МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии 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. О. Тояков
Группа: М8О-407Б-18
Преподаватели: К. Г. Крашенинников,
А. Ю. Морозов

Москва, 2021

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 4. Поэлементное вычисление модуля вектора.

ПРОГРАММНОЕ И АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

Device: GeForce MX250

Размер глобальной памяти: 3150381056

Размер константной памяти : 65536

Размер разделяемой памяти: 49152

Регистров на блок: 32768

Максимум потоков на блок: 1024

Количество мультипроцессоров : 3

OS: Linux Ubuntu 18.04

Редактор: VSCode

Компилятор: nvcc версии 11.4 (g++ версии 7.5.0)

МЕТОД РЕШЕНИЯ

Для нахождения модуля вектора необходимо сначала записать данные на CPU. Затем передать их GPU. Функцией kernel мы распараллелим обработку данных (применение функции abs поэлементно). Потоки будут обрабатывать по несколько элементов сразу, используя переменную offset, как шаг. Затем нужно передать готовый массив обратно хосту и вывести результат.

ОПИСАНИЕ ПРОГРАММЫ

Макрос **CSC** отвечает за отслеживание ошибок в функциях cuda, поэтому все cuda-вызовы оборачиваются в него и при cudaError_t != cudaSuccess выводится сообщение об ошибке.

__global__ void kernel(int* arr, n) – функция на GPU, в которой происходит обработка массива по потокам.

int main() – отвечает за ввод, передачу данных в kernel и вывод.

РЕЗУЛЬТАТЫ

В обоих программах числа для массивов генерировались рандомно в промежутке [-1000, 1000].

Работа на GPU:

Тест:	Результат:
3 -1 2 -3	kernel = «<1, 32»>, time = 0.013312 kernel = «<1, 64»>, time = 0.003072 kernel = «<1, 128»>, time = 0.003072 kernel = «<1, 256»>, time = 0.003072 kernel = «<1, 512»>, time = 0.003072 kernel = «<1, 1024»>, time = 0.003072 kernel = «<2, 32»>, time = 0.004096 kernel = «<2, 64»>, time = 0.003072 kernel = «<2, 128»>, time = 0.004096 kernel = «<2, 256»>, time = 0.003072 kernel = «<2, 512»>, time = 0.003072 kernel = «<2, 1024»>, time = 0.003072 kernel = «<4, 32»>, time = 0.004096 kernel = «<4, 64»>, time = 0.003072 kernel = «<4, 128»>, time = 0.003072 kernel = «<4, 256»>, time = 0.003072 kernel = «<4, 512»>, time = 0.003072 kernel = «<4, 1024»>, time = 0.004096 kernel = «<8, 32»>, time = 0.004096 kernel = «<8, 64»>, time = 0.003072 kernel = «<8, 128»>, time = 0.004096 kernel = «<8, 256»>, time = 0.004096 kernel = «<8, 512»>, time = 0.004064 kernel = «<8, 1024»>, time = 0.004096 kernel = «<16, 32»>, time = 0.004096 kernel = «<16, 64»>, time = 0.004096 kernel = «<16, 128»>, time = 0.004096 kernel = «<16, 256»>, time = 0.004096 kernel = «<16, 512»>, time = 0.003072 kernel = «<16, 1024»>, time = 0.003072 kernel = «<32, 32»>, time = 0.015360

