Laborator 1 – Stefan Sebastian 235

Programare paralela si distribuita

Analiza cerintelor

Se cere sa se implementeze un program care poate efectua adunarea a doua matrici de dimensiuni(n x m) si inmultirea a doua matrici de dimensiuni(n x k) respectiv (k x m) folosind multithreading.

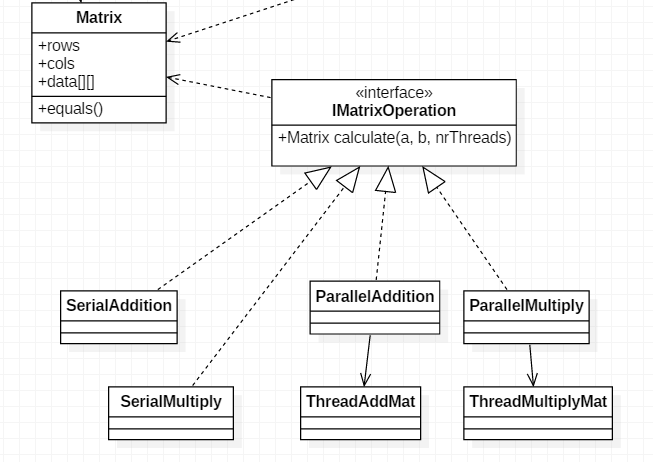
Programul va citi datele de intrare si va salva rezultatele in fisiere ale caror cai sunt configurabile. De asemenea, programul va putea genera aceste fisiere de intrare folosind parametrii configurabili de dimensiune si cale.

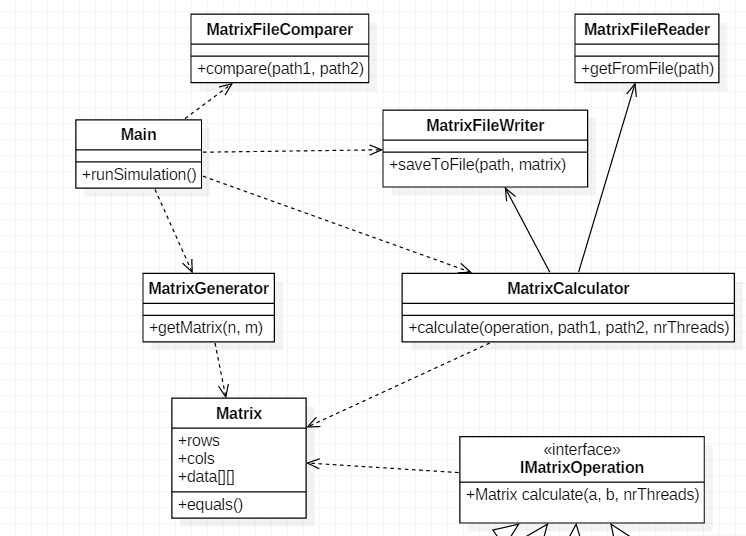
Constrangeri

Numarul de threaduri folosit trebuie sa fie un parametru care poate fi modificat inainte de inceperea executiei. Impartirea datelor pe threaduri trebuie sa urmeze o distributie echilibrata. Programul va afisa pe ecran timpul de executie al operatiilor.

Proiectare

In aceasta sectiune din diagram apar entitatile principale ale problemei: Matrix (wrapper peste un array bidimensional) si IMatrixOperation (interfata care defineste o operatie intre matrici) si implementarile ei.





In aceasta sectiune apar clasele care se ocupa de gestiunea fisierelor (MatrixFileReader, MatrixFileWriter, MatrixFileComparer), de generarea datelor(MatrixGenerator) si gruparea mai multor instructiuni de calcul(MatrixCalculator). Clasa Main este punctul de intrare in program iar metoda runSimulation() apeleaza toate metodele necesare pentru generare de date si efectuarea de calcule in functie de parametrii dati.

Detalii de implementare

Pentru simplitate notam: numar linii – nrL, numar coloane – nrC, numar threaduri – p.

