

**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**  
**Open CL**

Студентка гр. 9303

Москаленко Е.М.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

### **Цель работы.**

Ознакомиться со стандартом для кроссплатформенного параллельного программирования на графических и центральных процессорах OpenCL на примере реализации расчета фрактала Мандельброта.

### **Задание.**

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

### **Выполнение работы.**

Множество Мандельброта — это «множество таких точек  $C$  на комплексной плоскости, для которых рекуррентное соотношение  $Z_{n+1} = Z_n^2 + C$  при  $Z_0 = 0$  задаёт ограниченную последовательность.

Иными словами, на каждом шаге число возводится в квадрат и к нему добавляется константа. Все множество расположено на плоскости внутри круга радиуса 2. Если за определенное число шагов  $N$  (итераций) последовательность осталась в заданных пределах, то значение константы принадлежит множеству и точка окрашивается в черный цвет. Соответственно, если на каком-то этапе, меньшем  $N$  - общего количества итераций, элемент последовательности по модулю стал больше 2, то точка множеству не принадлежит и окрашивается в другой цвет, который вычисляется в функции *get\_color()*.

Было написано 2 алгоритма - последовательный однопоточный и параллельный с помощью OpenCL.

Для параллельного вычисления была написана Kernel функция *mandelbrot*, на вход которой подается количество итераций алгоритма и объект класса *image2d\_t* - изображение для записи результата. Индекс обрабатываемого пикселя берется с помощью вызова *get\_global\_id(0)*, *get\_global\_id(1)*.

Программа-хост контролирует запуск и выполнение ядра на вычислительном устройстве. Запуск команд OpenCL происходит с помощью вызова функции `CL_RUN_COMMAND`, которая проверяет код и в случае неудачи выводит сообщение об ошибке.

