

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
Московский государственный технический университет имени Н.Э.Баумана  
(МГТУ им. Н.Э.Баумана)

## **ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**«Распараллеливание алгоритма вычисления произведения двух матриц»**

Выполнила: Митрошкин А.А.  
(Фамилия И.О. студента)  
ИУ9-51Б  
(Индекс группы)

Преподаватель: Царев.А. С.  
(Фамилия И.О. преподавателя)

(Подпись)

Москва, 2023

## **Оглавление**

### **1. Цель работы**

### **2. Условие задачи**

**3**

#### **2.1. Листинг кода программы**

**4**

## 1. Задача

Сравнить время работы вычисления матрицы на одном потоке и нескольких.

## 2. Условие задачи

Две квадратные матрицы A и B размерности  $n$  сначала перемножить стандартным алгоритмом для получения матрицы C той же размерности. Замерить время вычисления, сравнить с временем при вычислении элементов матрицы C не по строкам, а по столбцам. Размер матриц подобрать таким образом, чтобы время выполнения на вашей машине было не слишком непоказательно малым (меньше нескольких минут), но и не чересчур большим (несколько часов). Использовать библиотечные функции для вычисления произведений матриц нельзя. Затем конечную матрицу C условно разделить на примерно равные прямоугольные подматрицы и распараллелить программу таким образом, чтобы каждый поток занимался вычислением своей подматрицы. Матрицы A и B для этого разделить на примерно равные группы строк и столбцов соответственно. Сделать для разного количества потоков (разных разбиений), также замерить время вычисления, сравнить с вычислениями стандартным алгоритмом. Также по окончании вычислений сравнивать получившуюся матрицу с той, что была вычислена стандартным алгоритмом, для проверки правильности вычислений (проверка во время выполнения задачи не входит). Уделить внимание проблеме синхронизации потоков: основной поток, который создает потоки, занимающиеся вычислением подматриц, не должен закончиться до того, как они закончат свою работу, поэтому организовать ожидание и сигнализацию завершения рабочих потоков.

## 3. Листинг кода программы

```
4. using System;  
5. using System.Diagnostics;  
6. using System.Threading.Tasks;  
7.  
8. class MatrixMultiplication
```

```

9.  {
10.     static int[,] MultiplyMatrices(int[,] matrixA, int[,] matrixB)
11.     {
12.         int n = matrixA.GetLength(0); // Количество строк в матрице A
13.         int m = matrixB.GetLength(1); // Количество столбцов в матрице B
14.         int[,] result = new int[n, m]; // Создание результирующей матрицы C
           размерности n x m
15.
16.         Parallel.For(0, n, i =>
17.         {
18.
19.             for (int j = 0; j < m; j++)
20.             {
21.                 for (int k = 0; k < matrixA.GetLength(1); k++)
22.                 {
23.                     result[i, j] += matrixA[i, k] * matrixB[k, j];
24.                 }
25.             }
26.         });
27.
28.         return result;
29.     }
30.
31.     static int[,] Fill_Matrix(int[,] matrixA)
32.     {
33.         int n = matrixA.GetLength(0); // Количество строк в матрице A
34.         int m = matrixA.GetLength(1); // Количество строк в матрице A
35.
36.         int[,] result = new int[n, m];
37.
38.         for (int i = 0; i < n; i++)
39.         {
40.             for (int j = 0; j < m; j++)
41.             {
42.                 for (int k = 0; k < matrixA.GetLength(1); k++)
43.                 {
44.                     result[i, j] += new Random().Next(0, 10);
45.                 }
46.             }
47.         }
48.
49.         return result;
50.     }
51.
52.     static void Print(int[,] Matrix)
53.     {
54.         int n = Matrix.GetLength(0); // Количество строк в матрице A
55.         int m = Matrix.GetLength(1); // Количество строк в матрице A
56.
57.         for (int i = 0; i < n; i++)
58.         {
59.             for (int j = 0; j < m; j++)
60.             {
61.                 Console.Write(Matrix[i, j].ToString() + " ");
62.             }
63.             Console.WriteLine();
64.         }
65.     }
66.
67.     static int[,] MultiplyMatrices_non_parallel(int[,] matrixA, int[,] matrixB)
68.     {
69.         int n = matrixA.GetLength(0); // Количество строк в матрице A
70.         int m = matrixA.GetLength(1); // Количество столбцов в матрице A (и
           строк в матрице B)

