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

ОТЧЕТ

по лабораторной работе №5
по дисциплине «Паралельнные алгоритмы»
Тема: Знакомство с программированием гетерогенных систем в стандарте Open CL

Студент гр. 0304 _____ Максименко Е.М.

Преподаватель _____ Сергеева Е.И

Санкт-Петербург
2023

Цель работы.

Изучить стандарт OpenCL. Реализовать построение визуализации множества Мандельброта с использованием OpenCL

Задание.

Реализовать расчёт фрактала Мандельброта на OpenCL. Визуализировать результат.

В отчёте: Произвести оценку производительности.

Выполнение работы.

1. Построение множества Мандельброта производилось с использованием алгоритма escape-time. Доказано, что все множество Мандельброта может быть расположено на плоскости в круге с радиусом 2. Суть алгоритма в том, чтобы проверить, что точка за определенное число итераций не покинет данный круг. Если точка не покинула круг, то она принадлежит множеству и окрашивается в черный. Если точка покинула множество, то, в зависимости от количества итераций, она окрашивается в другой цвет.

Для визуализации множества Мандельброта была написана программа с использованием фреймворка Qt5. Программа строит множество Мандельброта с использованием CPU последовательно или с использованием GPU/CPU параллельно (OpenCL). Интерфейс программы см. на рис. 1.

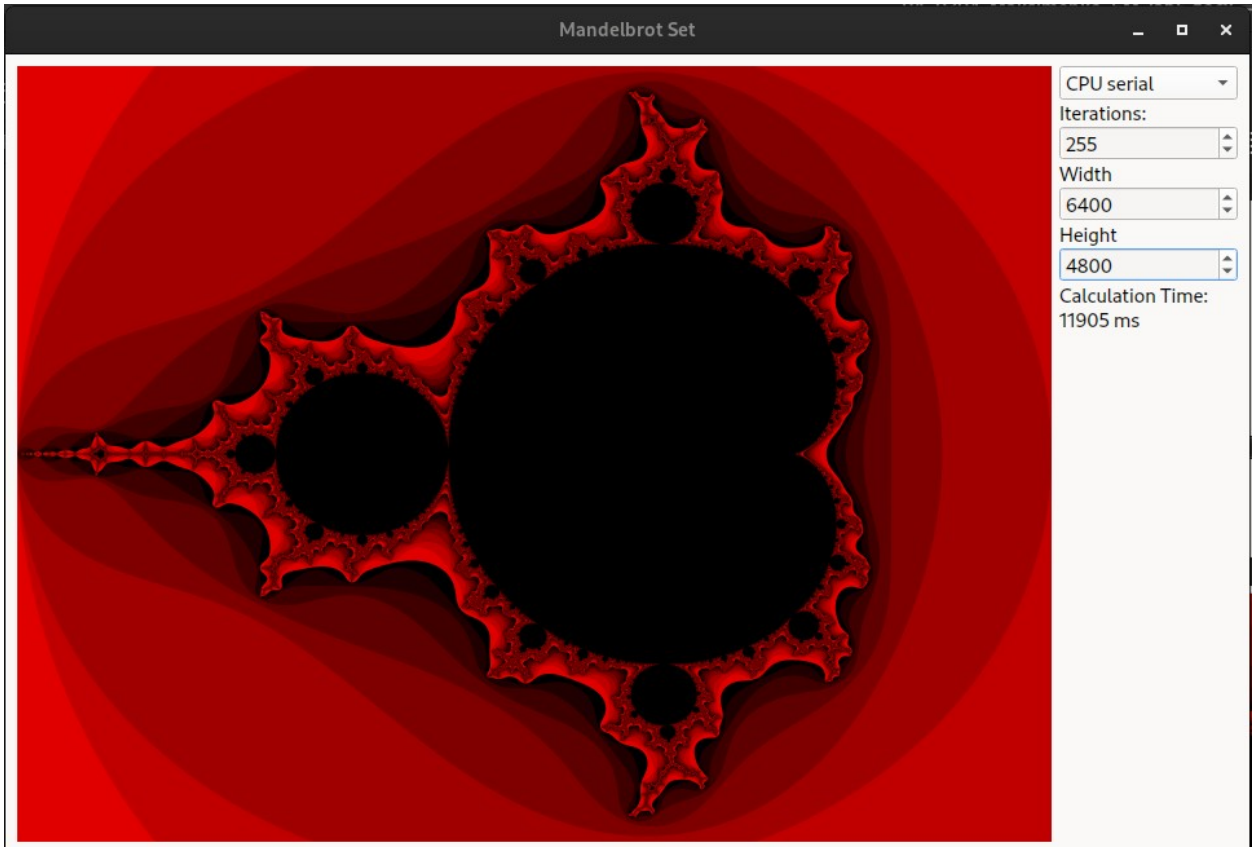

Рисунок 1. Визуализация множества Мандельброта. Интерфейс программы

2. Для построения множества Мандельброта с использованием OpenCL был написан класс MandelbrotCl, который занимается подготовкой OpenCL и запуском вычислений. В классе определяются девайсы, на которых будут производиться вычисления, создается контекст, буфер для передачи расчетов из программы OpenCL. Для осуществления вычислений было написано ядро OpenCL, которое производит вычисления количества итераций для одной точки множества.

3. Было проведено сравнение производительности расчетов с использованием CPU (последовательно) и с использованием GPU (параллельно, OpenCL). Множество рассматривалось размера 800x600 точек.

Измерялось время непосредственно расчетов с учетом подготовки к выполнению программы (для OpenCL создание контекста, создание буфера, передача данных в программу OpenCL) в зависимости от числа итераций. Результаты замеров см. в табл. 1.

Таблица 1. Замеры производительности построения множества Мандельброта в зависимости от количества итераций.

Количество итераций	**Время построения множества последовательно, мс**	**Время построения множества параллельно (OpenCL), мс**
255	187	89
511	348	67
1023	680	69
2047	1326	76
4095	2618	80
10000	6342	100
20000	12666	129
30000	18947	202

Как видно по табл. 1, время построения множества практически линейно зависит от количества итераций. Также видно, что время построения множества с использованием OpenCL практически константно для небольшого количества итераций (порядка 10^3). При большом количестве итераций (порядка 10^5) видно, что рост времени вычислений с использованием OpenCL также линейный. Однако время выполнения программы с использованием OpenCL сильно меньше времени выполнения программы, выполняющей расчеты последовательно. Это связано с тем, что в программе, использующей OpenCL, вычисления производятся параллельно для всех точек множества.

Зафиксируем количество итераций на 255. Зависимость времени построения множества от размера множества см. в табл. 2.

Таблица 2. Замеры производительности построения множества Мандельброта в зависимости от размера множества.

Размер множества	**Время построения множества последовательно, мс**	**Время построения множества параллельно (OpenCL), мс**
800x600	189	69
1600x1200	754	83
3200x2400	3024	105
6400x4800	11905	99

Как видно по табл. 2, зависимости времени выполнения последовательной программы от размера множества линейная. При этом зависимость времени выполнения параллельной программы от размера множества отсутствует. Это связано с тем, что вычисления производятся параллельно для всех точек множества.

Выводы.

Были изучены принципы построения программ с использованием OpenCL. Была разработана программа для визуализации множества Мандельброта с использованием последовательных вычислений и параллельных вычислений (с использованием OpenCL).

Было проведено сравнение последовательной и параллельной реализации. Время выполнения параллельной программы сильно меньше времени выполнения последовательной программы. Это связано с тем, что вычисления в OpenCL производятся параллельно для всех точек множества.