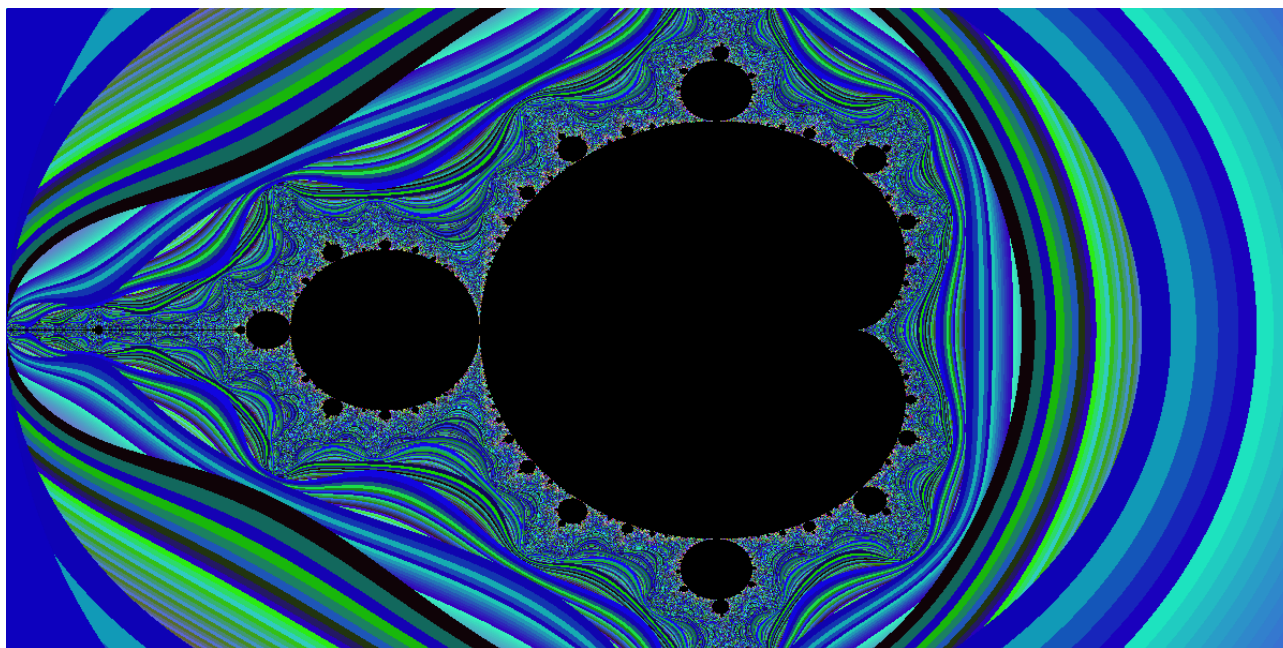


Рисунок 1 - Результат последовательного алгоритма

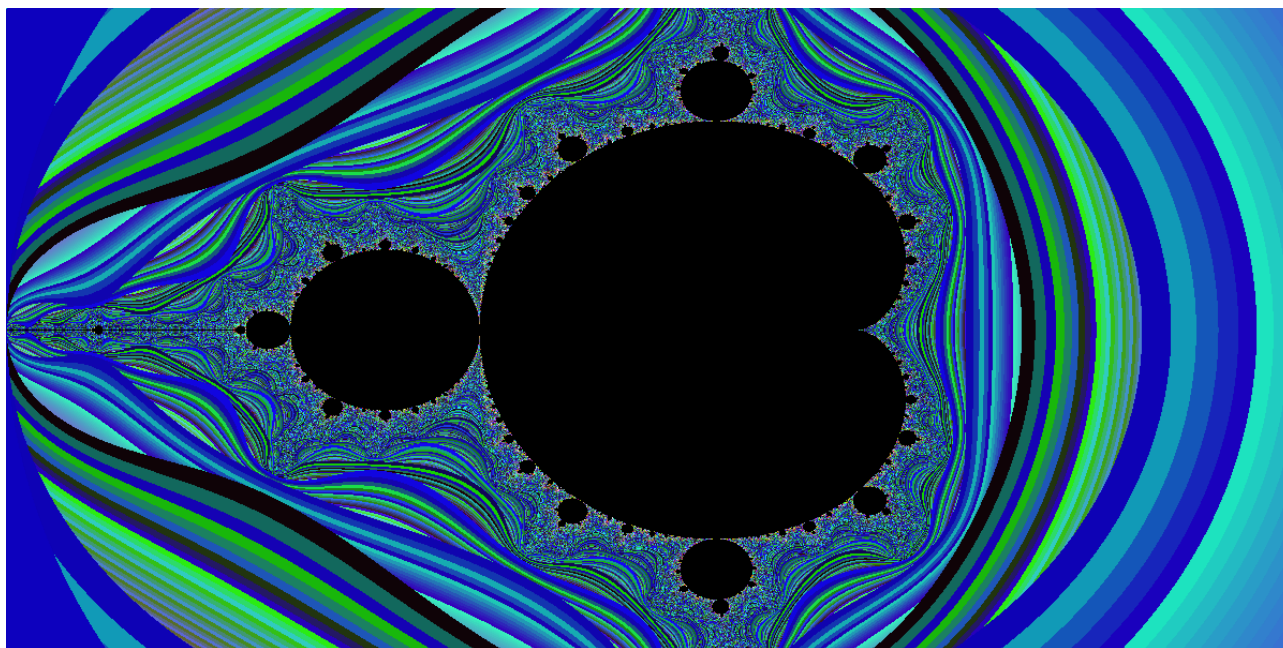


Рисунок 2 - Результат алгоритма OpenCL

## Оценка производительности

Размер изображения	Количество итераций алгоритма	Время OpenCL, мс	Время однопоточный, мс
600*1200	128	70	806
600*1200	1024	405	1392
800*1600	1024	690	2465
800*1600	2048	1328	4958
800*1600	256	165	906

Таблица 1 - Оценка производительности

Как видно из таблицы, параллельный алгоритм работает намного быстрее.

### Выводы.

Были изучены основы работы со стандартом для кроссплатформенного параллельного программирования на графических и центральных процессорах OpenCL на примере реализации расчета фрактала Мандельброта. Оценка производительности показали, что программы, написанные с помощью OpenCL работают быстрее, чем однопоточные реализации, вне зависимости от размера изображения и количества итераций алгоритма.