

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

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Параллельные алгоритмы»
Тема: Знакомство с программированием гетерогенных систем в
стандарте OpenCL

Студент гр. 9304

Преподаватель

Борисовский В.Ю.

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы

Ознакомиться с программированием гетерогенных систем в стандарте OpenCL.

Задание

Реализовать расчёт фрактала Мандельброта на OpenCL. Визуализировать полученный результат. Произвести оценку производительности.

Выполнение работы

Была реализована программа в стандарте OpenCL. При использовании GPU происходит распараллеливание вычисления всех пикселей. Каждый пиксель изображения считается с помощью work item. Цвет пикселя определялся количеством итераций, необходимых для его вычисления.

После вычисления значений каждого пикселя, производилось преобразование полученного массива данных в изображение с расширением .ppm (Portable Bitmap File Format).

Пример полученного изображения фрактала Мандельброта представлен на рисунке 1.

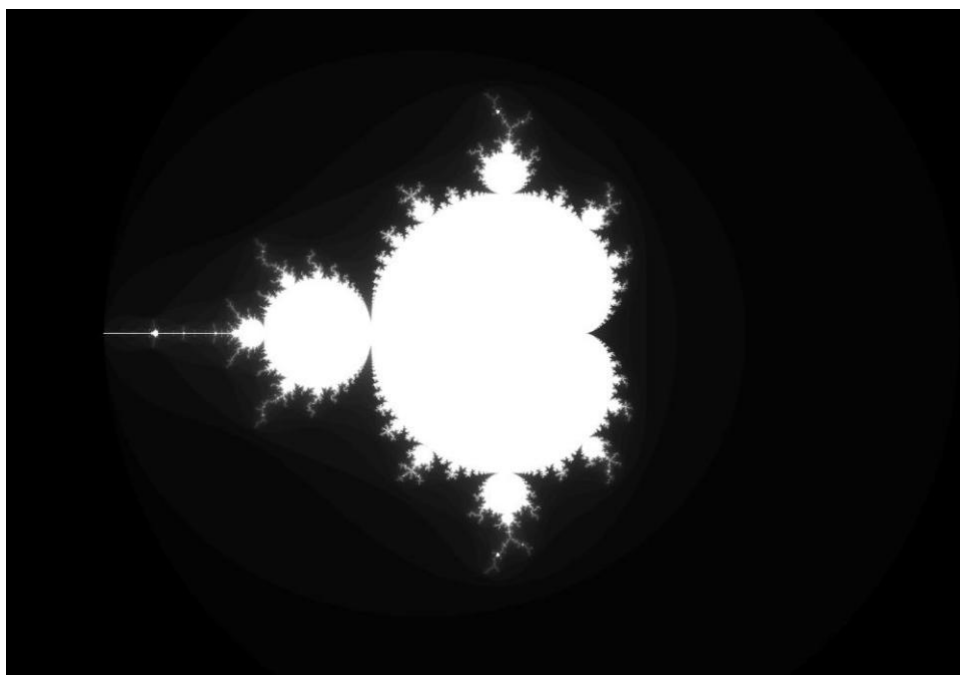


Рисунок 1 – Пример полученного изображения фрактала Мандельброта

Оценка производительности

Сравнение производительности между реализаций последовательного вычисления (на CPU) и параллельного вычисления (с помощью OpenCL на GPU) фрактала Мандельброта при максимальном ограничении итераций в 256 представлено в таблице 1. По таблице можно сделать вывод, что вычисление фрактала Мандельброта с помощью OpenCL оказывается значительно быстрее.

Таблица 1 – Сравнение последовательного и параллельного вычисления фрактала Мандельброта

Размер изображения	Затраченное время, мк. сек.	
	CPU	GPU
1280x720	90361	2804
1920x1080	181649	3446
2560x1440	318324	4806
3840x2160	738119	8292
7680x4320	2893316	63880

Выводы

В ходе выполнения лабораторной работы был реализована программа, вычисляющая фрактала Мандельброта в стандарте OpenCL. Было получено, что вычисление фрактала с помощью OpenCL оказывается значительно быстрее.

