

**ГБОУ ВПО Нижегородский государственный технический  
университет им. Р. Е. Алексеева  
Институт радиоэлектроники и информационных технологий,  
кафедра "Вычислительные системы и технологии"**

**СОГЛАСОВАНО**

Доцент каф. ВСТ

\_\_\_\_\_ Гай В. Е.

“ \_\_\_\_ ” \_\_\_\_\_

**ТЕХНОЛОГИИ РАСПРЕДЕЛЁННОЙ ОБРАБОТКИ ДАННЫХ  
Отчет к лабораторной работе №3**

**РАСПАРАЛЛЕЛИВАНИЕ АЛГОРИТМА С ПОМОЩЬЮ  
БИБЛИОТЕКИ CSR**

Инв. подл.	Подп. и дата	Инв. дубл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	Инв. подл.

Студент гр. 13-В-1

\_\_\_\_\_ Смирнов А. В.

“ \_\_\_\_ ” \_\_\_\_\_

# СОДЕРЖАНИЕ

<b>1</b>	<b>Цель и порядок выполнения работы</b>	<b>3</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>4</b>
2.1	Библиотека Concurrent and Coordination Runtime . . . . .	4
2.2	Создание проекта . . . . .	5
2.3	Оценка времени выполнения . . . . .	5
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Вариант задания . . . . .	6
3.2	Листинг программы . . . . .	6
3.3	Результат работы программы . . . . .	12
<b>4</b>	<b>Вывод</b>	<b>13</b>

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	Распараллеливание алгоритма с помощью библиотеки CCR Технологии распределённой обработки данных Отчет к лабораторной работе №3	Лит.	Лист	Листов	
	Изм.	Лист	докум.	Подп.					Дата
	Разраб.	Смирнов А. В							
	Пров.	Гай В. Е.							
							2	13	
	Н. контр.								
	Утв.								

# 1 ЦЕЛЬ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Цель работы: получить представления о возможности библиотеки Concurrency and Coordination Runtime для организации параллельных вычислений.

Порядок выполнения работы:

- а) Разработка последовательного алгоритма, решающего одну из приведённых задач в соответствии с выданным вариантом задания;
- б) Разработка параллельного алгоритма, соответствующий варианту последовательного алгоритма;
- в) Выполнение сравнения времени выполнения последовательного и параллельного алгоритмов обработки данных при различных размерностях исходных данных.

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата					
Изм.	Лист	докум.	Подп.	Дата	Распараллеливание алгоритма с помощью библиотеки CCR		Лист		
							3		

## 2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### 2.1 Библиотека Concurrent and Coordination Runtime

Библиотека Concurrent and Coordination Runtime (CCR) предназначена для организации обработки данных с помощью параллельно и асинхронно выполняющихся методов. Взаимодействие между такими методами организуется на основе сообщений. Рассылка сообщений основана на использовании портов. Основные понятия CCR:

- а) Сообщение – экземпляр любого типа данных;
- б) Порт – очередь сообщений типа FIFO (First-In-First-Out), сообщение остаётся в порте пока не будет извлечено из очереди порта получателем. Определение порта:

```
Port<int> p = new Port<int>();
```

Отправка сообщения в порт:

```
p.Post(1);
```

- в) получатель – структура, которая выполняет обработку сообщений. Данная структура объединяет:

- один или несколько портов, в которые отправляются сообщения;
- метод (или методы), которые используются для обработки сообщений (такой метод называется задачей);
- логическое условие, определяющее ситуации, в которых активизируется тот или иной получатель.

Делегат, входящий в получатель, выполнится, когда в порт `intPort` придёт сообщение. Получатели сообщений бывают двух типов: временные и постоянные (в примере получатель – временный). Временный получатель, обработав сообщение (или несколько сообщений), удаляется из списка получателей сообщений данного порта.

Инов. подл.	Подп. и дата	Взам. инв.	Инов. дубл.	Подп. и дата	<div>Распараллеливание алгоритма с помощью библиотеки CCR</div>					Лист
										4
Изм.	Лист	докум.	Подп.	Дата						

г) процессом запуска задач управляет диспетчер. После выполнения условий активации задачи (одним из условий активации может быть получение портом сообщения) диспетчер назначает задаче поток из пула потоков, в котором она будет выполняться. Описание диспетчера с двумя потоками в пуле:

```
Dispatcher d = new Dispatcher(2, "MyPool");
```

Описание очереди диспетчера, в которую задачи ставятся на выполнение:

```
DispatcherQueue dq = new DispatcherQueue("MyQueue d);
```

## 2.2 Создание проекта

Нужно выполнить следующие действия:

- а) Установить библиотеку CCR (CCR входит в состав Microsoft Robotics Developer Studio);
- б) Создать проект консольного приложения и добавьте к проекту библиотеку Microsoft.Ccr.Core.dll.

