МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
"ЛЭТИ" ИМ. В.И.УЛЬЯНОВА(ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЁТ

по лабораторной работе № 5
по дисциплине «Параллельные алгоритмы»
Тема: «Умножение матриц на GPU»

Студент гр.1304 _____ Поршнев Р.А.
Преподаватель _____ Сергеева Е.И.

Санкт-Петербург
2024

Задание

Реализовать блочное умножение матриц в стандарте OpenCL (или CUDA) с учётом оптимизаций доступа к локальной и глобальной памяти. Выполнить тестирование: сравнение результатов вычислений с полученными в работе 4. В отчете: Произвести сравнение производительности с CPU реализациями сделанными в лаб. работе 4.

Выполнение работы.

Для выполнения данной лабораторной работы за основу был взят код предыдущей лабораторной работы, за исключением функций, необходимых для непосредственного перемножения матриц. Также была реализована функция, которая двумерный массив преобразуют в одномерный, что удобнее при работе с OpenCL, а также функция, которая одномерный массив преобразовывает в одномерный с учетом заданного количества строк и столбцов. Для перемножения матриц реализована программа на OpenCL, которая запускается с хоста запускаемой программы.

Для запуска программы и выполнения вычислений необходимо было выполнить следующее:

- определить доступные платформы и выбрать одну из них;
- определить доступные девайсы для одной из платформ и выбрать один из них;
- создать контекст для одного из девайсов;
- создать программу на основе исходного кода OpenCL и контекста;
- создать буфер для левой и правой матриц, а также для результат их произведения;
- отправить в очередь буфер для левой и правой матрицы;
- отправить в очередь команду создания ядра для выполнения перемножения матриц;

• отправить в очередь команду для запуска ядра;

• отправить в очередь команду для считывания результата перемножения матриц.

Также было проведено сравнение результатов с вычислениями, полученными в предыдущей работе. Тестирование проводилось для матриц порядка 64, 128, 256, 512 и 1024. Для сравнение результатов был реализован скрипт на bash, который с помощью команды diff с флагом -s устанавливал являются ли файлы с результатами вычислений в данной и в предыдущей работе индентичными для матриц разного порядка.

Исследована зависимость времени умножения матриц от размера блока для матриц порядка 1024. Результаты представлены в Таблице 1.

Таблица 1 – Зависимость времени умножения матриц от размера блока для матриц порядка 1024

Время умножения, нс	Размер блока
1311951024	1
211429179	2
40154228	4
12037884	8
8489497	16
8139158	32

Таким образом, можно сделать вывод, что видеокарта NVIDIA GeFORCE GTX 1650 поддерживает рабочие группы размером до 1024 элементов и данный размер группы для задачи перемножения матриц является наилучшим.

На Рисунке 1 представлены результаты сравнения производительности GPU-алгоритма, блочного и алгоритма Штрассена.

