**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 |  | Аристархов И.Е. |
| Преподаватель |  | Сергеева Е.И. |

Санкт-Петербург

2023

**Цель работы.**

На практике освоить методы создания алгоритмов без блокировок.

**Поставленные задачи.**

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.**

Первый пункт был успешно выполнен еще в первой лабораторной работе, т.ч. он остается неизменным.

Реализуем умножение матриц алгоритмом Штрассена. Алгоритм подразумевает разбиение обеих матриц на 4 равные части: , , , и , , , соответственно. Чтобы такая операция была возможна на любой матрице, важно, чтобы размер такой матрицы был степенью двойки. Если это не так, то можно дополнить матрицу нулями до требуемого размера.

Затем считаются следующие 7 вспомогательных матриц:

Далее считаются четверти результирующей матрицы:

Очевидно, что операции умножения при вычислении также могут использовать этот алгоритм. Тем не менее это вряд ли будет эффективно. Поэтому алгоритм не используется, если размер очередной пары матриц менее 1/8 оригинального размера. Тогда используется обычное умножение матриц.

С точки зрения распараллеливания умножения достаточно будет создать потоки только для вычисления на самом высоком уровне. Если же создавать потоки для расчета на всех уровнях, то производительность сильно упадет.

**Анализ.**

Компьютер, на котором производились замеры, имел процессор 4 физическими ядрами и 8 потоками (по 2 потокам на ядро).

В качестве показателя производительности в данном случае использовалось время для перемножения матриц для масштабируемого алгоритма с количеством потоков, равным размеру матрицы (анализ в 1 работе показал, что это количество потоков наиболее эффективное) и для алгоритма Штрассена.

Были получены следующие результаты:

|  |  |  |
| --- | --- | --- |
| Размер матрицы | Время на масштабируемом алгоритме, сек | Время на алгоритме Штрассена, сек |
| 128 | 0,064 | 0,175 |
| 256 | 0,486 | 0,677 |
| 512 | 1,902 | 2,883 |
| 1024 | 14,267 | 13,569 |
| 2048 | 119,558 | 61,773 |

Были взяты размеры матрицы степенями двойки, т.к. алгоритм Штрассена первым шагом доводит размер матрицы до степени двойки.

Из таблицы видно, что на матрицах малого размера алгоритма Штрассена работает хуже масштабируемого, но чем больше матрица, тем эффективнее становится алгоритм Штрассена.

**Выводы.**

В результате выполнения лабораторной работы была разработана программа производящая умножение матрицы с применением масштабируемого алгоритма и алгоритма Штрассена. Последний алгоритм сравнивался с первым в производительности.

Опыты показали, что алгоритм Штрассена наиболее эффективен на матрицах большого размера (1024 и выше).