

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

ОТЧЕТ
по лабораторной работе №6
по дисциплине «Параллельные алгоритмы»
Тема: Оптимизация доступа к памяти в модели OpenCL.

Студент гр. 0304

Алексеев Р.В.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2023

Цель работы.

Изучить механизмы доступа к памяти в модели OpenCL.

Задание.

- Реализовать умножение матриц на OpenCL
- Произвести сравнение производительности с CPU реализацией из лаб. 4.

Выполнение работы.

Для умножения матриц при помощи OpenCL был написан kernel код. Т.к. при классическом умножении матриц требуется многократный доступ к одним и тем же элементам, то в kernel-коде необходимые ячейки матрицы были сохранены в локальной памяти, т. к. работа с глобальной памятью требует больших затрат. Для оптимизации умножения матрицы разбиваются на подматрицы одинакового размера. Каждая work-группа занимается умножением внутри одной подматрицы, каждый work-item занимается умножением одной строки на один столбец.

Т.к. умножаемые матрицы имеют одинаковый размер, то для вычисления итогового произведения можно посчитать сумму произведений строк и столбцов в каждой соответствующей подматрице.

При умножении матриц необходимо, чтобы их размеры нацело делились на размеры подматриц, поэтому в случае, если это условие не выполняется, матрицы дополняются нулями, что не влияет на итоговый результат.

Были произведены замеры времени, необходимого для умножения матриц различного размера (размеры левой и правой матриц одинаковы) на CPU и на GPU, результаты представлены в таблице 1.

Таблица 1 — Зависимость времени от размеров матриц.

Размеры матриц	Время на CPU, мс	Время на GPU, мс
128x128	15	145
256x256	79	104
512x512	521	127
1024x1024	3487	225
2048x2048	23116	678

По таблице видно, что при небольших размерах умножаемых матриц алгоритм Штарссена оказывается быстрее, что обусловлено временными затратами на подготовку к обработке кода и передаче данных в GRAM, но при росте размеров этот алгоритм оказывается медленнее алгоритма на OpenCL, т. к. умножения происходят последовательно.

Выводы.

Была изучена работа с глобальной и локальной памятью в OpenCL.

Была реализована программа, умножающая матрицы параллельно при помощи OpenCL.

Было произведено сравнение временных затрат на умножение при различных размерах матриц при помощи алгоритма Штрассена на CPU и при помощи OpenCL. Было установлено, что при небольших размерах, алгоритм Штрассена работает быстрее, т. к. не требует передачи данных в GRAM, но при увеличении размеров алгоритм на OpenCL оказывается быстрее.