

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Факультет прикладной математики и физики
Кафедра вычислительной математики и программирования

Лабораторная работа №4
по курсу «Программирование графических процессоров»

Работа с матрицам. Метод Гаусса.

Выполнил: Н.И. Забарин

Группа: 8О-408Б

Преподаватели: К.Г. Крашенинников,
А.Ю. Морозов

Москва, 2017

Условие

1. Цель работы:

Использование объединения запросов к глобальной памяти.
Реализация метода Гаусса с выбором главного элемента по столбцу. Ознакомление с библиотекой алгоритмов для параллельных расчетов Thrust.

2. Вариант задания:

Вариант 2. LU-разложение.

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

Необходимо вычислить LU-разложение квадратной матрицы: $A = LU$, где A - матрица $n \times n$, L -- нижняя треугольная матрица, с единичными элементами на диагонали, U -- верхняя треугольная матрица. Так же нужно получить вектор перестановок строк p , где $p[i]$ содержит номер строки которой будет i -ая.

Описание программы

Матрица хранится как одномерный массив размера $n * n$, причем значения группируются по столбцам, то есть $cols[col * n + row]$ (или $cols[col][row]$) – значение коэффициента в col столбце и row строке. Такая организация хранения данных нужна для того, чтобы эффективно находить максимум в столбце.

Максимум ищется с помощью функции `thrust::max_element`, которая возвращает указатель на максимальный элемент. Зная указатель на первый элемент в столбце, можно найти индекс максимального элемента в столбце.

В программе есть два ядра:

- `__global__ void k_swapcolumns(double *dlu, const int n, const int fr, const int to)`
Меняет в матрице `dlu` элементы в `fr` строке с элементами в `to` строке.
- `__global__ void k_lucol(double *dlu, const int n, const int i)`
производит i -тую итерацию алгоритма Гаусса

В матрице `dlu` после работы программы мы получаем матрицу U , что бы получить матрицу L нужно все элементы ниже диагонали разделить на диагональные элементы, это вычисляется на CPU.

Результаты

Замеры времени работы для матрицы $n * n$

n	Параметры ядра	Время
100	<<<8, 8>>>	21.103489
100	<<<16, 16>>>	21.261696
1000	<<<8, 8>>>	894.11
1000	<<<16, 16>>>	883.22

Выводы

Объем глобальной памяти самый большой из всех типов памяти, но в тоже время эта память – самая медленная по скорости считывания и записи, поэтому при работе с глобальной памятью применяются оптимизации.

Выравнивание размеров используемых типов:

позволяет скомпилировать запрос в глобальную память в одну команду GPU, в противном случае компилятор сгенерирует дополнительный код, что может значительно понизить производительность. Для оптимальной производительности тип данных должен иметь размерность 4, 8 или 16 байт.

Использование объединенных запросов (coalescing):

GPU имеет возможность объединять несколько запросов к глобальной памяти в один. Все обращения МР к памяти происходят независимо для каждой половины warp'a. Максимальное объединение получается, когда все запросы одного полу-warп'a удастся объединить в один большой запрос на чтение из глобальной памяти.