H) Write a program to implement matrix multiplication. Also implement multi-threaded matrix multiplication with either one thread per row or one thread per cell. Analyze and compare their performance. (Mini Project)

Aarti Ambekar

RollNo:20121056

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.util.concurrent.TimeUnit;

public class MultiThreadedMatrixMultiplication {

    public static int[][] multiplyMatrices(int[][] A, int[][] B) {

        int numRowsA = A.length;

        int numColsA = A[0].length;

        int numColsB = B[0].length;

        int[][] result = new int[numRowsA][numColsB];

        int numThreads = numRowsA;

        ExecutorService executor = Executors.newFixedThreadPool(numThreads);

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

            final int row = i;

            executor.submit(() -> {

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

                    for (int k = 0; k < numColsA; k++) {

                        result[row][j] += A[row][k] \* B[k][j];

                    }

                }

            });

        }

        executor.shutdown();

        try {

            executor.awaitTermination(1, TimeUnit.MINUTES);

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

        return result;

    }

    public static void main(String[] args) {

        int[][] matrixA = {

            {1, 2, 3},

            {4, 5, 6},

        };

        int[][] matrixB = {

            {7, 8},

            {9, 10},

            {11, 12},

        };

        int[][] result = multiplyMatrices(matrixA, matrixB);

        for (int i = 0; i < result.length; i++) {

            for (int j = 0; j < result[0].length; j++) {

                System.out.print(result[i][j] + " ");

            }

            System.out.println();

        }

    }

}

Ouput

