

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. 1304

Кардаш Я.Е.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2024

Цель работы.

Изучить параллельное умножение матриц, сравнить разные алгоритмы умножения.

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 $10^4 - 10^6$)

Дополнительно (на отл.) реализация 4.2 должна быть масштабируема (работать на разном количестве потоков)

Выполнение работы.

1 Реализован алгоритм блочного умножения матриц. При умножении матрица разбивается на блоки, что позволяет распараллелить процессы умножения и выполнять операцию быстрее

2 Реализован алгоритм быстрого умножения Штрассена. При выполнении операции часть умножений заменяется эквивалентными операциями сложения. Это позволяет снизить алгоритмическую сложность, а также значительно ускорить операцию, при условии, что для вычислительной среды (компьютера) умножение является значительно более дорогой операцией, чем сложение

Проведены тесты двух алгоритмов, проверено, что результаты, записываемые в выходные файлы, совпадают (написана функция сравнения файлов).

Проведено нагрузочное тестирование для размерности матриц $\text{size} = 2048$ при количестве потоков от 1 до 6. Результаты отображены на рисунке 1.

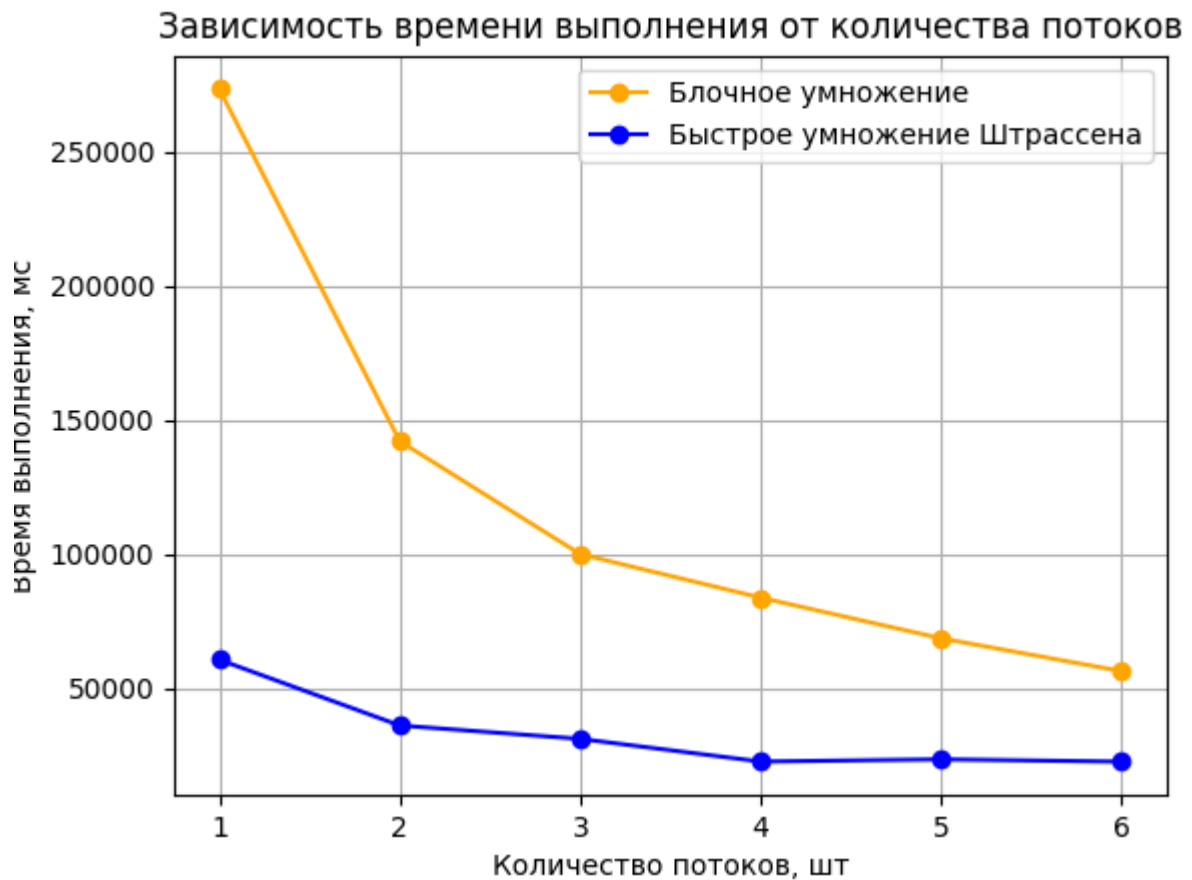


Рисунок 1. Результаты тестирования

Из результатов можно увидеть, что алгоритм Штрассена при больших размерностях работает значительно быстрее, при любом количестве потоков

В качестве дополнительного задания в реализацию алгоритма Штрассена добавлен пул потоков. Реализация пула потоков взята из библиотеки CTPL. В рекурсивную функцию передается объявленный пул потоков (`ctpl::thread_pool`). Перед переходом к вычислению ветвей рекурсии проверяется наличие свободных потоков, при их наличии ветвь рекурсии помещается на выполнение в пул потоков, иначе ветвь рекурсии выполняется в том же потоке, в котором она была вызвана. Проведено аналогичное с рисунком 1 тестирование, результаты на рисунке 2.