	kernel = «<32, 64»>, time = 0.003072 kernel = «<32, 128»>, time = 0.003072 kernel = «<32, 256»>, time = 0.004096 kernel = «<32, 512»>, time = 0.004096 kernel = «<32, 1024»>, time = 0.004096 kernel = «<64, 32»>, time = 0.004096 kernel = «<64, 64»>, time = 0.003072 kernel = «<64, 128»>, time = 0.004096 kernel = «<64, 256»>, time = 0.004096 kernel = «<64, 512»>, time = 0.004096 kernel = «<64, 1024»>, time = 0.005120 kernel = «<128, 32»>, time = 0.003072 kernel = «<128, 64»>, time = 0.004096 kernel = «<128, 128»>, time = 0.003072 kernel = «<128, 256»>, time = 0.004096 kernel = «<128, 512»>, time = 0.006144 kernel = «<128, 1024»>, time = 0.008192 kernel = «<256, 32»>, time = 0.005120 kernel = «<256, 64»>, time = 0.005120 kernel = «<256, 128»>, time = 0.005120 kernel = «<256, 256»>, time = 0.005120 kernel = «<256, 512»>, time = 0.008192 kernel = «<256, 1024»>, time = 0.014336 kernel = «<512, 32»>, time = 0.006144 kernel = «<512, 64»>, time = 0.006144 kernel = «<512, 128»>, time = 0.006144 kernel = «<512, 256»>, time = 0.008192 kernel = «<512, 512»>, time = 0.012288 kernel = «<512, 1024»>, time = 0.023552 kernel = «<1024, 32»>, time = 0.009216 kernel = «<1024, 64»>, time = 0.010240 kernel = «<1024, 128»>, time = 0.009216 kernel = «<1024, 256»>, time = 0.011264 kernel = «<1024, 512»>, time = 0.021504 kernel = «<1024, 1024»>, time = 0.044032
100	kernel = «<1, 32»>, time = 0.012288 kernel = «<1, 64»>, time = 0.004096 kernel = «<1, 128»>, time = 0.003072 kernel = «<1, 256»>, time = 0.003072 kernel = «<1, 512»>, time = 0.003072 kernel = «<1, 1024»>, time = 0.009216 kernel = «<2, 32»>, time = 0.005120 kernel = «<2, 64»>, time = 0.004000 kernel = «<2, 128»>, time = 0.003072 kernel = «<2, 256»>, time = 0.003072 kernel = «<2, 512»>, time = 0.003072 kernel = «<2, 1024»>, time = 0.003072 kernel = «<4, 32»>, time = 0.004096 kernel = «<4, 64»>, time = 0.004096 kernel = «<4, 128»>, time = 0.003072 kernel = «<4, 256»>, time = 0.003072 kernel = «<4, 512»>, time = 0.003072 kernel = «<4, 1024»>, time = 0.003072 kernel = «<8, 32»>, time = 0.004064 kernel = «<8, 64»>, time = 0.004096 kernel = «<8, 128»>, time = 0.005120 kernel = «<8, 256»>, time = 0.004096 kernel = «<8, 512»>, time = 0.004096

	kernel = «<8, 1024»>, time = 0.003072 kernel = «<16, 32»>, time = 0.004096 kernel = «<16, 64»>, time = 0.004096 kernel = «<16, 128»>, time = 0.005120 kernel = «<16, 256»>, time = 0.004096 kernel = «<16, 512»>, time = 0.004096 kernel = «<16, 1024»>, time = 0.004096 kernel = «<32, 32»>, time = 0.003072 kernel = «<32, 64»>, time = 0.003072 kernel = «<32, 128»>, time = 0.004064 kernel = «<32, 256»>, time = 0.005056 kernel = «<32, 512»>, time = 0.005088 kernel = «<32, 1024»>, time = 0.005120 kernel = «<64, 32»>, time = 0.004096 kernel = «<64, 64»>, time = 0.004096 kernel = «<64, 128»>, time = 0.004096 kernel = «<64, 256»>, time = 0.004096 kernel = «<64, 512»>, time = 0.007072 kernel = «<64, 1024»>, time = 0.006144 kernel = «<128, 32»>, time = 0.005120 kernel = «<128, 64»>, time = 0.004096 kernel = «<128, 128»>, time = 0.004096 kernel = «<128, 256»>, time = 0.005120 kernel = «<128, 512»>, time = 0.006144 kernel = «<128, 1024»>, time = 0.008192 kernel = «<256, 32»>, time = 0.006144 kernel = «<256, 64»>, time = 0.006016 kernel = «<256, 128»>, time = 0.005120 kernel = «<256, 256»>, time = 0.006144 kernel = «<256, 512»>, time = 0.008192 kernel = «<256, 1024»>, time = 0.016320 kernel = «<512, 32»>, time = 0.006144 kernel = «<512, 64»>, time = 0.008064 kernel = «<512, 128»>, time = 0.006144 kernel = «<512, 256»>, time = 0.007168 kernel = «<512, 512»>, time = 0.012288 kernel = «<512, 1024»>, time = 0.023552 kernel = «<1024, 32»>, time = 0.009216 kernel = «<1024, 64»>, time = 0.011104 kernel = «<1024, 128»>, time = 0.010240 kernel = «<1024, 256»>, time = 0.012288 kernel = «<1024, 512»>, time = 0.021504 kernel = «<1024, 1024»>, time = 0.041984
10000	kernel = «<1, 32»>, time = 0.070080 kernel = «<1, 64»>, time = 0.065504 kernel = «<1, 128»>, time = 0.039872 kernel = «<1, 256»>, time = 0.030720 kernel = «<1, 512»>, time = 0.029696 kernel = «<1, 1024»>, time = 0.029696 kernel = «<2, 32»>, time = 0.056320 kernel = «<2, 64»>, time = 0.035840 kernel = «<2, 128»>, time = 0.026624 kernel = «<2, 256»>, time = 0.049152 kernel = «<2, 512»>, time = 0.030720 kernel = «<2, 1024»>, time = 0.026624 kernel = «<4, 32»>, time = 0.037888 kernel = «<4, 64»>, time = 0.027648 kernel = «<4, 128»>, time = 0.033824 kernel = «<4, 256»>, time = 0.029696

