

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

Боблаков Д.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Реализовать параллельный алгоритм и алгоритм «быстрого» умножения матриц.

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.

Matrix представляет из себя двойной вектор для хранения данных.

Для параллельного алгоритма умножения матриц была реализована функция `parallelMultiply`, которая принимает две матрицы `Matrix` и количество потоков. Каждому потоку достается определенная область, в которой необходимо провести умножение и отобразить результат на результирующую матрицу.

Алгоритм «быстрого» умножения представлен функцией `strassenMultiply`, которому на вход так же подается две матрицы и количество потоков. Для реализации алгоритма Штрассена были реализованы функции, которые могут работать с подматрицами – четвертей оригинальной матрицы. Также была реализована рекурсивная функция, задача которой производить разбиение и подсчет до тех пор, пока подматрицы не станут размером 64, ибо дальнейшее разбиение не целесообразно.

Сравнение производительности параллельного и «быстрого» алгоритмов.

В таблице 1 представлено время выполнения для каждого алгоритма, при разных размерах матрицы:

Таблица 1 – Зависимость времени выполнения от размера матриц

Размер матрицы	Алгоритм	Время, с
8x8	Тривиальный, 1 поток	0.000367
8x8	Тривиальный, 4 потока	0.000786
8x8	Штрассена, глубина 1	0.00031
64x64	Тривиальный, 1 поток	0.013798
64x64	Тривиальный, 4 потока	0.009913
64x64	Штрассена, глубина 1	0.019034
64x64	Штрассена, глубина 2	0.009422
128x128	Тривиальный, 1 поток	0.062142
128x128	Тривиальный, 4 потока	0.057144
128x128	Штрассена, глубина 1	0.031796
128x128	Штрассена, глубина 2	0.030638
2048x2048	Тривиальный, 1 поток	365.002
2048x2048	Тривиальный, 4 потока	162.189
2048x2048	Штрассена, глубина 1	86.5452
2048x2048	Штрассена, глубина 2	83.9603
2048x2048	Штрассена, глубина 4	83.1651
2048x2048	Штрассена, глубина 16	89.3605

Выводы.

В ходе выполнения лабораторной работы была реализована программа на языке программирования C++ для умножения матриц. Было произведено сравнение и установлено, что алгоритм Штрассена показывает лучший результат на больших данных по сравнению с обычным параллельным алгоритмом.