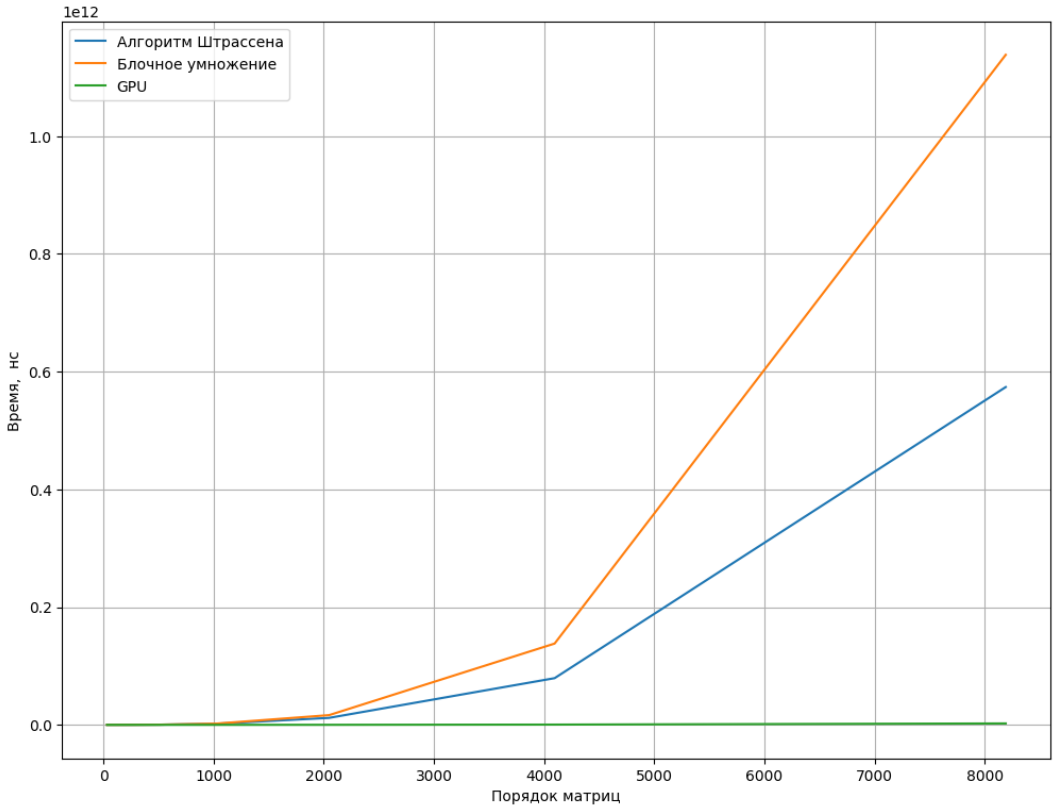

Рисунок 1 – Зависимость времени умножения матриц трёх алгоритмов при измененеии порядка матриц

Более детальный график зависимости для GPU представлен на Рисунке 2.

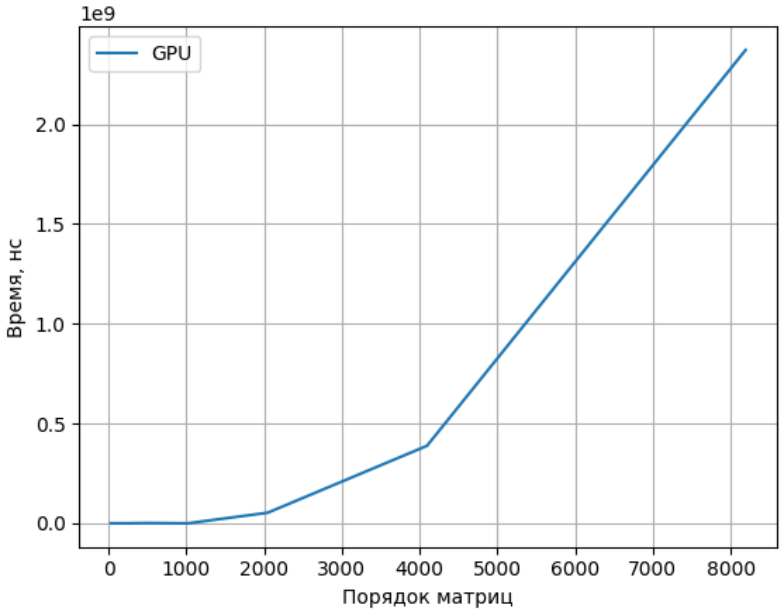

Рисунок 2 – Зависимость времени умножения матриц для GPU-алгоритма от порядка матриц

Исходя из полученных данных, можно сделать вывод, что GPU-алгоритм перемножения матриц значительно быстрее алгоритмов на CPU.

Вывод

В ходе выполнения работы изучено применение видеокарт в UNIX-подобных системах для решения практической задачи: перемножение матриц. Было выполнено следующее:

• Построена таблица зависимости времени умножения матриц от размера блока для матрицы порядка 1024, исходя из которой можно сделать вывод, что оптимальный размер блока бля умножения больших матриц равен 32.

• Проведено сравнение алгоритма перемножения матриц на GPU, а также блочного алгоритма и алгоритма Штрассена на CPU, в результате которого было установлено, что алгоритм на GPU значительно быстрее алгоритмов на CPU, что связано с тем, что видеокарта представляет собой большое количество вычислительных блоков, которые состоят из большого количество ALU.

Реализована host-программа на C++, которая на первой доступной платформе и на первом доступном девайсе запускет реализованный алгоритм перемножения матриц на GPU на языке OpenCL.