МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»

Лабораторная работа №1
по курсу «Разработка параллельных программ и тематических решателей при проектировании сложных технических систем»

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

Выполнил: П.А. Королев
Группа: М8О-107М-21
Преподаватель: А.Ю. Морозов

Москва, 2022

Условие

1. Ознакомление и установка программного обеспечения для работы с программно-аппаратной архитектурой параллельных вычислений(CUDA). Реализация одной из примитивных операций над векторами.
2. Вариант 3. Поэлементное умножение векторов.

Программное и аппаратное обеспечение

Используется компилятор nvcc версии 7.0(g++ версии 4.8.4) на 64-х битной Ubuntu 14.04 LTS.

Параметры графического процессора:

- Compute capability : 6.1
- Name : GeForce GTX 1050
- Total Global Memory : 2096103424
- Shared memory per block : 49152
- Registers per block : 65536
- Max threads per block : (1024, 1024, 64)
- Max block : (2147483647, 65535, 65535)
- Total constant memory : 65536
- Multiprocessors count : 5

Метод решения

Для поэлементного умножения векторов использовалась циклическая схема распределения данных по потокам, когда i-ый элемент обрабатывается i-ым потоком, а в случае когда потоков меньше чем элементов, то i-ый элемент будет обработан потоком с номером i+offset, где offset - это шаг равный общему количеству потоков.

Описание программы

Разделение по файлам, описание основных типов данных и функций. Обязательно описать реализованные ядра.

Вся программа написана в файле lab1.cu. На каждое умножение выделяется по одному потоку.

```
__global__ void kernel(double* first_vector, double* second_vector, double*
result_vector, double vector_size) {
    int idx = blockIdx.x * blockDim.x + threadIdx.x;
    int offset = blockDim.x * gridDim.x;
    while (idx < (int)vector_size) {
        result_vector[idx] = first_vector[idx] * second_vector[idx];
        idx += offset;
    }
}
```

Результаты

На вход подается два вектора размерностью 1 000 000.
Работа ядра с различными конфигурациями:

Конфигурация	Время работы
<<< 1, 32 >>>	20.149183 ms
<<< 2, 64 >>>	5.115456 ms
<<< 256, 256 >>>	0.704480 ms
<<< 512, 512 >>>	0.701728 ms

Время работы на GPU <<<256, 256>>>	Время работы на CPU
0.704480 ms	9312 ms

Выводы

В результате выполнения лабораторной работы было установлено программное обеспечение для работы с программно-аппаратной архитектурой параллельных вычислений(CUDA) и реализован параллельный алгоритм поэлементного умножения двух векторов с использованием технологии CUDA. После измерения времени выполнения программы на CPU и GPU было выявлено, что реализация с использованием GPU намного эффективнее реализации на CPU.