

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. 0304

Алексеев Р.В.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2023

Цель работы.

Изучить параллельное умножение матриц с масштабируемым количеством потоков и алгоритм Штрассена для быстрого умножения матриц.

Задание.

4.1 Реализовать параллельный алгоритм умножения матриц с масштабируемым разбиением по потокам.

Исследовать масштабируемость выполненной реализации с реализацией из работы 1.

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.

1. Для реализации параллельного алгоритма был взят код первой лабораторной работы без изменений.

В данной реализации представлен ряд функций, которые создают матрицы, умножают матрицы, разделяют умножение по потокам, выводят результаты в стандартный поток вывода.

2. Для реализации алгоритма Штрассена создан класс `Matrix`, который содержит поля `n` и `mat`, размерность матрицы и элементы матрицы соответственно. Сам алгоритм реализован в функциях `strassen()` и `multiStrassen()`.

Реализованный алгоритм работает только с матрицами, размер которых является степенью двойки. В противном случае матрицы дополняются строками и столбцами нулей. При выводе нулевые строки и столбцы удаляются.

Для умножения матрицы рекурсивно делятся на 4 подматрицы на каждом шаге, которые складываются и вычитаются между собой. После вычислений матрицы соединяются обратно. Рекурсивное деление происходит пока не будет достигнут размер матрицы — 1×1 .

3. Алгоритм параллельного умножения взят из 1 лабораторной работы, код умножения аналогичен. Структура `matrixes` заменена на класс `Matrix`, аналогичный классу в реализации алгоритма Штрассена.

Были проведены замеры зависимости времени от размеров складываемых матриц, результаты представлены в таб. 1.

Таблица 1 — Зависимость времени от размеров матриц.

Размер	Последовательный алгоритм, мс	Параллельный алгоритм, мс	Алгоритм Штрассена, мс
16x16	6	3	6
64x64	19	4	23
256x256	940	110	341
1024x1024	38694	6669	5102
2048x2048	539297	151128	24019

По таблице видно, что для матриц небольших размеров наиболее эффективен параллельный алгоритм, а для больших размеров — алгоритм Штрассена. Это обусловлено тем, что алгоритм Штрассена использует большое количество копирований, что замедляет его при небольших размерах.

Выводы.

В ходе работы был исследован алгоритм Штрассена для быстрого умножения матриц.

Было установлено, что алгоритм Штрассена эффективен при больших размерах матриц и неэффективен при малых, т. к. использует большое количество копирований. Алгоритм параллельного умножения работает быстрее до размера 512×512 , при больших размерах быстрее оказывается алгоритм Штрассена.