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

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Параллельные алгоритмы»

Тема: Параллельное умножение матриц

Студент гр. 9303 _____ Куршев Е.О.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Изучить принципы параллельного умножения матриц на языке C++, реализовать алгоритм «быстрого» умножения Штрассена.

Задание.

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

2. Реализовать параллельный алгоритм “быстрого” умножения матриц (Штрассена или его модификации).

Проверить, что результаты вычислений реализаций 4.1 и 4.2 совпадают.

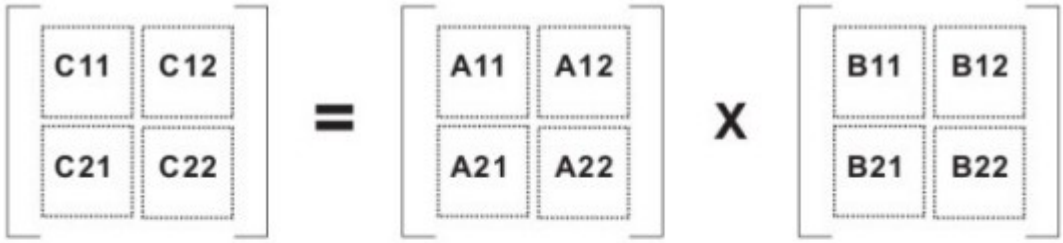
Сравнить производительность с реализацией 4.1 на больших размерностях данных (порядка 10^4 – 10^6)

Выполнение работы.

Было реализовано параллельное умножение матриц с помощью «простого» условного разбиения матрицы на столбцы и строки. На вход программе подаётся число потоков, которые равномерно умножают две матрицы. Было проведено исследование зависимости времени выполнения работы от размера матрицы и числа потоков. Результаты представлены в табл 1.
Таблица 1 — зависимость времени выполнения работы от числа потоков и размер матрицы

	2	3	4	5
64x64	0.013046s	0.011679s	0.011812s	0.011485s
128x128	0.061048s	0.065087s	0.063887s	0.063887s
256x256	0.273942s	0.242319s	0.262645s	0.252298s
512x512	2.30559s	2.31489s	2.15665s	2.07145s

Затем был реализован быстрый алгоритм Штрассена для умножения матриц. Схема алгоритма представлена на рисунке ниже.



$$
\begin{aligned}
M_1 &= (A_{11} + A_{22})(B_{11} + B_{22}) \\
M_2 &= (A_{21} + A_{22})B_{11} \\
M_3 &= A_{11}(B_{12} - B_{22}) \\
M_4 &= A_{22}(B_{21} - B_{11}) \\
M_5 &= (A_{11} + A_{12})B_{22} \\
M_6 &= (A_{21} - A_{11})(B_{11} + B_{12}) \\
M_7 &= (A_{12} - A_{22})(B_{21} + B_{22})
\end{aligned}
\qquad
\begin{aligned}
C_{11} &= M_1 + M_4 - M_5 + M_7 \\
C_{12} &= M_3 + M5 \\
C_{21} &= M_2 + M_4 \\
C_{22} &= M_1 - M_2 + M_3 + M_6
\end{aligned}
$$

Алгоритм работает с матрицами размерами 2^nx2^n. Данный алгоритм позволяет уменьшить сложность с $O(n^3)$ до $O(n^{2.81})$. Последовательность параллельного умножения матриц с помощью быстрого алгоритма Штрассена:

1. Изначально создаются 7 потоков для вычисления значений M;
2. Затем матрица делится на 4 части;
3. Повторяется рекуррентно до того, как не будет достигнута нужная глубина или не получен заданный примитив, который уже вычисляется, как обычное умножение матриц;
4. Затем полученные значения M рекуррентно собираются в матрицу C, в итоге получая окончательный ответ.

Было проведено исследование зависимости времени выполнения работы от размера матрицы и числа потоков для быстрого алгоритма Штрассена. Результаты представлены в табл 2.

Таблица 2 — зависимость времени выполнения работы от числа потоков и размер матрицы

	2	3	4	5
64x64	0.000153s	0.000154s	0.000256s	0.000217s
128x128	0.046784s	0.063958s	0.062984s	0.062872s
256x256	0.253893s	0.240289s	0.252384s	0.268904s
512x512	2.073893s	2.124253s	2.238922s	2.002393s

Выводы.

В ходе выполнения лабораторной работы были изучены принципы параллельного умножения матриц на языке C++, был реализован алгоритм «быстрого» умножения Штрассена