## 2.3 Оценка времени выполнения

Время выполнения вычислений будем определять с помощью класса

Stopwatch :

```
Stopwatch sWatch = new Stopwatch();
```

```
sWatch.Start();
```

```
<выполняемый код>
```

```
sWatch.Stop();
```

```
Console.WriteLine(sWatch.ElapsedMilliseconds.ToString());
```

Изм.	Лист	докум.	Подп.	Дата	<i>Распараллеливание алгоритма с помощью библиотеки CCR</i>	Лист 5
Изм.	Лист	докум.	Подп.	Дата		



```

static int n;    //количество столбцов матрицы
static int nc;   //количество ядер

static void Test()
{
    nc = 2;
    ConsoleKey press;

    Console.WriteLine("Желаете задать размер матрицы самостоятельно? (Y/N)\n");

    press = Console.ReadKey(true).Key;
    if (press != ConsoleKey.N & press != ConsoleKey.Y)
    {
        Console.WriteLine("Некорректная клавиша");
        press = ConsoleKey.N;
    }

    if (press == ConsoleKey.Y)
    {
        Console.Write("Введите количество строк матрицы: ");
        m = Convert.ToInt32(Console.ReadLine());
        Console.Write("Введите количество столбцов матрицы: ");
        ;
        n = Convert.ToInt32(Console.ReadLine());
    }
    else if (press == ConsoleKey.N)
    {
        m = 15000;
        n = 15000;
        Console.WriteLine("\nРазмеры матрицы заданы автоматически и составляют {0} x {1}\n",m,n);
    }

    A = new int[m, n];
    B = new int[n];

```

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата	Распараллеливание алгоритма с помощью библиотеки CCR					Лист
										7
Изм.	Лист	докум.	Подп.	Дата						

```
C = new int[m];
```

```
Console.WriteLine("\n\nЖелаете заполнить матрицу и вектор–  
столбец самостоятельно? (Y/N)\n");  
press = Console.ReadKey(true).Key;
```

```
if (press != ConsoleKey.N & press != ConsoleKey.Y)  
{  
    Console.WriteLine("Некорректная клавиша");  
    press = ConsoleKey.N;  
}
```

```
if (press == ConsoleKey.Y)  
{  
    Console.WriteLine("Ввод матрицы\n");  
    for (int i = 0; i < m; i++)  
    {  
        for (int j = 0; j < n; j++)  
        {  
            Console.Write("A[{0},{1}] = ", i+1, j+1);  
            A[i, j] = Convert.ToInt32(Console.ReadLine());  
        }  
    }
```

```
Console.WriteLine("\n\nВвод вектор–столбца\n");  
for (int j = 0; j < n; j++)  
{  
    Console.Write("B[{0}] = ", j+1);  
    B[j] = Convert.ToInt32(Console.ReadLine());  
}
```

```
else if (press == ConsoleKey.N)  
{  
    Console.WriteLine("Заполнение случайными значениями  
... \n");  
    Random r = new Random();  
    for (int i = 0; i < m; i++)  
    {  
        for (int j = 0; j < n; j++)
```

Изм.	Лист	докум.	Подп.	Дата	Распараллеливание алгоритма с помощью библиотеки CCR	Лист
						8



```

        A[i, j] = r.Next(100);
    }
    for (int j = 0; j < n; j++)
        B[j] = r.Next(100);

    Console.WriteLine("Исходная матрица и вектор-столбец у
        спешно заполнены случайными значениями!\n");
}

static void SequentialMul()
{
    System.Diagnostics.Stopwatch sWatch = new System.
        Diagnostics.Stopwatch();
    sWatch.Start();
    for (int i = 0; i < m; i++)
    {
        C[i] = 0;
        //Console.WriteLine("\n"); //эта строка тоже относится
        //к блоку проверки правильности
        for (int j = 0; j < n; j++)
        {
            C[i] += A[i, j] * B[j];
            //Внимание!
            //ниже закомментирован блок отображения проверки пр
            //авильности умножения
            //Если включить данный блок в исполняемый код,
            //то для корректного вывода проверки
            //рекомендуется брать матрицы небольшого размера
            //например, 3x3
            //
            //Console.Write(" {0} * {1} ", A[i, j], B[j]);
            //if (j+1 != n)
            //{
            //    Console.Write("+");
            //}
            //else
            //{
            //    Console.Write("= {0}", C[i]);
            //}
        }
    }
}

