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 _____ Середенков А.А.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург
2023

Цель работы.

Изучение и практическая реализация алгоритма Штрассена для перемножения матриц.

Задание.

1. Реализовать параллельный алгоритм умножения матриц с масштабируемым разбиением по потокам.

Исследовать масштабируемость выполненной реализации с реализацией из работы 1.

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.

Для выполнения данной лабораторной работы, был использован и расширен класс *Matrix* из предыдущих лабораторных работ. В данном классе были определены операторы суммы и вычитания для удобства дальнейших вычислений, а также сравнения для проверки итоговых вычислений.

Были реализованы функции:

- *parallel* – умножает, переданные в качестве аргументов матрицы, с масштабируемым разбиением по потокам;
- *strassen_alg* – умножает переданные на вход матрицы по алгоритму Штрассена;

Параллельный алгоритм умножения реализован следующим образом. каждый *i*-й поток вычисляет $i+k*n$ элементы результирующей матрицы, где n – общее количество потоков,, а k принимает значения 1, …, m ($k*n < m^2$, m – размерность результирующей матрицы).

Алгоритм Штрассена работает только с квадратными матрицами размерности степени 2. Поэтому были реализованы несколько вспомогательных методов (*prepare_mat, expand_mat*), которые добавляют нулевые столбцы и строки, чтобы расширить матрицу до нужного размера.

Алгоритм Штрассена вычисляет следующие вспомогательные матрицы:

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

На основе этих вспомогательных матриц, вычисляются элементы результирующей матрицы:

$$
\begin{pmatrix}
D+D_1+V_1-H_1 & V_2+H_1 \\
V_1+H_2 & D+D_2+V_2-H_2
\end{pmatrix}
$$

Сравним эти алгоритмы на тестовых данных, для умножения по строкам и масштабируемого умножения количество потоков равно 7. Результат времени работы программы представлен в табл. 1.

Таблица 1 — Время работы алгоритмов.

Размерность матрицы	Параллельное умножение по строкам	Масштабируемое параллельное умножение	Алгоритм Штрассена
128×128	20 мс	12 мс	17 мс
256×256	67 мс	56 мс	44 мс
512×512	432 мс	423 мс	256 мс
1024×1024	4796 мс	5543 мс	1722 мс
2048×2048	65331 мс	57048 мс	13735 мс
4096×4096	939409 мс	1019146 мс	131825 мс

Исходя из результатов таблицы можно сделать вывод, что масштабируемое параллельное умножение работает быстрее, чем алгоритм Штрассена при небольших размерностях. Алгоритм Штрассена дает работает намного быстрее на больших плотных матрицах. В среднем масштабируемой параллельное умножение матриц работает столько же сколько и умножение матриц из лабораторной работы 1.

Выводы.

В процессе выполнения лабораторной работы были исследованы и реализованы алгоритмы перемножения матриц – масштабируемое параллельное умножение и алгоритм Штрассена.