

**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: А. Е. Максимов
Группа: М8О-407Б-19
Преподаватель: А. Ю. Морозов**

Москва, 2023

1 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 6. Поэлементное возведение в квадрат вектора.

Требуется реализовать поэлементное возведение в квадрат вектора. Ограничения:

- $n < 2^{25}$
- все результаты выводить с относительной точностью 10^{-10}

2 Программное и аппаратное обеспечение

Характеристики графического процессора:

- Compute capability: 7.5
- Наименование: NVIDIA GeForce GTX 1660 Ti
- Графическая память: 6143mb
- Константная память: 64kb
- Разделяемая память: 48kb
- Количество регистров на блок: 65536
- Максимальное количество потоков на блок: (1024, 1024, 64)
- Максимальное количество блоков: (2147483647, 65535, 65535)
- Количество мультипроцессоров: 6

Характеристики системы:

- Процессор: «Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz»
- Память: 16,0 ГБ встроенной памяти ноутбука
- SSD: «Hynix BC511 NVMe SK hynix 512GB »

Программное обеспечение:

- ОС: «Windows 10 Домашняя x64»
- Текстовый редактор: «Notepad++»
- Компилятор: «nvcc: Cuda compilation tools, release 12.0, V12.0.76»

3 Метод решения

Для решения этой задачи использовались записи лекций по предмету и официальный путеводитель от Nvidia.

Сам алгоритм не представляет интереса, поскольку является базовым обучающим примером.

4 Результаты

Замеры времени работы CPU и ядер с различными конфигурациями.

Конфигурация	$n = 10$, мс	$n = 100$, мс	$n = 10^4$, мс	$n = 10^6$, мс	$n = 3.3 * 10^7$, мс
CPU	0.0	0.0	0.0	1.995	98.032
(1, 32)	0.004	0.004	0.065	10.217	314.873
(32, 32)	0.005	0.005	0.006	0.352	47.8
(1024, 32)	0.006	0.006	0.006	0.067	43.541
(1, 256)	0.005	0.005	0.013	1.347	84.875
(64, 256)	0.004	0.005	0.006	0.067	2.084
(256, 256)	0.005	0.006	0.006	0.067	28.750
(1, 1024)	0.005	0.005	0.009	0.429	14.073
(16, 1024)	0.006	0.013	0.005	0.067	44.875
(64, 1024)	0.004	0.005	0.018	0.065	46.327

5 Выводы

Использование технологий CUDA открывает возможность более оптимальных и быстрых вычислений благодаря высокой пропускной способности графических мультипроцессоров.

Сложность алгоритма $O(n)$.

Список литературы

[1] *CUDA C++ Programming Guide*

URL: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#>