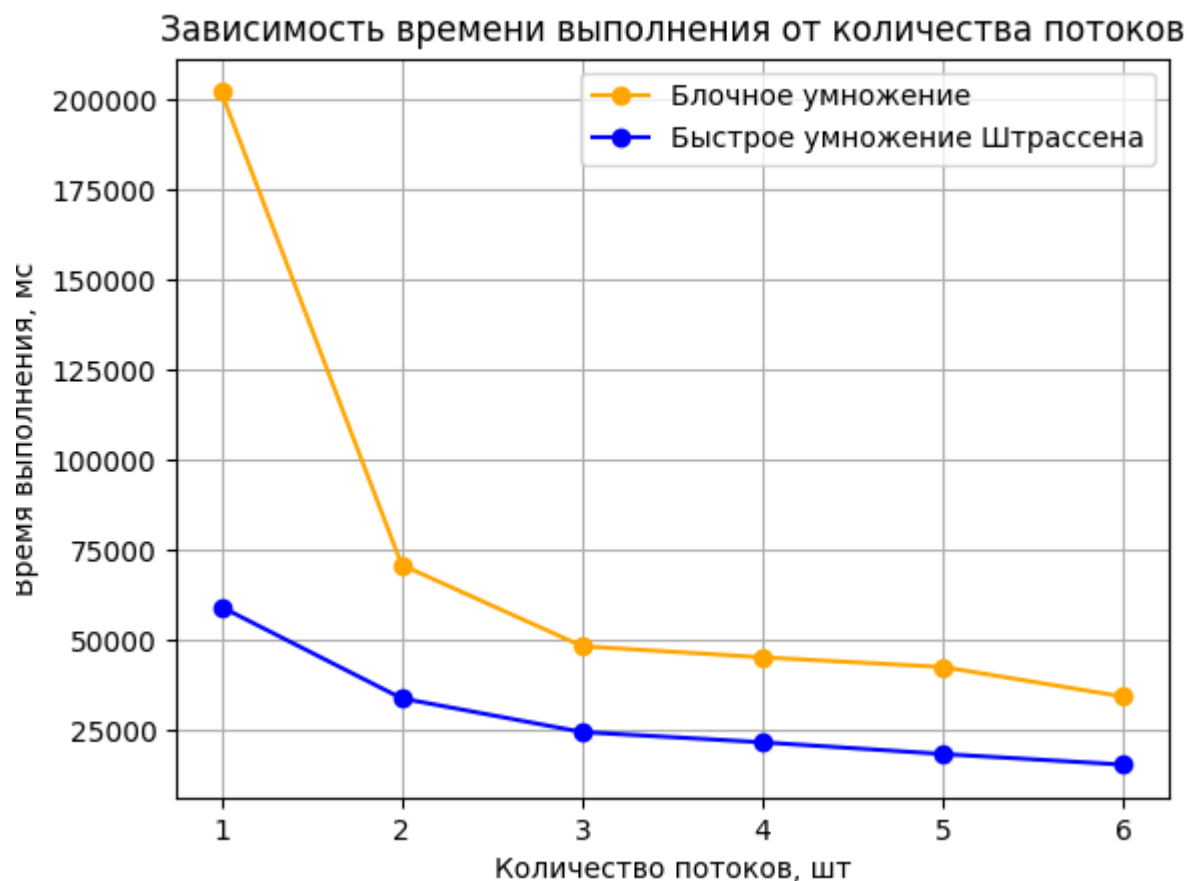


Рисунок 2. Результаты тестирования с пулом потоков.

Выводы.

Изучены алгоритмы параллельного умножения матриц – блочный алгоритм и алгоритм Штрассена.

Установлено, что до размерности матрицы 1024 включительно алгоритм Штрассена работает с такой же скоростью, что и алгоритм блочного умножения. При больших размерах матрицы алгоритм Штрассена начинает работать значительно быстрее.

Проведено нагрузочное тестирование при размерности матриц 2048 при количестве потоков от 1 до 6. Установлено, что при больших размерностях алгоритм Штрассена работает быстрее, при любом количестве потоков.