	kernel = «<4, 512»>, time = 0.034816 kernel = «<4, 1024»>, time = 0.035840 kernel = «<8, 32»>, time = 0.029696 kernel = «<8, 64»>, time = 0.024576 kernel = «<8, 128»>, time = 0.024576 kernel = «<8, 256»>, time = 0.025536 kernel = «<8, 512»>, time = 0.024608 kernel = «<8, 1024»>, time = 0.026624 kernel = «<16, 32»>, time = 0.023552 kernel = «<16, 64»>, time = 0.023552 kernel = «<16, 128»>, time = 0.025536 kernel = «<16, 256»>, time = 0.026624 kernel = «<16, 512»>, time = 0.024544 kernel = «<16, 1024»>, time = 0.025600 kernel = «<32, 32»>, time = 0.025600 kernel = «<32, 64»>, time = 0.025600 kernel = «<32, 128»>, time = 0.024672 kernel = «<32, 256»>, time = 0.024576 kernel = «<32, 512»>, time = 0.025664 kernel = «<32, 1024»>, time = 0.025568 kernel = «<64, 32»>, time = 0.025632 kernel = «<64, 64»>, time = 0.024576 kernel = «<64, 128»>, time = 0.024576 kernel = «<64, 256»>, time = 0.023552 kernel = «<64, 512»>, time = 0.023552 kernel = «<64, 1024»>, time = 0.023552 kernel = «<128, 32»>, time = 0.023552 kernel = «<128, 64»>, time = 0.030720 kernel = «<128, 128»>, time = 0.025600 kernel = «<128, 256»>, time = 0.028672 kernel = «<128, 512»>, time = 0.029696 kernel = «<128, 1024»>, time = 0.025600 kernel = «<256, 32»>, time = 0.023552 kernel = «<256, 64»>, time = 0.023552 kernel = «<256, 128»>, time = 0.022528 kernel = «<256, 256»>, time = 0.024576 kernel = «<256, 512»>, time = 0.024576 kernel = «<256, 1024»>, time = 0.030720 kernel = «<512, 32»>, time = 0.025600 kernel = «<512, 64»>, time = 0.025600 kernel = «<512, 128»>, time = 0.025696 kernel = «<512, 256»>, time = 0.025600 kernel = «<512, 512»>, time = 0.031744 kernel = «<512, 1024»>, time = 0.039968 kernel = «<1024, 32»>, time = 0.027648 kernel = «<1024, 64»>, time = 0.027648 kernel = «<1024, 128»>, time = 0.026624 kernel = «<1024, 256»>, time = 0.030720 kernel = «<1024, 512»>, time = 0.038912 kernel = «<1024, 1024»>, time = 0.062464
100000	kernel = «<1, 32»>, time = 0.856608 kernel = «<1, 64»>, time = 0.436224 kernel = «<1, 128»>, time = 0.223232 kernel = «<1, 256»>, time = 0.116736 kernel = «<1, 512»>, time = 0.069632 kernel = «<1, 1024»>, time = 0.045056 kernel = «<2, 32»>, time = 0.436224 kernel = «<2, 64»>, time = 0.225280 kernel = «<2, 128»>, time = 0.117760

