

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. 0303

Сологуб Н.А.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2023

Цель работы.

1) Реализовать умножение матриц на OpenCL

В отчете: Произвести сравнение производительности с CPU реализацией из лаб.4

Выполнение работы.

1) Функции файла main.cpp

Были реализованы функции, реализующие отрисовку фрактала Мандельброта с помощью GPU. Функция **std::string get_program_text()** читает код из файла `opencl`. Функция **cl_program build_program(cl_context ctx, cl_device_id device)** генерирует указатель на исполняемую программу `opencl`. Функция **cl_device_id create_device()** создаёт устройство, которое будет выполнять команды `opencl` (в данном случае GPU). Функция **void invoke_kernel(cl_kernel kernel, cl_command_queue queue, cl_mem& A, cl_mem& B, cl_mem& buffer, cl_int* result, const int size)** запускает умножение матриц.

2) Функция умножения матриц

Была реализована функция для умножения матриц. Функция **__kernel void mult(__global int* A, __global int* B, __global int* C, int size)**

получает локальные и глобальные позиции строки и столбца результирующей матрицы. Затем задача разделяется на $n = \frac{size}{localSize}$ блоков, которые записываются в локальную память. За счёт большей эффективности локальной памяти скорость вычисления происходит быстрее. Принцип работы представлен на рис. 1

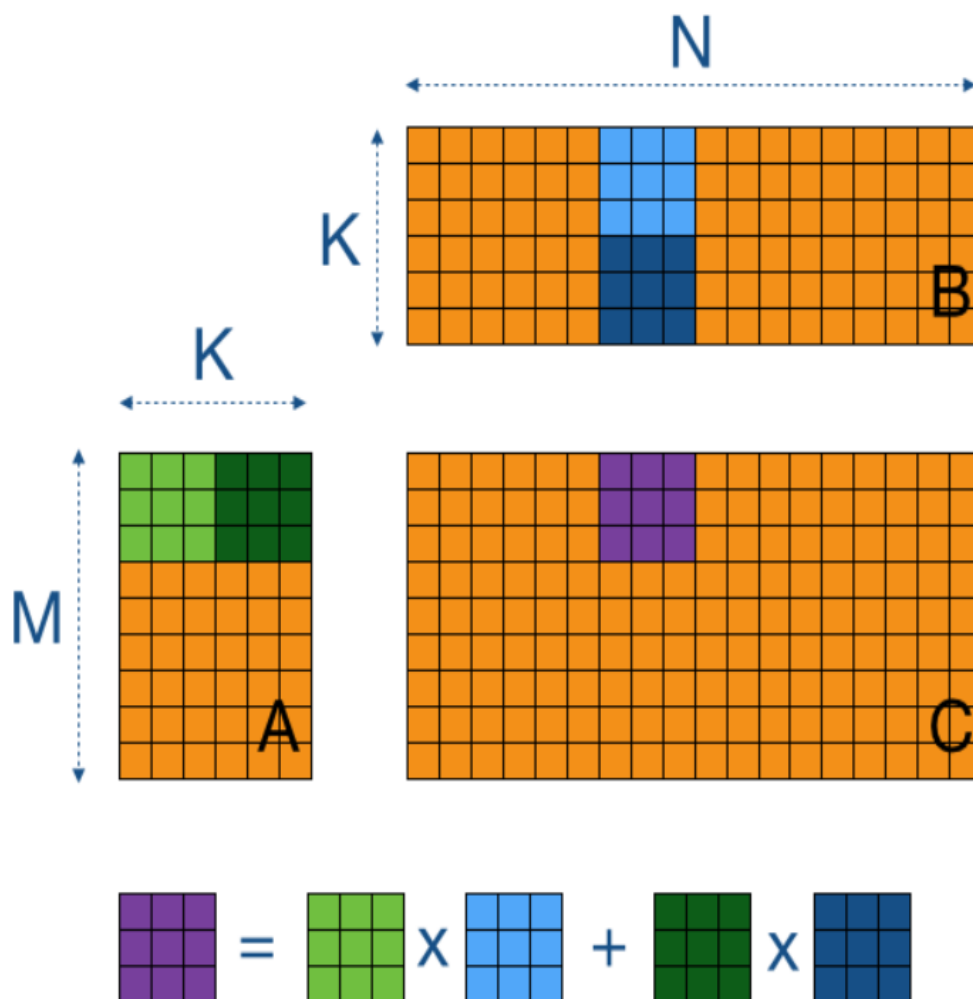


Рисунок 1 - разделение матрицы на блоки локальной памяти

3) Сравнение скорости умножения матриц

Произведем сравнение скорости умножения матриц с помощью GPU и CPU в зависимости от размерности матрицы. Результаты представлены в табл. 1.

Таблица 1 - зависимость времени от размерности матрицы и алгоритма умножения

Размерность матриц	Время масштабированного параллельного	Время параллельного умножения, с	Время параллельного умножения Штрассена, с	Время умножения с помощью GPU, с

	умножения, с			
32	0.0012997	0.0013364	0.0040139	0.0005445
64	0.0017698	0.002104	0.0079131	0.0005493
128	0.0070457	0.0089909	0.0216973	0.0005927
256	0.0358545	0.0603649	0.0848752	0.0008215
512	0.213983	0.605712	0.381595	0.0036493
1024	2.36056	4.39453	2.08541	0.0192117
2048	22.2422	29.8992	13.8839	0.169964
4096	203.339	186.812	148.463	1.34361

Из результатов видно, что GPU работает примерно в 150 раз быстрее за счёт параллельной обработки данных при умножении строки на столбец.

Выводы.

В ходе выполнения работы было изучено средство OpenCL для решения задач общей направленности на GPU. Было произведено умножения матриц средствами OpenCL и оценена производительность по сравнению с CPU.