Pentru a imparti cat mai echilibrat o matrice am luat in considerare urmatoarele aspecte. Elementele matricei pot fi numerotate incepand de la 0 pana la nrL \* nrC, incepand din coltul de stanga-sus si terminand in coltul din dreapta-jos. Pentru un numar n din acest interval putem determina atat linia cat si coloana coresponzatoare cu formulele: linia = n / nrC, coloana = n % nrC. Atat pentru adunarea cat si pentru inmultirea matricelor se face o cate o singura scriere pe o pozitie a matricei rezultat iar din cele initiale doar citiri.

Astfel, putem imparti matricea rezultat conform metodei de mai sus, in (nrL \* nrC) / p bucati care pot fi procesate in paralel. Sarcina clasei care implementeaza Runnable este de a parcurge o submatrice determinata de cele 2 colturi si de a aplica operatia corespunzatoare. Daca impartirea pe bucati nu este exacta, se adauga cate un element extra la primele (nrL \* nrC) % p.

Cazuri de testare

Testarea s-a facut pe un procesor intel i5-5200U, 2.2GHz, 2 cores, 4 threads.

Adunare

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date de intrare | Date de iesire adunare seriala | Date de iesire adunare paralela | Numar threaduri | Timp executie adunare seriala (ms) | Time executie adunare paralela (ms) |
| 3 3  48.06 1.16 44.22  45.49 91.03 1.53  76.61 52.23 52.09  3 3  59.05 9.34 78.51  45.64 58.29 47.79  67.73 0.55 58.51 | 3 3  107.11 10.5 122.73  91.13 149.32 49.32  144.34 52.77 110.6 | 3 3  107.11 10.5 122.73  91.13 149.32 49.32  144.34 52.77 110.6 | 3 | Aprox 0 | 4 |
| 2 3  90.42 99.73 51.69  26.61 49.76 17.87  2 3  43.82 90.97 10.61  22.26 87.68 86.63 | 2 3  134.24 190.7 62.3  48.87 137.44 104.5 | 2 3  134.24 190.7 62.3  48.87 137.44 104.5 | 2 | Aprox 0 | 1 |
| 2 3  11.6 30.28 55.41  28.85 92.25 86.29  3 3  36.21 3.57 28.44  41.19 68.47 32.57  67.69 52.79 83.28 | Invalid input | Invalid input | 2 | - | - |
| 1000 x 1000 | - | - | 10 | 11 | 9 |
| 1200 x 1200 | - | - | 4 | 20 | 8 |
| 800 x 800 | - | - | 10 | 9 | 7 |
| 3000 x 3000 | - | - | 4 | 75 | 38 |

Inmultire

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date de intrare | Date de iesire inmultire seriala | Date de iesire inmultire paralela | Numar threaduri | Timp executie inmult seriala (ms) | Time executie inmult paralela (ms) |
| 3 2  52.44 8.95  40.09 71.55  59.4 7.58  2 3  81.75 73.36 74.55  73.99 47.89 33.65 | 3 3  4949.1805 4275.6139 4210.5695  8571.342 6367.5319 5396.367  5416.7942 4720.5902 4683.3369999999995 | 3 3  4949.1805 4275.6139 4210.5695  8571.342 6367.5319 5396.367  5416.7942 4720.5902 4683.3369999999995 | 3 | Aprox 0 | 2 |
| 2 2  21.88 65.64  5.34 19.65  2 2  41.99 90.57  9.35 18.43 | 2 2  1532.4752 3191.4168  407.9541 845.7932 | 2 2  1532.4752 3191.4168  407.9541 845.7932 | 3 | Aprox 0 | 3 |
| 3 2  33.34 88.74  67.78 90.75  1.58 7.12  3 2  33.34 88.74  67.78 90.75  1.58 7.12 | Invalid input | Invalid input | 3 | - | - |
| 600 x 500  500 x 600 | - | - | 3 | 557 | 242 |
| 1000 x 1000  1000 x 1000 | - | - | 4 | 5589 | 4312 |
| 700 x 700  700 x 800 | - | - | 4 | 2551 | 1607 |
| 1200 x 1100  1100 x 1000 | - | - | 4 | 9379 | 5981 |
| 2000 x 2000  2000 x 2000 | - | - | 4 | 94779 | 54505 |