	kernel = «<2, 256»>, time = 0.069440 kernel = «<2, 512»>, time = 0.045056 kernel = «<2, 1024»>, time = 0.037760 kernel = «<4, 32»>, time = 0.227200 kernel = «<4, 64»>, time = 0.118784 kernel = «<4, 128»>, time = 0.068608 kernel = «<4, 256»>, time = 0.044032 kernel = «<4, 512»>, time = 0.036864 kernel = «<4, 1024»>, time = 0.036864 kernel = «<8, 32»>, time = 0.120672 kernel = «<8, 64»>, time = 0.068608 kernel = «<8, 128»>, time = 0.043008 kernel = «<8, 256»>, time = 0.035840 kernel = «<8, 512»>, time = 0.035840 kernel = «<8, 1024»>, time = 0.035840 kernel = «<16, 32»>, time = 0.064512 kernel = «<16, 64»>, time = 0.043008 kernel = «<16, 128»>, time = 0.035840 kernel = «<16, 256»>, time = 0.035840 kernel = «<16, 512»>, time = 0.035616 kernel = «<16, 1024»>, time = 0.032768 kernel = «<32, 32»>, time = 0.040960 kernel = «<32, 64»>, time = 0.036736 kernel = «<32, 128»>, time = 0.035840 kernel = «<32, 256»>, time = 0.035840 kernel = «<32, 512»>, time = 0.033792 kernel = «<32, 1024»>, time = 0.033792 kernel = «<64, 32»>, time = 0.033792 kernel = «<64, 64»>, time = 0.035840 kernel = «<64, 128»>, time = 0.036864 kernel = «<64, 256»>, time = 0.032768 kernel = «<64, 512»>, time = 0.033792 kernel = «<64, 1024»>, time = 0.034816 kernel = «<128, 32»>, time = 0.034816 kernel = «<128, 64»>, time = 0.033792 kernel = «<128, 128»>, time = 0.033792 kernel = «<128, 256»>, time = 0.034816 kernel = «<128, 512»>, time = 0.033792 kernel = «<128, 1024»>, time = 0.032768 kernel = «<256, 32»>, time = 0.032768 kernel = «<256, 64»>, time = 0.032768 kernel = «<256, 128»>, time = 0.033792 kernel = «<256, 256»>, time = 0.034816 kernel = «<256, 512»>, time = 0.031744 kernel = «<256, 1024»>, time = 0.040960 kernel = «<512, 32»>, time = 0.032800 kernel = «<512, 64»>, time = 0.032768 kernel = «<512, 128»>, time = 0.036864 kernel = «<512, 256»>, time = 0.032768 kernel = «<512, 512»>, time = 0.039936 kernel = «<512, 1024»>, time = 0.049152 kernel = «<1024, 32»>, time = 0.033600 kernel = «<1024, 64»>, time = 0.034816 kernel = «<1024, 128»>, time = 0.032768 kernel = «<1024, 256»>, time = 0.038912 kernel = «<1024, 512»>, time = 0.048128 kernel = «<1024, 1024»>, time = 0.068608
1000000	kernel = «<1, 32»>, time = 8.444096 kernel = «<1, 64»>, time = 4.220928

