Project 1: Problem 2

소프트웨어학부 20204898 박소은

## **Environment**

* **Processor**: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz
* **Number of cores**: 4개
* **RAM**: 16.0GB(15.8GB available)
* **OS**: Windows 11 (64 bit)

## **Tables and graphs**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Thread #** | **1** | **2** | **4** | **6** | **8** | **10** | **12** | **14** | **16** | **32** |
| Exec Time(ms) | 667 | 490 | 323 | 302 | 268 | 257 | 257 | 254 | 234 | 223 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Thread #** | **1** | **2** | **4** | **6** | **8** | **10** | **12** | **14** | **16** | **32** |
| Performance(1/ms) | 0.0015 | 0.00204 | 0.0031 | 0.003311 | 0.003731 | 0.003891 | 0.003891 | 0.003937 | 0.004274 | 0.004484 |

## **Explanation / Analysis**

텍스트, 낱말맞추기 퍼즐이(가) 표시된 사진

자동 생성된 설명

## **Java source code**

package Prob2;

import java.util.\*;

import java.lang.\*;

/\*

< command-line execution >

Ex) java MatmultD 6 < mat500.txt

6 means the number of threads to use

< mat500.txt means the file that contains two matrices is given as standard input

\*/

public class MatmultD {

private static int NUM\_THREADS = 1;

private static Scanner sc = new Scanner(System.in);

public static void main(String[] args)

{

if (args.length == 1)

NUM\_THREADS = Integer.valueOf(args[0]);

int a[][] = readMatrix();

int b[][] = readMatrix();

MulMatrixThread[] threads = new MulMatrixThread[NUM\_THREADS];

ThreadController controller = new ThreadController(a.length, b[0].length);

long startTime = System.currentTimeMillis(); // program execution time starts

// Start threads

for (int i=0; i<NUM\_THREADS; i++) {

threads[i] = new MulMatrixThread(a, b, controller);

threads[i].start();

}

// Thread join()

try {

for (int i=0; i<NUM\_THREADS; i++) {

threads[i].join();

}

} catch (InterruptedException e) {}

long endTime = System.currentTimeMillis(); // program execution time ends

// Print Result

System.out.printf("[thread\_no]:%2d , [Total Execution Time]:%4d ms\n", NUM\_THREADS, endTime-startTime);

for (int i=0; i<NUM\_THREADS; i++) {

System.out.println(i+1 + " Thread: " + threads[i].diffTime + "ms");

}

}

public static int[][] readMatrix() {

int rows = sc.nextInt();

int cols = sc.nextInt();

int[][] result = new int[rows][cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = sc.nextInt();

}

}

return result;

}

}

class MulMatrixThread extends Thread {

long diffTime;

static int[][] a, b;

int m, p, n;

ThreadController controller;

MulMatrixThread(int a[][], int b[][], ThreadController controller) {

this.a = a;

this.b = b;

this.controller = controller;

m = a.length;

p = b[0].length;

n = a[0].length;

}

@Override

public void run() {

long startTime = System.currentTimeMillis();

int mIndex = controller.generateRowIndex(); // controller gives which row the thread has to calculate

while (mIndex < m) {

controller.writeResult(mIndex, multMatrix(mIndex, n, p));

mIndex = controller.generateRowIndex();

}

long endTime = System.currentTimeMillis();

diffTime = endTime - startTime;

}

public static int[] multMatrix(int mIndex, int n, int p) { // a[m][n], b[n][p]

/\*

thread calculate one row at one time

\*/

int ans[] = new int[p];

for (int i=0; i<p; i++) {

for (int j=0; j<n; j++) {

ans[i] += a[mIndex][j] \* b[j][i];

}

}

return ans;

}

}

class ThreadController {

// a[m][n], b[n][p] -> result[m][p]

public static int mIndex = 0;

public static int[][] result;

int m, p;

ThreadController(int m, int p) {

result = new int[m][p];

mIndex = -1;

this.m = m;

this.p = p;

}

public synchronized int generateRowIndex() {

mIndex += 1;

return mIndex;

}

public synchronized void writeResult(int index, int[] indexArray) {

/\*

Thread can write the result with this method.

indexArray: one row the thread calculated

the lenght of indexArray should be 'p'

\*/

for (int i=0; i<p; i++) {

result[index][i] = indexArray[i];

}

}

}

## **Screen capture image of program execution and output**

텍스트이(가) 표시된 사진

자동 생성된 설명

## **How to compile and execute the source code**

* $ java MatmultD.java 4 < mat500.txt