

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»
ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ**

**ОТЧЁТ
по лабораторной работе №1
по дисциплине «Параллельные алгоритмы»
Тема: Основы работы с процессами и потоками**

Студент гр. 9303

Эйсвальд М.И.

Преподаватель

Сергеева Е.И.

Санкт-Петербург
2022

Цель работы.

Научиться работать с процессами и потоками, изучить базовые механизмы взаимодействия процессов и потоков.

Задача.

Выполнить поэлементное сложение 2х матриц $M \times N$. Входные матрицы вводятся из файла (или генерируются). Результат записывается в файл.

1.1 Выполнить задачу, разбив её на 3 процесса. Выбрать механизм обмена данными между процессами. Процесс 1: заполняет данными входные матрицы (читает из файла или генерирует их некоторым образом).

Опционально: в этом процессе могут быть 2 потока ввода/генерации данных

Процесс 2: выполняет сложение.

Процесс 3: выводит результат.

1.2.1 Аналогично 1.1, используя потоки (threads).

1.2.2 Разбить сложение на P потоков. Исследовать зависимость между количеством потоков, размерами входных данных и параметрами целевой вычислительной системы.

Выполнение работы.

Матрица представляется в программах классом `Matrix`, который содержит реализацию различных операций с матрицей, которые производятся в рамках работы: генерация матрицы, создание пустой матрицы, чтение и запись элемента по индексу строки и индексу столбца, сериализация в массив чисел и десериализация и т. п.

В задаче сложения матриц с помощью нескольких процессов межпроцессное взаимодействие происходит через `pipe`: сначала ответившийся от

основного процесса процесс-генератор создаёт две матрицы, сериализует их и посылает через `pipe`, открытый перед ответвлением дочернего процесса, родительскому процессу. Родительский процесс складывает матрицы и аналогичным образом посылает результат печатающему процессу через `pipe`. Демонстрация работы программы представлена на рисунке 1.

```
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/Labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./processes 4 2 20
Generating and adding 4x2 matrices with elements up to 20
/ 17 8 \
| -9 -5 |
| -8 16 |
\ 16 18 /

/ -15 -5 \
| -3 -6 |
| 7 12 |
\ 18 11 /
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/Labs_PA_22/9303_Eiswald_Mikhail/lb1$ cat output.txt
/ 2 3 \
| -12 -11 |
| -1 28 |
\ 34 29 /
```

Рисунок 1 – Демонстрация работы трёх процессов для сложения матриц

Поскольку задание сложения матриц с помощью P потоков является частным случаем сложения матриц с помощью двух потоков, было выполнено сразу только последнее задание. Код генерации матриц и сложения элементов в определённом диапазоне был вынесен в отдельные функции `generate_matrices` и `sum_elements` соответственно, код вывода результата в файл был вынесен в лямбда-функцию. Каждая из функций вызывается в отдельном потоке (или нескольких потоках). Диапазоны индексов, в которых поток с номером i должен складывать элементы матрицы, вычисляются в родительском потоке, а в складывающие потоки явно передаётся их номер и ссылки на вектора, где хранятся начала и концы диапазонов.

В рамках исследования зависимости затраченного на сложение времени от размера входных данных и количества потоков было выявлено следующее:

- С увеличением количества потоков время обработки одинакового объёма данных уменьшается приблизительно кратно количеству потоков; в какой-то момент эффект от добавления дополнительных потоков сходит на нет (рисунок 2);

```

michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 1
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 515 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 2
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 2 threads
Elapsed time: 297 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 4
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 4 threads
Elapsed time: 158 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 8
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 8 threads
Elapsed time: 135 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 16
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 16 threads
Elapsed time: 141 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 100
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 135 ms
Done. Check output.txt for results.

```

Рисунок 2 – Обработка данных разным количеством потоков

- На очень малых данных от увеличения числа потоков наблюдается обратный эффект: с увеличением числа потоков увеличивается время обработки данных (рисунок 3);

```

michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 3 2 20 1000
Generating and adding 3x2 matrices with elements up to 20 using 1000 threads
/ 19 20 \
| -9 -9 |
\ 8 -9 /
Elapsed time: 40 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 3 2 20 1
Generating and adding 3x2 matrices with elements up to 20 using 1 threads
/ 13 -8 \
| 6 5 |
\ -10 8 /
Elapsed time: 0 ms
Done. Check output.txt for results.

```

Рисунок 3 – Обработка малого объёма данных разным количеством потоков

- При одинаковом числе потоков время обработки увеличивается линейно относительно количества элементов в матрице, но при большом числе потоков увеличивается медленнее (рисунки 4–6).

