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. 0303	____________________	Морозов А.Ю.
Преподаватель	____________________	Сергеева Е.И.

Санкт-Петербург
2023

Цель работы.

Реализовать параллельный алгоритм умножения матриц с масштабируемым разбиением по потокам и параллельный алгоритм «быстрого» умножения матриц (алгоритм Штрассена). Проверить правильность вычислений и сравнить производительность алгоритма умножения матриц с масштабируемым разбиением с реализацией из первой лабораторной работы.

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.

В классе *Matrix* были перегружены операторы «+» и «-» для удобной реализации операций сложения и вычитания матриц, используемых в алгоритме «быстрого» умножения матриц (алгоритме Штрассена).

Для умножения матриц с масштабируемым разделением по потокам была реализована функция *scalable_resolve,* принимающая в себя две матрицы множителя, ссылку на матрицу, в которую будет записываться результат и количество умножающих потоков. Элементы результирующей матрицы поровну делятся между потоками и считаются параллельно.

Для «быстрого» умножения матриц была реализована функция *strassen_resolve,* принимающая в себя две матрицы множителя, ссылку на матрицу, в которую будет записываться результат. Исходные матрицы разделяются на 4 равных квадратных матрицы при помощи функции *split.*

Исходные матрицы делятся на четыре равных квадратных матрицы (см. рис. 1).

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix},\ \mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix}$$

Рисунок 1 – Разбиение матриц.

Далее считаются переходные матрицы по следующим формулам (см. рис. 2).

$$\begin{aligned}
\mathbf{P}_1 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2}) \\
\mathbf{P}_2 &:= (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1} \\
\mathbf{P}_3 &:= \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2}) \\
\mathbf{P}_4 &:= \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1}) \\
\mathbf{P}_5 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2} \\
\mathbf{P}_6 &:= (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2}) \\
\mathbf{P}_7 &:= (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2})
\end{aligned}$$

Рисунок 2 – Формулы для переходных матриц.

Результирующая матрица получается по формулам, представленным на рисунке ниже (см. рис. 3).

$$\begin{aligned}
\mathbf{C}_{1,1} &= \mathbf{P}_1 + \mathbf{P}_4 - \mathbf{P}_5 + \mathbf{P}_7 \\
\mathbf{C}_{1,2} &= \mathbf{P}_3 + \mathbf{P}_5 \\
\mathbf{C}_{2,1} &= \mathbf{P}_2 + \mathbf{P}_4 \\
\mathbf{C}_{2,2} &= \mathbf{P}_1 - \mathbf{P}_2 + \mathbf{P}_3 + \mathbf{P}_6
\end{aligned}$$

Рисунок 3 – Сборка результирующей матрицы.

Каждый поток считает свою переходную матрицу. Если размерность переходной матрицы велика, то рекурсивно вызывается функция, считающая ее тем же самым способом. Когда размерность матрицы мала, используется обычное умножение матриц.

Исследование.

Исследуем скорость работы алгоритма умножения матриц с масштабируемым разбиением с реализацией из первой лабораторной работы (см. табл. 1).

Таблица 1 – Сравнение 4.1 и реализации из первой лабораторной работы.

Размер матрицы	Лабораторная 1, сек.	4.1, сек.
32	0.000873	0.0013
64	0.0026	0.0036
128	0.013	0.014
256	0.07	0.075
512	0.557	0.61
1024	4.93	5.22
2048	47.722	60.725

Результаты совпадают, так как при реализации алгоритма в первой лабораторной работе был взят такой же алгоритм разбиения элементов результирующей матрицы между вычисляющими потоками.

Исследуем производительность параллельного алгоритма умножения матриц с масштабируемым разбиением по потокам и алгоритма Штрассена (см. табл. 2).

Таблица 2 – Сравнение алгоритма с масштабируемым разбиением и Штрассена.

Размер матрицы	Масштабируемый, сек.	Штрассен, сек.
32	0.0013	0.0017
64	0.0036	0.0076
128	0.014	0.018
256	0.075	0.112
512	0.61	0.761
1024	5.22	5.232
2048	60.725	36.165

При увеличении размера перемножаемых матриц алгоритм Штрассена начинает работать быстрее, поскольку имеет меньшую сложность ($n^{2.81}$ против n^3). На маленьких размерах матриц выигрывает алгоритм с масштабируемым разбиением по потокам.

Для проверки корректности была реализована функция *binary_compare* сравнивающая каждый байт в файлах, в которые производится вывод результатов умножения всех сгенерированных пар матриц.

Выводы.

В ходе выполнения лабораторной работы были реализованы параллельный алгоритм умножения матриц с масштабируемым разбиением по потокам и параллельный алгоритм «быстрого» умножения матриц (алгоритм Штрассена). Была проверена корректность вычислений обоих алгоритмов и проведено сравнение производительности алгоритма умножения матриц с масштабируемым разбиением с реализацией из первой лабораторной работы и с алгоритмом «быстрого» умножения матриц (алгоритмом Штрассена).