	kernel = «<1, 128»>, time = 2.151424 kernel = «<1, 256»>, time = 1.170432 kernel = «<1, 512»>, time = 0.667648 kernel = «<1, 1024»>, time = 0.451584 kernel = «<2, 32»>, time = 4.431872 kernel = «<2, 64»>, time = 2.181120 kernel = «<2, 128»>, time = 1.162240 kernel = «<2, 256»>, time = 0.653312 kernel = «<2, 512»>, time = 0.410624 kernel = «<2, 1024»>, time = 0.345088 kernel = «<4, 32»>, time = 2.191360 kernel = «<4, 64»>, time = 1.163264 kernel = «<4, 128»>, time = 0.660480 kernel = «<4, 256»>, time = 0.420864 kernel = «<4, 512»>, time = 0.333824 kernel = «<4, 1024»>, time = 0.332800 kernel = «<8, 32»>, time = 1.162240 kernel = «<8, 64»>, time = 0.655360 kernel = «<8, 128»>, time = 0.409600 kernel = «<8, 256»>, time = 0.336896 kernel = «<8, 512»>, time = 0.331776 kernel = «<8, 1024»>, time = 0.332864 kernel = «<16, 32»>, time = 0.653312 kernel = «<16, 64»>, time = 0.413696 kernel = «<16, 128»>, time = 0.332800 kernel = «<16, 256»>, time = 0.332800 kernel = «<16, 512»>, time = 0.332800 kernel = «<16, 1024»>, time = 0.330752 kernel = «<32, 32»>, time = 0.408576 kernel = «<32, 64»>, time = 0.369664 kernel = «<32, 128»>, time = 0.366592 kernel = «<32, 256»>, time = 0.363520 kernel = «<32, 512»>, time = 0.357376 kernel = «<32, 1024»>, time = 0.361472 kernel = «<64, 32»>, time = 0.371712 kernel = «<64, 64»>, time = 0.364544 kernel = «<64, 128»>, time = 0.364544 kernel = «<64, 256»>, time = 0.364544 kernel = «<64, 512»>, time = 0.363520 kernel = «<64, 1024»>, time = 0.364544 kernel = «<128, 32»>, time = 0.385024 kernel = «<128, 64»>, time = 0.364544 kernel = «<128, 128»>, time = 0.364544 kernel = «<128, 256»>, time = 0.365568 kernel = «<128, 512»>, time = 0.363520 kernel = «<128, 1024»>, time = 0.351200 kernel = «<256, 32»>, time = 0.365568 kernel = «<256, 64»>, time = 0.364544 kernel = «<256, 128»>, time = 0.356352 kernel = «<256, 256»>, time = 0.378880 kernel = «<256, 512»>, time = 0.381952 kernel = «<256, 1024»>, time = 0.389120 kernel = «<512, 32»>, time = 0.376832 kernel = «<512, 64»>, time = 0.365568 kernel = «<512, 128»>, time = 0.352256 kernel = «<512, 256»>, time = 0.345088 kernel = «<512, 512»>, time = 0.351232 kernel = «<512, 1024»>, time = 0.346112 kernel = «<1024, 32»>, time = 0.353280

