

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. 1304

Чернякова В.А.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2024

Цель работы.

Изучить умножения матриц с помощью алгоритма Штрассена с разбиением на потоки и сравнить производительность разных алгоритмов.

Задание.

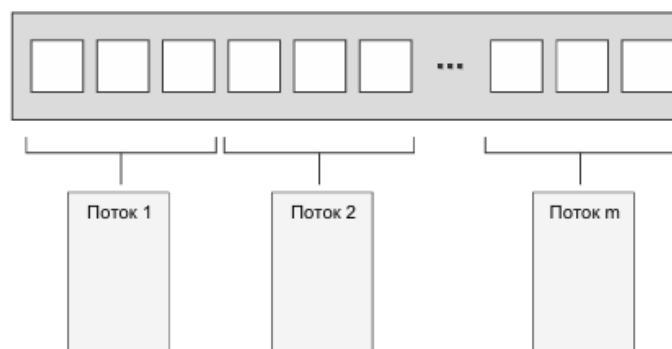
4.1 Реализовать параллельный алгоритм умножения матриц с блочным разбиением по потокам. Исследовать масштабируемость выполненной реализации, сравнить с реализацией из работы 1.

4.2 Реализовать параллельный алгоритм “быстрого” умножения матриц (Штрассена или его модификации). Тестирование: проверить, что результаты вычислений реализаций 4.1 и 4.2 совпадают (в том числе на больших размерностях). Сравнить производительность с реализацией 4.1 на больших размерностях данных (порядка $10^4 - 10^6$).

Выполнение работы.

Задание 4.1

Простейший способ распределения данных заключается в том, чтобы назначить первые N элементов одному потоку, следующие N – другому и так далее.



Как бы ни были распределены данные, каждый поток обрабатывает только назначенные ему элементы, никак не взаимодействуя с другими потоками до тех пор, пока не завершит обработку.

В реализации параллельного умножения матриц общее количество элементов финальной матрицы делится на блоки равного размера между P

потоками. Если невозможно нацело разделить количество умножений, остаток операций выполняется в последнем потоке. С помощью этого достигался баланс загрузки каждого потока и равномерное вычисление.

Задание 4.2

Штрассен придумал сложный набор соотношений, которые позволили заменить одно из этих восьми умножений 14 дополнительными сложениями.

Смысл этого многократного разбиения больших матриц на более мелкие заключается в том, что можно снова и снова применять алгоритм Штрассена к меньшим матрицам и с помощью его метода сокращать количество шагов на каждом этапе. В целом алгоритм Штрассена увеличил скорость умножения матриц с n^3 до $n^{2.81}$ мультипликативных шагов.

Если добавить к матрицам A и B одинаковые нулевые строки и столбцы, их произведение станет равно матрице AB с теми же добавленными строками и столбцами. Поэтому можно рассматривать только матрицы размера $n = 2^k, k \in \mathbb{N}$, а другие случаи сводить к этому добавлением нулей, отчего n может увеличиться лишь вдвое.

Пусть A, B – матрицы размера $2^k \times 2^k$. Их можно представить как блочные матрицы размера (2×2) из $(2^{k-1} \times 2^{k-1})$ матриц.

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

По принципу блочного умножения, матрица AB выражается через их произведение:

$$AB = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix},$$

где в правой части происходит восемь умножений матриц размера $2^{k-1} \times 2^{k-1}$. Поскольку матрицы образуют кольцо, то для вычисления правой части годится любой алгоритм умножения (2×2) - матриц, использующий лишь сложения, вычитания и умножения. Штрассен предложил такой алгоритм с семью умножениями:

$$\begin{aligned}
D &= (A_{11} + A_{22})(B_{11} + B_{22}); \\
D_1 &= (A_{12} - A_{22})(B_{21} + B_{22}); \\
D_2 &= (A_{21} - A_{11})(B_{11} + B_{12}); \\
H_1 &= (A_{11} + A_{12})B_{22}; \\
H_2 &= (A_{21} + A_{22})B_{11}; \\
V_1 &= A_{22}(B_{21} - B_{11}); \\
V_2 &= A_{11}(B_{12} - B_{22}); \\
AB &= \begin{pmatrix} D & 0 \\ 0 & D \end{pmatrix} + \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} + \begin{pmatrix} -H_1 & H_1 \\ H_2 & -H_2 \end{pmatrix} + \begin{pmatrix} V_1 & V_2 \\ V_1 & V_2 \end{pmatrix} \\
&= \begin{pmatrix} D + D_1 + V_1 - H_1 & V_2 + H_1 \\ V_1 + H_2 & D + D_2 + V_2 - H_2 \end{pmatrix}.
\end{aligned}$$

Каждое умножение можно совершать рекурсивно по той же процедуре, а сложение – тривиально, складывая $(2^{k-1})^2$ элементов.

Сравнение скорости умножения

Были проведены измерения времени от размеров вычисляемых матриц, результаты приведены в таблице. Количество потоков для простого и параллельного алгоритма 7.

Размер	Простой алгоритм, мс	Параллельный алгоритм, мс	Алгоритм Штрассена, мс
16 × 16	1.51015	0.55202	0.116481
32 × 32	0.655445	0.579161	0.685582
64 × 64	2.56756	1.1307	1.46393
128 × 128	9.04588	4.31222	11.4782
256 × 256	65.2177	35.6696	31.7894
512 × 512	615.772	334.645	237.683
1024 × 1024	4918.72	2668.66	1796.51
2048 × 2048	43059.5	22126.1	12056.3
4096 × 4096	332325	165552	81226

Как можно заметить, алгоритм Штрассена превосходит параллельный алгоритм по производительности, когда сторона матрицы превосходит 256. До

этого размера преимущество имеет параллельный алгоритм. Когда сторона матрицы меньше 256 - в алгоритме Штрассена происходит много копирования и вставки матриц, из-за чего его производительность падает на фоне параллельного алгоритма, которому копирование не требуется. Когда сторона матрицы больше 256 - алгоритм Штрассена выигрывает за счёт меньшего числа перемножений матриц (7 против 8).

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 256×256 , при больших размерах быстрее оказывается алгоритм Штрассена.