МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №4
по дисциплине «Параллельные алгоритмы»
Тема: Параллельное умножение матриц

Студент гр. 0304	____________________	Максимов Е.А.
Преподаватель	____________________	Сергеева Е.И.

Санкт-Петербург
2023

Цель работы.

Исследование производительности параллельных алгоритмов.

Постановка задачи.

Реализовать параллельный алгоритм умножения матриц с масштабируемым разбиением по потокам. Исследовать масштабируемость выполненной реализации с реализацией из работы №1.

Реализовать параллельный алгоритм “быстрого” умножения матриц (Штрассена или его модификации). Проверить, что результаты вычислений реализаций совпадают. Сравнить производительность на больших размерностях данных (порядка 10^4 .. 10^6).

Выполнение работы.

Алгоритм умножения матриц с масштабируемым разбиением по потокам был взят из лабораторной работы №1. Для оптимальной работы алгоритма было определено количество потоков равное $\sqrt{N}$, где N — размер матрицы.

Для реализации алгоритма Штрассена были написаны функции, перечисленные ниже.

`void multiplyStrassen(vector<Matrix> matrices*, int N)`, `void multiplyStrassenRoutine(vector<Matrix> matrices*, int recursionDepth)` — основные функции, реализующие алгоритм Штрассена.

`Vector<Matrix> split(Matrix A)`, `Matrix join (vector<Matrix> Cs)` — обратные функции: разбивает матрицу на 4 подматрицы меньшего размера, объединяет 4 матрицы в одну (слева направо, сверху вниз).

`operator+(Matrix A, Matrix B)`, `Matrix operator-(Matrix A, Matrix B)` — вспомогательные функции для краткой записи сложения и вычитания объектов типа `Matrix`.

Максимальная глубина рекурсии равна 3, минимальный размер матрицы, при котором увеличивается глубина рекурсии, равен 4.

Если размер матрицы не равен 2^N, то перед работой алгоритма Штрассена матрица расширяется до минимально возможного N, пустые ячейки заполняются нулями, а после работы алгоритма уменьшается до первоначального размера.

Пусть A, B — входные матрицы. Вычисляются следующие промежуточные M_i матрицы, перечисленные ниже.

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

Умножение матриц производится рекурсивно этим же алгоритмом.

Затем вычисляется результирующая матрица C на основе полученных матриц M_i по формуле ниже.

$$
C = \begin{pmatrix} C_1 & C_2 \\ C_3 & C_4 \end{pmatrix} = \begin{pmatrix} M_1 + M_2 + M_6 - M_4 & M_7 + M_4 \\ M_6 + M_5 & M_1 + M_3 + M_7 - M_5 \end{pmatrix}.
$$

Для измерения времени работы алгоритмов была использована библиотека `chrono`. Результаты тестирования представлены в приложении А.

Выводы.

В ходе лабораторной работы было проведено исследования эффективности алгоритмов умножения матриц. Результаты показали, что алгоритм Штрассена эффективнее, чем умножение с масштабируемым разбиением по потокам, при больших размерностях матрицы ($N > 10^3$).

Практическим результатом лабораторной работы является программный код, реализующий алгоритм Штрассена.

ПРИЛОЖЕНИЕ А
ТЕСТИРОВАНИЕ

Таблица А1 — Сравнение эффективности программ

<table>
<tr><th>№</th><th>Размерность матрицы</th><th>Время работы с масштабируемым разбиением по потокам, мс</th><th>Время работы с алгоритмом Штрассена, мс</th></tr>
<tr><td>1</td><td>4</td><td><1</td><td>2</td></tr>
<tr><td>2</td><td>16</td><td>1</td><td>54</td></tr>
<tr><td>3</td><td>64</td><td>3</td><td>60</td></tr>
<tr><td>4</td><td>256</td><td>62</td><td>160</td></tr>
<tr><td>5</td><td>512</td><td>263</td><td>417</td></tr>
<tr><td>6</td><td>1024</td><td>1782</td><td>2302</td></tr>
<tr><td>7</td><td rowspan="2">2048</td><td rowspan="2">21723</td><td>18426</td></tr>
<tr><td>8*</td><td>15888</td></tr>
</table>

Примечание (*) — тест №8 был выполнен при создании потоков в функции, реализующей алгоритма Штрассена (см. п. «Выполнение работы»), только на 1-ой глубине рекурсии.