```

```

71.
72.         int[,] result = new int[n, m]; // Результирующая матрица C размерности n
x p
73.
74.         if (m != matrixB.GetLength(0))
75.         {
76.             throw new ArgumentException("Несовместимые размерности матриц для
умножения.");
77.         }
78.
79.         for (int i = 0; i < n; i++)
80.         {
81.             for (int j = 0; j < m; j++)
82.             {
83.                 int sum = 0;
84.                 for (int k = 0; k < m; k++)
85.                 {
86.                     sum += matrixA[i, k] * matrixB[k, j];
87.                 }
88.                 result[i, j] = sum;
89.             }
90.         }
91.
92.         return result;
93.     }
94.
95.     static void Main()
96.     {
97.
98.         int n = 500; // Количество строк в матрице A
99.         int m = 500; // Количество столбцов в матрице B
100.        int[,] matrixA = new int[n, m]; // Создание матрицы A размерности
n x m
101.        int[,] matrixB = new int[m, n]; // Создание матрицы B размерности
m x n
102.
103.        matrixA = Fill_Matrix(matrixA);
104.        matrixB = Fill_Matrix(matrixB);
105.
106.        // Заполнение матриц A и B данными (ваш код)
107.        Console.WriteLine("start");
108.
109.        Stopwatch stopwatch = new Stopwatch();
110.        stopwatch.Start();
111.
112.        int[,] result = MultiplyMatrices(matrixA, matrixB);
113.
114.        stopwatch.Stop();
115.        Console.WriteLine($"Время выполнения параллельного алгоритма:
{stopwatch.ElapsedMilliseconds} мс");
116.
117.        stopwatch.Start();
118.
119.        int[,] result_non_parallel =
MultiplyMatrices_non_parallel(matrixA, matrixB);
120.        stopwatch.Stop();
121.        Console.WriteLine($"Время выполнения алгоритма:
{stopwatch.ElapsedMilliseconds} мс");
122.
123.    }
124. }

```

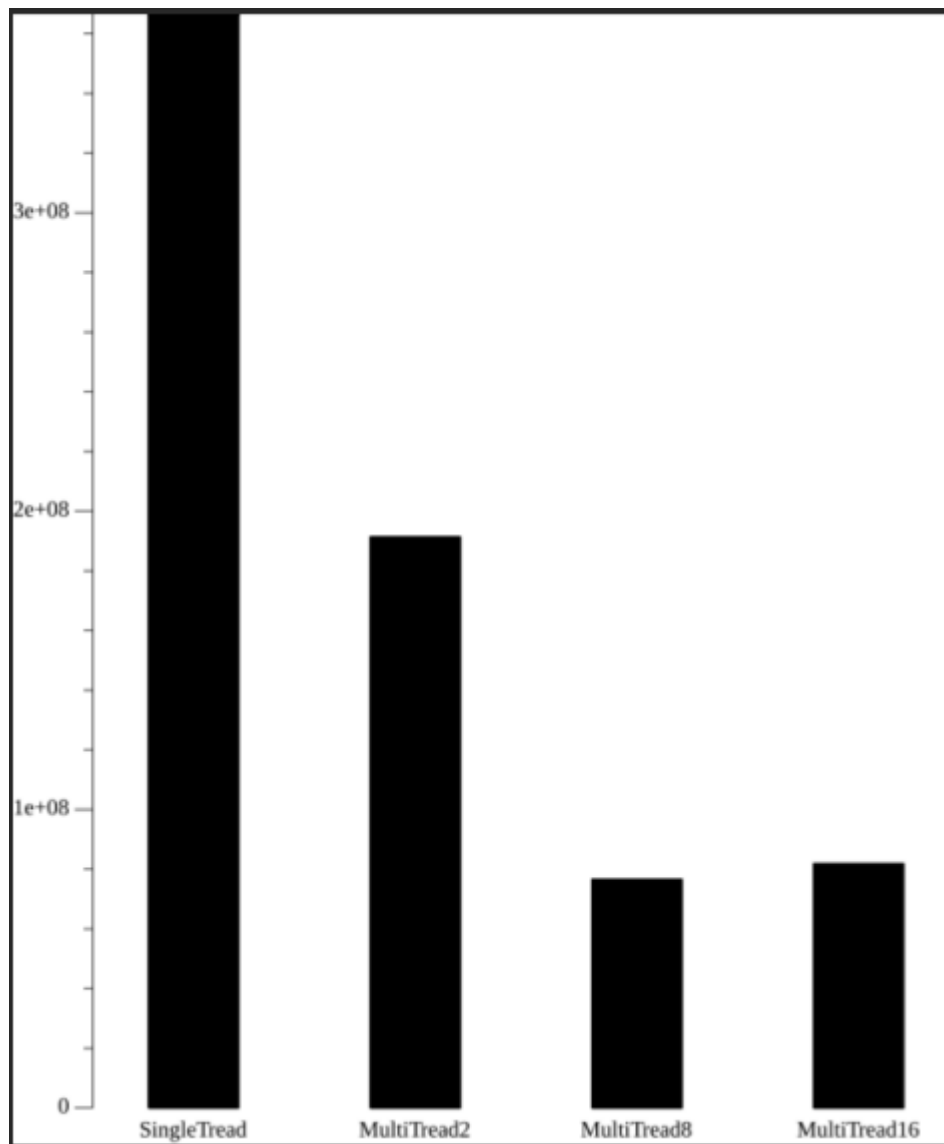
## Характеристики компьютера

**12th Gen Intel(R) Core(TM) i5-12450H 2.50 GHz**

**16,0 ГБ (доступно: 15,7 ГБ)**

**64-разрядная операционная система, процессор x64**

## 4. Результаты



Можно заметить, что результат улучшается с увеличением потоков, но остается примерно равным при размере от 8. Именно столько физических ядер есть на компьютере, которые позволяют считать матрицу параллельно