

**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**

Студентка гр. 0381

Ионина К.С.

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.**

4.1 Реализовать параллельный алгоритм умножения матриц. Исследовать масштабируемость выполненной реализации.

4.2 Реализовать параллельный алгоритм “быстрого” умножения матриц (Штрассена или его модификации). Проверить, что результаты вычислений реализаций 4.1 и 4.2 совпадают.

Сравнить производительность с реализацией 4.1 на больших размерностях данных (порядка  $10^4 - 10^6$ )

### **Выполнение работы.**

Было реализовано параллельное умножение матриц.

Для реализации алгоритма Штрассена была написана функция, которая:

- 1) Если размер матрицы меньше 128, то происходит обычное умножение матриц. (Выбрано 128, т.к. после такой размерности алгоритм Штрассена не эффективен)
- 2) Если глубина рекурсии выше, чем задал пользователь при запуске программы, то функция вызывается рекурсивно без создания нового потока.
- 3) Если предыдущие условия не выполняются, то функция вызывается рекурсивно в новом потоке.

Внутренняя функция реализует алгоритм Штрассена. Схема представлена на рис.1

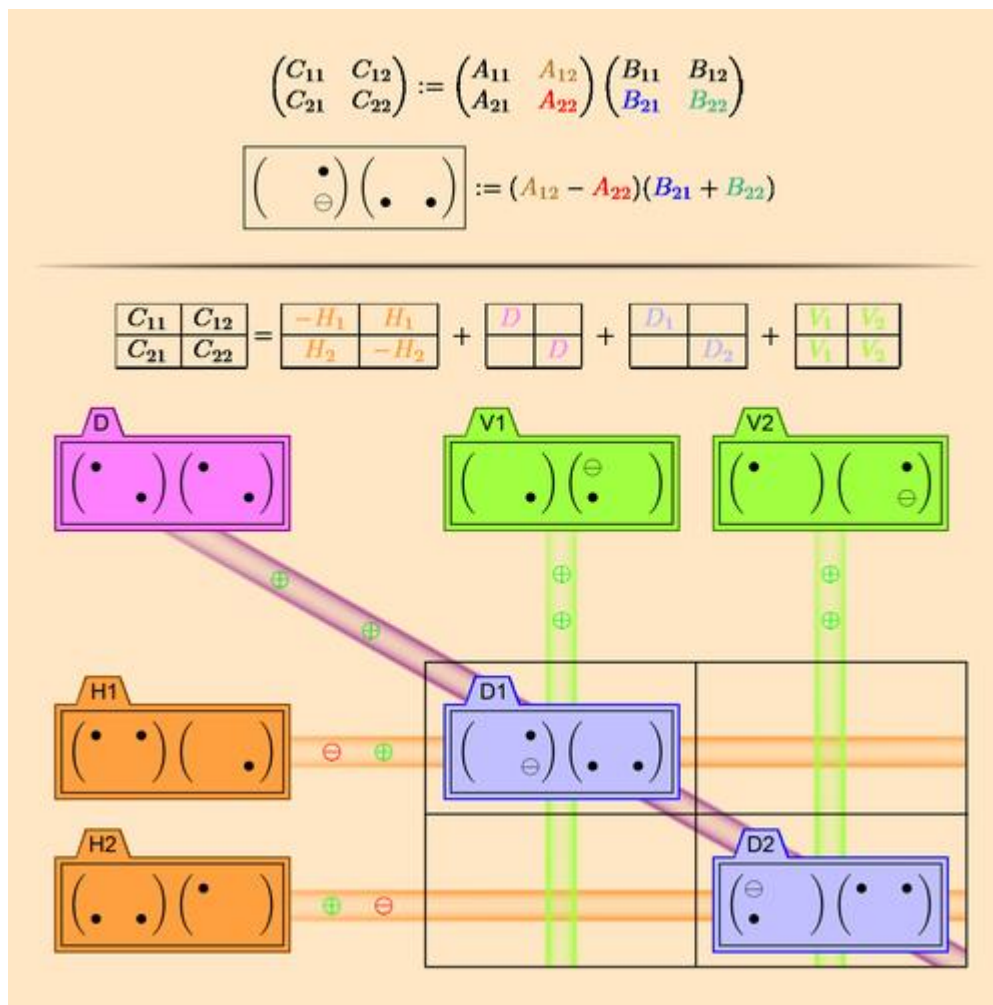


Рисунок 1 – алгоритм Штрассена

На рис.2 и рис3. показано, что результаты вычислений реализаций 4.1 и 4.2 совпадают.

```

≡ strassen_mult.txt
1  4 4
2  9293 4679 8734 4272
3  5432 2296 5412 1748
4  5340 3820 4176 2024
5  13439 6929 10886 4756

```

Рисунок 2 – Демонстрация работы программы

```

≡ parallel_mult.txt
1  4 4
2  9293 4679 8734 4272
3  5432 2296 5412 1748
4  5340 3820 4176 2024
5  13439 6929 10886 4756

```

Рисунок 3 – Демонстрация работы программы

В таблице 1 представлено сравнение времени работы алгоритма в зависимости от размера входных данных. Количество потоков для параллельного умножения – 7. Максимальная глубина рекурсии для алгоритма Штрассена – 1.

Размер данных	Время работы алгоритма умножения	Время работы алгоритма Штрассена
32*32	557ms	582ms
256*256	84688ms	142050ms
2024*2024	728934859ms	300864813ms

Заметим, что с увеличением размера матрицы алгоритм Штрассена работает быстрее.

### **Выводы.**

В ходе выполнения лабораторной работы была написана программа на языке программирования C++ для параллельного умножения матриц. Также был реализован алгоритм Штрассена.