

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Факультет прикладной математики и физики
Кафедра вычислительной математики и программирования

Лабораторная работа №1
по курсу «Программирование графических процессоров»

**Освоение программного обеспечения
для работы с технологией CUDA.
Примитивные операции над векторами.**

Выполнил: Н.И. Забарин
Группа: 8О-408Б
Преподаватели: К.Г. Крашенинников,
А.Ю. Морозов

Москва, 2017

Условие

1. Цель работы:

Ознакомление и установка программного обеспечения для работы с программно- аппаратной архитектурой параллельных вычислений(CUDA).
Реализация одной из примитивных операций над векторами.

2. Вариант задания:

Вариант 6. Поэлементное возведение в квадрат вектора.

Программное и аппаратное обеспечение

Спецификации GPU

Name:	GeForce GT 620M
Compute capability:	2.1
Warp size:	32
Max threads per block:	1024
Clock rate:	1250000
Multiprocessor count:	2
Max threads dim:	1024 1024 64
Max grid size:	65535 65535 65535

Спецификации видеопамяти

Total global memory:	1024 MB
Shared memory per block:	48 KB
Registers per block:	32 KB
Total constant memory:	64 KB

Спецификации CPU

Процессор	Intel Core i5-3317U
Ядер	4
Базовая частота	1.7 GHz

Спецификация оперативной памяти

Объем памяти	10 Гб
Частота	1600 МГц

Спецификация жесткого диска

Тип	SSD
Интерфейс	M.2
Объем	240Gb

Спецификация программного обеспечения

CUDA Toolkit	7.5
OS	Ubuntu 16.10
IDE	Vim
Compiler	nvcc V7.5.17

Метод решения

Для параллельного вычисления «квадрата» вектора, вычисление каждого элемента итогового вектора производится в отдельном потоке. Если потоков не хватает, каждый поток проходит по элементам с шагом, равным количеству потоков.

Описание программы

```
__global__ void kernel(double *darr, int n) {
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    int offset = gridDim.x * blockDim.x;

    while (idx < n) {
        darr[idx] *= darr[idx];
        idx += offset;
    }
}
```

Результаты

Размер векторов	Параметры ядра	Время
100	<<<16, 16>>>	0.032
100	<<<16, 16>>>	0.025
100	<<<16, 16>>>	0.007
		Среднее значение: 0.021
100	<<<256, 256>>>	0.044
1000000	<<<16, 16>>>	2.505
1000000	<<<16, 16>>>	2.497
1000000	<<<16, 16>>>	2.509
		Среднее значение: 2.503
1000000	<<<256, 256>>>	1.242
1000000	<<<256, 256>>>	1.240
1000000	<<<256, 256>>>	1.241
		Среднее значение: 1.241

Выводы

Данная лабораторная работа показывает принцип SIMD (Single Instruction Multiple Data) – подход, при котором группа параллельно работающих процессоров осуществляют действия над разными данными, но при этом все они в произвольный момент времени должны выполнять одинаковую команду.