	kernel = «<1024, 64»>, time = 0.338944 kernel = «<1024, 128»>, time = 0.337920 kernel = «<1024, 256»>, time = 0.339968 kernel = «<1024, 512»>, time = 0.337920 kernel = «<1024, 1024»>, time = 0.343040
10000000	kernel = «<1, 32»>, time = 84.331200 kernel = «<1, 64»>, time = 52.324287 kernel = «<1, 128»>, time = 25.056225 kernel = «<1, 256»>, time = 14.165024 kernel = «<1, 512»>, time = 6.953984 kernel = «<1, 1024»>, time = 5.403648 kernel = «<2, 32»>, time = 52.106239 kernel = «<2, 64»>, time = 24.250368 kernel = «<2, 128»>, time = 14.936096 kernel = «<2, 256»>, time = 8.897440 kernel = «<2, 512»>, time = 4.804608 kernel = «<2, 1024»>, time = 3.348480 kernel = «<4, 32»>, time = 24.724480 kernel = «<4, 64»>, time = 15.467520 kernel = «<4, 128»>, time = 6.571008 kernel = «<4, 256»>, time = 6.450176 kernel = «<4, 512»>, time = 3.362816 kernel = «<4, 1024»>, time = 3.454976 kernel = «<8, 32»>, time = 16.161793 kernel = «<8, 64»>, time = 8.201216 kernel = «<8, 128»>, time = 4.695040 kernel = «<8, 256»>, time = 3.920896 kernel = «<8, 512»>, time = 3.463168 kernel = «<8, 1024»>, time = 4.634624 kernel = «<16, 32»>, time = 9.823232 kernel = «<16, 64»>, time = 4.616192 kernel = «<16, 128»>, time = 3.475456 kernel = «<16, 256»>, time = 3.880960 kernel = «<16, 512»>, time = 3.422208 kernel = «<16, 1024»>, time = 3.974144 kernel = «<32, 32»>, time = 4.751360 kernel = «<32, 64»>, time = 3.936256 kernel = «<32, 128»>, time = 4.586496 kernel = «<32, 256»>, time = 3.497984 kernel = «<32, 512»>, time = 4.211712 kernel = «<32, 1024»>, time = 3.417088 kernel = «<64, 32»>, time = 3.397632 kernel = «<64, 64»>, time = 4.613120 kernel = «<64, 128»>, time = 4.009984 kernel = «<64, 256»>, time = 4.374528 kernel = «<64, 512»>, time = 4.572160 kernel = «<64, 1024»>, time = 4.220928 kernel = «<128, 32»>, time = 3.786752 kernel = «<128, 64»>, time = 4.200448 kernel = «<128, 128»>, time = 4.966400 kernel = «<128, 256»>, time = 3.727360 kernel = «<128, 512»>, time = 4.196352 kernel = «<128, 1024»>, time = 3.426304 kernel = «<256, 32»>, time = 3.550208 kernel = «<256, 64»>, time = 4.271104 kernel = «<256, 128»>, time = 4.004864 kernel = «<256, 256»>, time = 3.289088 kernel = «<256, 512»>, time = 3.305472 kernel = «<256, 1024»>, time = 5.124096

	kernel = «<512, 32»>, time = 3.578880 kernel = «<512, 64»>, time = 4.210528 kernel = «<512, 128»>, time = 4.940800 kernel = «<512, 256»>, time = 3.644416 kernel = «<512, 512»>, time = 4.052992 kernel = «<512, 1024»>, time = 3.291136 kernel = «<1024, 32»>, time = 5.515264 kernel = «<1024, 64»>, time = 3.507200 kernel = «<1024, 128»>, time = 3.314688 kernel = «<1024, 256»>, time = 5.006336 kernel = «<1024, 512»>, time = 3.497984 kernel = «<1024, 1024»>, time = 4.982784

Работа на CPU:

Тест:	Результат:
3 -1 2 -3	0.003
100	0.007
10000	0.161
100000	0.864
1000000	3.587
10000000	22.845

Как видно из тестов производительности, разницу в несколько раз можно заметить только при количестве входных данных > 1000000 чисел.

ВЫВОДЫ

Алгоритм, который я реализовал, очень простой и является вводным в курс ПГП. Однако, когда необходимо произвести какие-то операции над массивами или матрицами очень большого размера, эта программа выигрывает по времени у программы на CPU, следовательно, может использоваться в прикладных задачах.

При написании кода я не столкнулся ни с какими трудностями, но при установке технологии cuda пришлось изрядно помучиться. Нужно было установить дополнительный драйвер для видеокарты, затем прописать путь

корню для файлов cuda, а после отключить secure boot в bios, т. к. он почему-то не давал компьютеру видеть установленный софт. После этого всё успешно заработало.

В конце концов можно сделать вывод, что на маленьких данных программы на CPU работают эффективнее, чем на GPU, т. к. тратится лишнее время на вычисление распределения работы потоков и перенос данных, но на больших объёмах графические процессоры дают существенное преимущество в скорости.