Результаты можно объяснить следующим:

- Эффект от распараллеливания операций есть до тех пор, пока есть вычислительные ресурсы, способные параллельно выполнять операции. В случае домашних компьютеров параллельно выполняют вычисления

```

michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 100 100 1000000000 1
Generating and adding 100x100 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 0 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 500 500 1000000000 1
Generating and adding 500x500 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 14 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 1000 1000 1000000000 1
Generating and adding 1000x1000 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 44 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 1500 1500 1000000000 1
Generating and adding 1500x1500 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 75 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 2000 2000 1000000000 1
Generating and adding 2000x2000 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 135 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 2500 2500 1000000000 1
Generating and adding 2500x2500 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 196 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 3000 3000 1000000000 1
Generating and adding 3000x3000 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 297 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 4000 4000 1000000000 1
Generating and adding 4000x4000 matrices with elements up to 1000000000 using 1 threads
Elapsed time: 535 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$

```

Рисунок 4 – Обработка разных объёмов данных одним потоком

```

michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 100 100 1000000000 100
Generating and adding 100x100 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 5 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 500 500 1000000000 100
Generating and adding 500x500 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 7 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 1000 1000 1000000000 10
0
Generating and adding 1000x1000 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 13 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 1500 1500 1000000000 10
0
Generating and adding 1500x1500 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 24 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 2000 2000 1000000000 10
0
Generating and adding 2000x2000 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 40 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 2500 2500 1000000000 10
0
Generating and adding 2500x2500 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 60 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$ ./threads 3000 3000 1000000000 10
0
Generating and adding 3000x3000 matrices with elements up to 1000000000 using 100 threads
Elapsed time: 87 ms
Done. Check output.txt for results.
michael@michael-TUF-Gaming-FX505DT-FX505DT:~/Desktop/work/PA/labs_PA_22/9303_Eiswald_Mikhail/lb1$

```

Рисунок 5 – Обработка разных объёмов данных несколькими потоками

ядра процессора, поэтому приращение скорости от нового потока есть тогда, когда потоков не больше, чем ядер;

- Распараллеливание вычислений на количество потоков, большее, чем количество ядер, вносит накладные расходы из-за смены контекста. Если объём вычислений маленький, то накладные расходы могут переве-

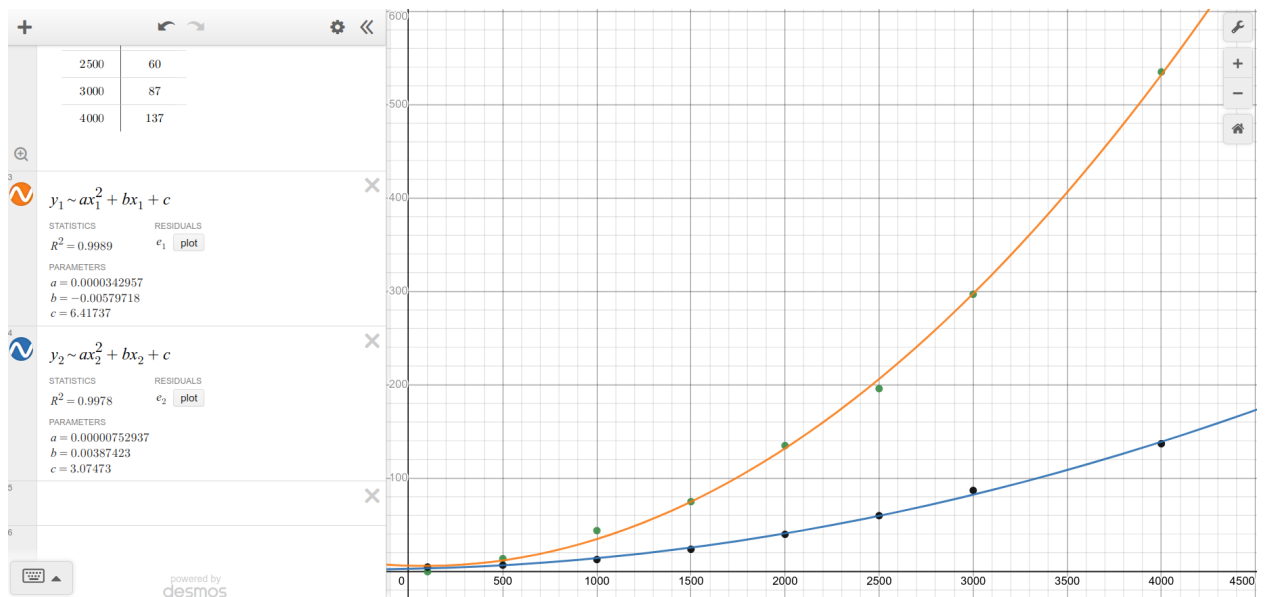


Рисунок 6 – Зависимость времени обработки данных от размерности квадратной матрицы

шивать пользу от параллелизма;

- Сложение матриц имеет сложность $O(mn)$, где m и n — размеры матриц (или $O(n^2)$ для квадратных матриц). Поскольку график зависимости построен с использованием размерности квадратной матрицы как аргумента, зависимость является квадратичной.

Вывод.

В ходе выполнения работы были изучены принципы работы с процессами и потоками, был исследован и оценён эффект от внедрения параллелизма в зависимости от объёма задачи и количества потоков. Из результатов исследования следует, что оптимальное количество потоков для минимизации времени вычислений равно количеству вычислительных устройств в системе (ядер процессора).