

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. 9304

Аксёнова Е.А.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Реализовать параллельный алгоритм умножения матриц и параллельный алгоритм “быстрого” умножения.

Задание.

Реализовать параллельный алгоритм умножения матриц.

Выполнение работы.

Реализация параллельного алгоритма умножения матриц.

Для параллельного умножения матриц задается количество потоков.

Исходя из количества потоков рассчитывается сколько строк левой матрицы и столбцов правой матрицы будет передано в каждый поток (в приведенном алгоритме ищутся делители числа потоков).

Далее с помощью вложенных циклов распределяем данные между потоками, чтобы каждая строка была умножена с каждым столбцом количество строк и столбцов для умножения в потоке и индекс цикла передается в поток.

С помощью переданных индексов и количеств строк и столбцов для вычисления в одном потоке вычисляются номер строки начала, строки конца, столбца начала, столбца конца передаются потоку, которые передаются в функцию умножения по частям.

Вложенными циклами проходимся по строке и столбцу, вычисляя сумму попарного произведения элементов, и записываем в ячейку матрицы с индексом [номер строки, номер столбца].

Главный поток join'ит все умножающие потоки, чтобы дождаться завершения умножения перед записью в файл.

Исследовать масштабируемость выполненной реализации.

В таблице 1 представлено сравнение времени работы алгоритма в зависимости от размера входных данных.

Размер данных	Время работы алгоритма
256x128, 128x312	0.050085
512x1024, 1024x512	2.044621
1x1024, 1024x1	0.000467
1024x1, 1x1024	0.012887

Реализация алгоритма Штрассена.

Для реализации алгоритма Штрассена была написана функция, которая:

1. Если количество колонок(строк) в матрице меньше 64, то происходит обычное умножение матриц. (Выбрано 64, так как после такой размерности алгоритм Штрассена не эффективен)
2. Если глубина опускания в рекурсию > 3 , 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 алгоритма представлена на рисунке 1.

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

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

Рисунок 1 - Алгоритм Штрассена

Проверить, что результаты вычислений реализаций 4.1 и 4.2 совпадают.

Для проверки правильности работы алгоритма было произведено умножение матриц (представлены ниже) обоими алгоритмами.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

64	63	62	61	60	59	58	57
56	55	54	53	52	51	50	49
48	47	46	45	44	43	42	41
40	39	38	37	36	35	34	33
32	31	30	29	28	27	26	25
24	23	22	21	20	19	18	17
16	15	14	13	12	11	10	9
8	7	6	5	4	3	2	1

На рисунке 2 представлены результаты обоих вычислений. Как можно заметить, они совпадают.

672	644	616	588	560	532	504	476
2976	2884	2792	2700	2608	2516	2424	2332
5280	5124	4968	4812	4656	4500	4344	4188
7584	7364	7144	6924	6704	6484	6264	6044
9888	9604	9320	9036	8752	8468	8184	7900
12192	11844	11496	11148	10800	10452	10104	9756
14496	14084	13672	13260	12848	12436	12024	11612
16800	16324	15848	15372	14896	14420	13944	13468
672	644	616	588	560	532	504	476
2976	2884	2792	2700	2608	2516	2424	2332
5280	5124	4968	4812	4656	4500	4344	4188
7584	7364	7144	6924	6704	6484	6264	6044
9888	9604	9320	9036	8752	8468	8184	7900
12192	11844	11496	11148	10800	10452	10104	9756
14496	14084	13672	13260	12848	12436	12024	11612
16800	16324	15848	15372	14896	14420	13944	13468

Рисунок 2 - Совпадение результатов алгоритмов

Сравнить производительность с реализацией 4.1 на больших размерностях данных.

В таблице 2 представлены результаты сравнения производительности алгоритма Штрассена и обычного умножения матриц.

Таблица 2 - Сравнение производительности алгоритмов

Размер матрицы	Время работы алгоритма умножения	Время работы алгоритма Штрассена
128	0.011968	0.013663
512	0.950430	0.669849
2048	275.624854	28.587471

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

Выводы.

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