

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

Нагибин И.С

Преподаватель

Сергеева Е.И

Санкт-Петербург

2023

Цель работы.

Изучение алгоритмов параллельного умножения матриц. Изучение алгоритма Штрассена.

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. Добавлены новые функции для работы с матрицами.

```
Matrix addMatrix(Matrix &m1, Matrix &m2);
Matrix subMatrix(Matrix &m1, Matrix &m2);
Matrix slice(Matrix &matrix,
             std::size_t piece,
             std::size_t startRow,
             std::size_t startCol);
void insert(Matrix& matrix,
            Matrix& matrixToInsert,
            std::size_t startRow,
            std::size_t startCol);

inline Matrix operator+(Matrix &m1, Matrix &m2)
{
    return addMatrix(m1, m2);
}

inline Matrix operator-(Matrix &m1, Matrix &m2)
{
    return subMatrix(m1, m2);
}
```

Рисунок 1 — добавленные функции для работы с матрицами.

Добавлены функции сложение и вычитания матриц, также добавлены аналогичные операторы. Добавлены функции для вырезания и вставки подматрицы.

2. За алгоритмы перемножения отвечают следующие функции:

- `badConcurentMultiplication` — алгоритм из первой лабораторной работы, разбивает матрицу по последовательному набору строк.
- `goodConcurentMultiplication` — алгоритм из второй лабораторной работы, разбивает матрицу по элементам, каждый поток вычисляет свои элементы.
- `StrassenMultiplication` — алгоритм Штрассена для умножения матриц.

Алгоритм Штрассена на первом этапе выполняет задачи параллельно, после первого этапа — последовательно. Это позволяет не создавать большое количество потоков умножения, что может привести к снижению производительности. Если размеры матриц меньше порогового значения, объявленного глобальной переменной, то умножение производится последовательно.

2. Измерение времени работы алгоритмов умножения матриц.

Таблица 1. Время работы программ умножения матриц.

Размер матриц	Разбиение по строкам, сек	Разбиение по элементам, сек	Алгоритм Штрассена, сек
64x64	0.00070432	0.000502678	0.00403689
128x128	0.000796741	0.000568626	0.0135227
256x256	0.00416343	0.00473454	0.0410855
512x512	0.0388155	0.0382849	0.135857
1024x1024	0.351267	0.360628	0.577865
2048x2048	8.11186	8.36168	2.45038
4096x4096	80.003	80.8557	16.7038

8192x8192	1087.79	903.978	150.928
-----------	---------	---------	---------

Как видно по табл. 1, алгоритм Штрассена уступает по производительности при перемножении матриц размерами меньше 2048x2048, но начиная с этой размерности алгоритм показывает результаты производительности превосходящие другие представленные способы перемножения матриц.

Выводы.

В ходе работы были изучены алгоритмы параллельного умножения матриц, изучен алгоритм Штрассена.

Было проведено исследование скорости выполнения алгоритмов перемножения матриц в зависимости от их размеров. Алгоритмом Штрассена показывает наилучшую производительность при перемножении больших квадратных матриц (2048x2048 и более), размер которых является степенью 2. Алгоритмы с разбиением по строкам и элементам показывает лучшую производительность при перемножении матриц меньших 2048x2048.