

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

**Институт №8 «Информационные технологии и прикладная математика»
Кафедра 810 «Информационные технологии в моделировании и
управлении»**

**Лабораторная работа №1
по курсу «Основы Python, Java и Scala, платформы CUDA для анализа
данных»**

Изучение технологии CUDA

Выполнил: А.С.Бобряков
Группа: М8О-103М-19
Преподаватель: А.Ю. Морозов

Москва, 2020

Условие

Цель работы – установить программное обеспечение для запуска CUDA, разобраться в основе синтаксиса и настроек.

Вариант задания 1. Сложение векторов.

Программное и аппаратное обеспечение

Видеокарта: NVIDIA GeForce GTX 1060 3Gb

Компоненты	Подробности
GeForce GTX 1060 3GB	Версия драйвера: 441.22 Тип драйвера: Standard Версия API Direct3D: 12 Уровень возможносте... 12_1 Ядра CUDA: 1152 Тактовая частота гра... 1594 МГц Скорость передачи д... 8.01 Гбит/с Интерфейс памяти: 192 бит Пропускная способнос... 192.19 ГБ/с Доступная графическ... 11237 МБ Выделенная видеопам... 3072 МБ GDDR5 Системная видеопамя... 0 МБ Разделяемая системна... 8165 МБ Версия BIOS видео: 86.06.3C.00.7D IRQ: Not used

Процессор: Intel® Core™ i7-8700K CPU @ 3.70GHz

Другое: ОС Windows, IDE – Clion EAP,

Метод решения

Задача установки программного обеспечения для CUDA выполнена на ОС Windows под IDE CLion EAP (2020г) путем чтения документации на сайте компании JetBrains.

Задача сложения двух векторов выполнена путем стандартного сложения двух векторов из курса алгебры 8 класса по формуле сложения двух векторов.

Описание программы

В программе использован макрос для CUDA на листинге 1.

```
#define CSC(call) do { \
    cudaError_t res = call; \
    if (res != cudaSuccess) { \
        fprintf(stderr, "CUDA Error in %s:%d: %s", __FILE__, \
            __LINE__, cudaGetErrorString(res)); \
        exit(0); \
    } \
} while (0)
```

Листинг 1 – Макрос для CUDA для отлова ошибок.

Программа требует на вход размер векторов и сами вектора, после чего запускает ядро с логикой сложения полученных векторов.

Результаты

Время работы ядра в зависимости от конфигурации представлены в Таблице 1.

Таблица 1. Время выполнения ядра программы в зависимости от конфигурации.

Число потоков \ Число блоков	32	128	512	1024
32	3.809600	1.619968	1.524544	1.524192
128	1.628160	1.526592	1.527168	1.861088
512	1.540608	1.538976	1.524704	1.556896
1024	1.552768	1.562336	1.531904	1.548288

На CPU время выполнения 0.020000.

Выводы

Область применения реализованного алгоритма – учебное изучение основ CUDA. Типовые задачи, решаемые этим методом – алгебраические действия с матрицами. Возникшие проблемы – практическая невозможность установить платформу CUDA на нормальную IDE именно под ОС Windows.

Результат работы на CPU значительно лучше, чем результаты с различной конфигурацией по GPU на тестируемой машине.