2022.1 Multicore Computing, Project #1

Problem 2

Document

소프트웨어학부

20176342 송민준

1. **Result**
2. **Execution environment**

CPU : AMD Ryzen 5 2600X Six-Core Processor (12 CPUs), ~3.6GHz

Memory : DDR4 16384MB RAM

OS : Windows 10

1. **Tables and graphs**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thread# | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 32 |
| Exec time | 271 | 157 | 116 | 111 | 105 | 104 | 125 | 107 | 120 | 174 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thread# | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 32 |
| Performance  (1/exec time) | 0.00369 | 0.00636 | 0.00862 | 0.00900 | 0.00952 | 0.00961 | 0.00800 | 0.00934 | 0.00833 | 0.00574 |

**(c) Explanation of results**

I implemented this matrix multiplication project by dividing 3-nested for iteration.

    for(int i = this.start;i < this.end;i++){

      for(int j = 0;j < p;j++){

        for(int k = 0;k < n;k++){

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

        }

      }

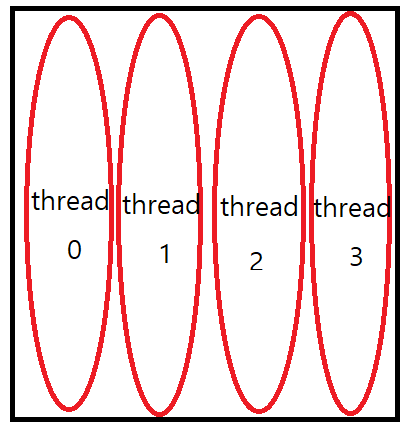
    }

This iteration is the basic of multiplication of two matrices.

At the outermost loop of for, i is divided by start and end point.

If matrix is 100x100 and If I have 4 threads, each threads has start and end value like

0~25, 26~50, 51~75, 76~100 respectively.

****

If the matrix above is the multiplication result of the two matrices, Different threads take charge of the results of each column of the matrix.

So more threads reduce execution time, but too many threads slow execution time.

The more threads, the higher the overhead cost of context switching between threads, the slower the execution time.

In this matrix multiplication example, 8 to 10 threads appear to be the most efficient.

I think the results will be similar even if dynamic load balancing is implemented in 'for iteration'. Because the calculation time for multiplication is similar no matter what, so the time required for each thread is already similar.

1. **entire JAVA source code and screen capture image of program execution and output**

MatmultD.java

import java.util.\*;

import java.lang.\*;

// command-line execution example) 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

//

// In eclipse, set the argument value and file input by using the menu [Run]->[Run Configurations]->{[Arguments], [Common->Input File]}.

// Original JAVA source code: http://stackoverflow.com/questions/21547462/how-to-multiply-2-dimensional-arrays-matrix-multiplication

public class MatmultD

{

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

  public static void main(String [] args)

  {

    int thread\_no=0;

    if (args.length==1) thread\_no = Integer.valueOf(args[0]);

    else thread\_no = 2;

    int a[][]=readMatrix();

    int b[][]=readMatrix();

    long startTime = System.currentTimeMillis();

    ThreadforMatrix.a = a;

    ThreadforMatrix.b = b;

    ThreadforMatrix.ans = new int[a.length][a.length];

    ArrayList<ThreadforMatrix> thread\_arr = new ArrayList<ThreadforMatrix>();

    for(int i = 0; i<thread\_no;i++){

      int start = i\*(a.length/thread\_no);

      int end = i == thread\_no-1 ? a.length : (i+1)\*(a.length/thread\_no);

      System.out.println("new thread range "+start+ " ~ "+end);

      ThreadforMatrix thread = new ThreadforMatrix(start,end, a.length);

      thread\_arr.add(thread);

      thread.start();

    }

    for(int i = 0;i<thread\_arr.size();i++){

      try {

        thread\_arr.get(i).join();

      } catch (InterruptedException e) {

        e.printStackTrace();

      }

    }

    long endTime = System.currentTimeMillis();

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

    printMatrix(ThreadforMatrix.ans);

  }

   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;

   }

  public static void printMatrix(int[][] mat) {

  System.out.println("Matrix["+mat.length+"]["+mat[0].length+"]");

    int rows = mat.length;

    int columns = mat[0].length;

    int sum = 0;

    for (int i = 0; i < rows; i++) {

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

        // System.out.printf("%4d " , mat[i][j]);

        sum+=mat[i][j];

      }

      // System.out.println();

    }

    // System.out.println();

    System.out.println("Matrix Sum = " + sum + "\n");

  }

}

class ThreadforMatrix extends Thread {

  static int a[][];

  static int b[][];

  static int ans[][];

  int start;

  int end;

  ThreadforMatrix(int start, int end, int size){

    this.start = start;

    this.end = end;

  }

  public void run(){

    long startTime = System.currentTimeMillis();

    int n = a[0].length;

    int m = a.length;

    int p = b[0].length;

    for(int i = this.start;i < this.end;i++){

      for(int j = 0;j < p;j++){

        for(int k = 0;k < n;k++){

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

        }

      }

    }

    long endTime = System.currentTimeMillis();

    long timeDiff = endTime - startTime;

    System.out.println(this.getName()+" Execution Time: "+timeDiff+"ms");

  }

}

**MatmultD.java**

MatmultD thread #1

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #2

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #4

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #6

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #8

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #10

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #12

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #14

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #16

텍스트이(가) 표시된 사진

자동 생성된 설명

MatmultD thread #32

텍스트이(가) 표시된 사진

자동 생성된 설명

**How to compile and execute**

텍스트이(가) 표시된 사진

자동 생성된 설명

Just use ‘javac’ to compile in my directory(problem2) and run ‘java MatmultD #num\_thread < matrix file name like above.