```

Интв. подл.	Подп. и дата	Интв. дубл.	Подп. и дата	Взам. инв.	Интв. дубл.	Подп. и дата	Интв. подл.
Изм	Лист	докум.	Подп.	Дата	Распараллеливание алгоритма с помощью библиотеки CCR		Лист
							9

Инв. подл.	Подп. и дата	Взам. инв.	Инв. дубл.	Подп. и дата

```

//Первый параметр метода Arbiter.Activate очередь дисп
етчера,
//который будет управлять выполнением задачи, второй парам
етр
//запускаемая задача.

//С помощью метода Arbiter.MultipleItemReceive запускается
задача
//(приёмник), которая обрабатывает получение двух сообщени
й портом p:
Arbiter.Activate(dq, Arbiter.MultipleItemReceive(true, p,
nc, delegate (int[] array)
{
    dispResult();
    Console.WriteLine("Вычисления завершены");
    Console.ReadKey(true);
    Environment.Exit(0);
}));
}

```

```

static void Mul(InputData data, Port<int> resp)
{
    System.Diagnostics.Stopwatch sWatch = new System.
        Diagnostics.Stopwatch();
    sWatch.Start();

    for (int i = data.start; i < data.stop; i++)
    {
        C[i] = 0;
        for (int j = 0; j < n; j++)
            C[i] += A[i, j] * B[j];
    }
    sWatch.Stop();
    Console.WriteLine("Поток {0}: Паралл. алгоритм = {1} м
        с.",
        Thread.CurrentThread.ManagedThreadId,
        sWatch.ElapsedMilliseconds.ToString());
    resp.Post(1);
}

```

Инов. подл.	Подп. и дата	Инов. дубл.	Подп. и дата	Взам. инв.	Подп. и дата	Инов. подл.	Изм.	Лист	докум.	Подп.	Дата	Распараллеливание алгоритма с помощью библиотеки CCR		Лист
														11



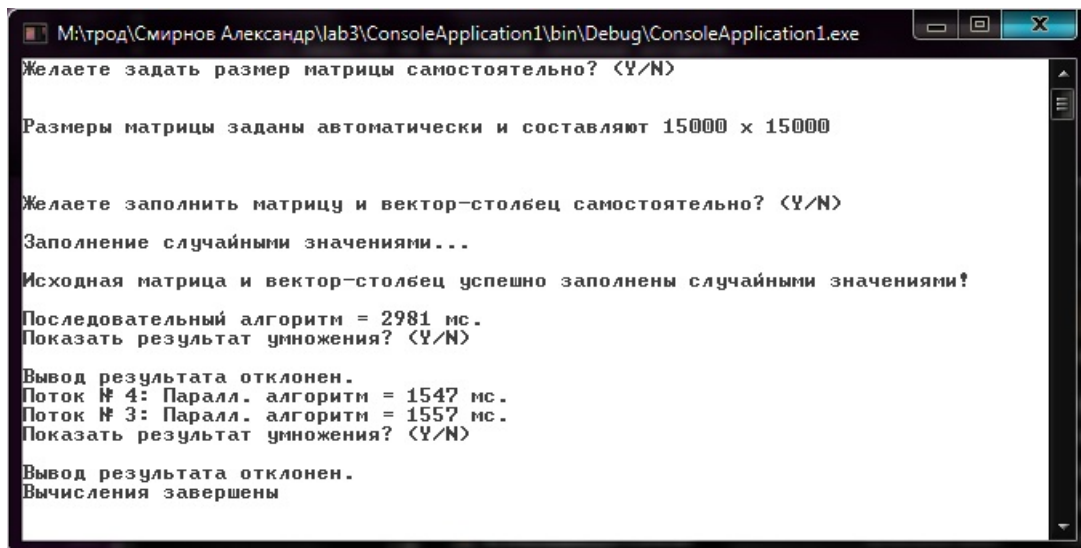


Рисунок 2

## 4 ВЫВОД

В результате выполнения лабораторной работы мы получили представление о возможности библиотеки Concurrent and Coordination Runtime для организации параллельных вычислений. Мы выяснили, что скорость работы параллельного алгоритма превосходит скорость работы последовательного алгоритма. Быстродействие параллельного алгоритма напрямую зависит от числа используемых ядер.

Инов. подл.	Подп. и дата	Взам. инв.	Инов. дубл.	Подп. и дата
<div> <div>Изм</div> <div>Лист</div> <div>докум.</div> <div>Подп.</div> <div>Дата</div> </div> <div> <div>Распараллеливание алгоритма с помощью библиотеки CCR</div> <div>Лист</div> <div>13</div> </div>				
<div>Копировал</div> <div>Формат А4</div>				