

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)**

**Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»**

**Лабораторная работа №1
по курсу «Программирование графических процессоров»**

**Освоение программного обеспечения для работы с технологией
CUDA.**

Примитивные операции над векторами.

Выполнил: К.М. Воронов

Группа: 8О-407Б

Преподаватель: А.Ю. Морозов

Москва, 2022

Условие

Цель работы: ознакомление и установка программного обеспечения для работы с программно-аппаратной архитектурой параллельных вычислений (CUDA), реализация одной из примитивных операций над векторами.

Вариант 5. Поэлементное нахождение максимума векторов.

Программное и аппаратное обеспечение

GPU:

- Название NVIDIA GeForce GTX 1050
- Compute capability: 6.1
- Графическая память: 4236378112
- Разделяемая память: 49152
- Константная память: 65536
- Количество регистров на блок: 65536
- Максимальное количество нитей: (1024, 1024, 64)
- Максимальное количество блоков: (2147483647, 65535, 65535)
- Количество мультипроцессоров: 5

Сведения о системе:

- Процессор: Intel Core i5-8300H 2.30GHz
- ОЗУ: 32 ГБ
- SSD 1ТБ
- HDD 1ТБ

Программное обеспечение:

- OS: Kubuntu 20.04
- Текстовый редактор: Sublime text
- Компилятор: nvcc

Метод решения

Для решения этой задачи я создаю массив в котором подряд храню два исходных вектора. Далее в каждом потоке я сравниваю i элемент этого массива с $i + n$ элементом, где n — длина векторов, и записываю в первую часть максимум.

Описание программы

Все элементы векторов имеют тип данных `double`. В функции ядра `kernel` происходит непосредственно нахождение максимума, в качестве индекса используется `id` потока. Для того, чтобы обрабатывать все элементы в случае, если их больше, чем потоков, используется `offset`.

Результаты

CPU:

n	Время, мс
100	0.003000 ms
1000	0.011000 ms
1000000	9.780000 ms
10000000	131.533000 ms

GPU:

<1, 32>

n	Время, мс
100	0.017056 ms
1000	0.022528 ms
1000000	10.768640 ms
10000000	107.665314 ms

<128, 128>

n	Время, мс
100	0.014176 ms
1000	0.016992 ms
1000000	0.256512 ms
10000000	2.452864 ms

<512, 512>

n	Время, мс
100	0.022752 ms
1000	0.022400 ms
1000000	0.249600 ms
10000000	2.427648 ms

<1024, 1024>

n	Время, мс
100	0.041056 ms
1000	0.041184 ms
1000000	0.253888 ms
10000000	2.418176 ms

Выводы

Выполнив данную лабораторную, я получил основы работы с технологией CUDA, а также реализовал простой алгоритм. Из сложностей столкнулся с установкой CUDA и драйверов на видеокарту, приходилось несколько раз все сносить и устанавливать заново.