply(T[][] first, T[][] second, int numThreads)` - metoda statica care primeste 2 matrici si returneaza rezultatul inmultirii matriciale. Operatia va rula pe numarul specificat de thread-uri.
- **Main** - clasa main care creaaza matricile random, cheama metodele din `MatrixAdder` si `MatrixMultiplier` pe ele si cronometreaza executia lor.

Performanta

Sistem: Antergos Linux(Arch Linux) 64bit - Intel® Core™ i7-5500U CPU @ 2.40GHz × 4

Adunare

dimensiune	# thread-uri	timp(ms)
1000 x 1000	1	177.813526
1000 x 1000	2	255.770882
1000 x 1000	4	103.896254
1000 x 1000	6	106.024314
1000 x 1000	8	143.861229

Inmultire

dimensiune	# thread-uri	timp(ms)
1000 x 1000	1	811.05972
1000 x 1000	2	445.144503
1000 x 1000	4	417.072139
1000 x 1000	6	401.540484
1000 x 1000	8	406.693372