

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

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Параллельные алгоритмы»**  
**ТЕМА: ПАРАЛЛЕЛЬНОЕ УМНОЖЕНИЕ МАТРИЦ**

Студент гр. 9304

Афанасьев А.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

### **Цель работы.**

Реализовать параллельный алгоритм умножения матриц. Исследовать масштабируемость выполненной реализации. Реализовать параллельный алгоритм “быстрого” умножения матриц Штрассена.

### **Задание.**

Реализовать параллельный алгоритм умножения матриц. Исследовать масштабируемость выполненной реализации.

Реализовать параллельный алгоритм “быстрого” умножения матриц Штрассена.

Проверить, что результаты вычислений разных реализаций совпадают.

Сравнить производительность на больших размерностях данных (порядка  $10^4 - 10^6$ ).

### **Выполнение работы.**

Первый алгоритм рассматривает матрицу как массив и заключается в разбиении данных на независимые части. Поскольку части независимы друг от друга, то и нет необходимости синхронизации самого алгоритма,

Алгоритм Штрассена является рекурсивным. Для распараллеливания разбиваем данные на минимальные независимые части, чтобы уменьшить конкуренцию между потоками и увеличить пропускную способность. Кратко алгоритм описывается так:

1. Разбить матрицу на 7 частей.
2. Рекурсивно для каждой из 7 частей выполнить п.1, п.2, а затем объединить в единый результат.

Пример работы программы представлен на рисунке 1.

```
lab4 x
/home/afanasev-a@ad.speechpro.com/repos/labs_PA_22/9304_Afanasyev_A/lab4/cmake-build-debug/lab4
Enter matrix size
1024
Enter strassen depth
16
Pid: 855295 ThreadId: 140330803394368 Strassen duration: 2462911
Enter thread count
8
Pid: 855295 ThreadId: 140330803394368 Parallel duration: 3460806

Process finished with exit code 0
|
```

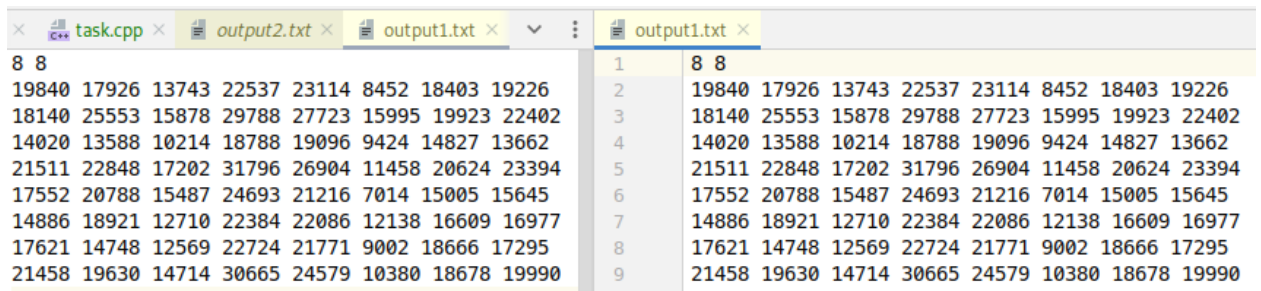
Рисунок 1 - Пример работы

Далее приведено сравнение алгоритмов при разных параметров.

Таблица 1: Результаты измерений

Название	Параметры	Размер матрицы	Время, мк. с.
Первый	1 поток	8 на 8	40
Первый	8 потоков	8 на 8	147
Штрассен	Глубина 1	8 на 8	176
Первый	1 поток	64 на 64	2562
Первый	8 потоков	64 на 64	956
Первый	32 потока	64 на 64	1053
Штрассен	Глубина 1	64 на 64	937
Штрассен	Глубина 2	64 на 64	874
Штрассен	Глубина 4	64 на 64	884
Штрассен	Глубина 16	64 на 64	959
Первый	8 потоков	2048 на 2048	28814276
Штрассен	Глубина 1	2048 на 2048	15809930
Штрассен	Глубина 2	2048 на 2048	15881261
Штрассен	Глубина 3	2048 на 2048	16086299
Штрассен	Глубина 4	2048 на 2048	17033200
Штрассен	Глубина 16	2048 на 2048	17137252

На рисунке 2 представлено подтверждение идентичности результатов умножения обоих алгоритмов.



task.cpp	output2.txt	output1.txt	output1.txt
8 8	1	8 8	
19840 17926 13743 22537 23114 8452 18403 19226	2	19840 17926 13743 22537 23114 8452 18403 19226	
18140 25553 15878 29788 27723 15995 19923 22402	3	18140 25553 15878 29788 27723 15995 19923 22402	
14020 13588 10214 18788 19096 9424 14827 13662	4	14020 13588 10214 18788 19096 9424 14827 13662	
21511 22848 17202 31796 26904 11458 20624 23394	5	21511 22848 17202 31796 26904 11458 20624 23394	
17552 20788 15487 24693 21216 7014 15005 15645	6	17552 20788 15487 24693 21216 7014 15005 15645	
14886 18921 12710 22384 22086 12138 16609 16977	7	14886 18921 12710 22384 22086 12138 16609 16977	
17621 14748 12569 22724 21771 9002 18666 17295	8	17621 14748 12569 22724 21771 9002 18666 17295	
21458 19630 14714 30665 24579 10380 18678 19990	9	21458 19630 14714 30665 24579 10380 18678 19990	

Рисунок 2 - Сравнение результатов алгоритмов

## Выводы.

На языке программирования C++ были реализованы параллельные алгоритмы умножения матриц. В результате сравнения было установлено, что первое разработанное решение успешно масштабируется при увеличении размеров матрицы и потоков, а также предложенная модификация алгоритма Штрассена в среднем быстрее первого алгоритма подхода. Масштабируется алгоритм Штрассена на тестовом окружении крайне плохо по причине недостаточного количества вычислительных ядер (4 физических ядер, 8 логических).