МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №4
по курсу «Программирование графических процессоров»

Работа с матрицам. Метод Гаусса.

Выполнил: О.А. Мезенин
Группа: 8О-406Б
Преподаватель: А.Ю. Морозов

Москва, 2024

Условие

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. Использование двухмерной сетки потоков. Исследование производительности программы с помощью утилиты ncu.

Вариант 1. Вычисление детерминанта матрицы.

Входные данные. На первой строке задано число n -- размер матрицы. В следующих n строках, записано по n вещественных чисел -- элементы матрицы. $n \leq 10^4$.

Выходные данные. Необходимо вывести одно число -- детерминант матрицы.

Программное и аппаратное обеспечение

Характеристики графического процессора:

- Compute capability: 8.9
- Name: NVIDIA GeForce RTX 4070 SUPER
- Total Global Memory: 12584550400
- Shared memory per block: 49152
- Registers per block: 65536
- Warp size: 32
- Max threads per block: (1024, 1024, 64)
- Max block: (2147483647, 65535, 65535)
- Total constant memory: 65536
- Multiprocessors count: 56

Характеристики центрального процессора:

- Version: AMD Ryzen 7 7700
- Max Speed: 5389 MHz
- Core Count: 8
- Thread Count: 16

Характеристики оперативной памяти:

- Size: 2x16 GB
- Type: DDR5
- Speed: 4800 MT/s

Характеристики жесткого диска:

- Device Model: ADATA SU635
- User Capacity: 480 103 981 056 bytes [480 GB]
- Sector Size: 512 bytes logical/physical
- SATA Version is: SATA 3.2, 6.0 Gb/s (current: 6.0 Gb/s)

Программное обеспечение:

- OS: Linux
- Kernel: 6.10.13-3-MANJARO
- Текстовый редактор: Kate
- Компилятор: nvcc V12.4.131

Метод решения

Для вычисления детерминанта приведём матрицу к верхнетреугольному виду методом Гаусса, затем перемножим числа, стоящие на главной диагонали.

Алгоритм следующий. На CPU для каждого столбца i матрицы ищем максимальный элемент с помощью библиотеки Thrust. Затем, если максимальный элемент не находится в строке i, то меняем эту строку со строкой i на GPU. После чего обновляем элементы матрицы на GPU (обнуление нижних элементов текущего столбца). После приведения матрицы к верхнетреугольному виду, на CPU перемножаем числа на главной диагонали и умножаем результат на -1, если количество перестановок было нечётное.

Описание программы

Вся программа описана в одном файле main.cu. Есть следующие функции:

- int main() — входная точка программы, которая считывает данные, запускает метод Гаусса, затем вычисляет детерминант;
- double calculateDet(double* matrix, int n, bool is_even_swaps) — возвращает детерминант верхнетреугольной матрицы matrix размера n;
- __global__ void swapRows(double* matrix, int n, int row1, int row2) — переставляет строки row1 и row2 в матрицу matrix размера n;
- float3* calculateAvgVector(uchar4 *data, int width, std::vector<std::vector<int2>>& coors) — вычисляет вектор средних.
- __global__ void updateMatrix(double* matrix, int n, int i) — обновляет элементы матрицы (для обнуления нижних элементов).

Результаты

Будет 4 теста со следующими n: 10, 100, 1000, 10000. Все элементы матрицы являются вещественными числами и варьируются от -1000 до 1000. Конфигурация для swapRows фиксирована — <<<32, 32>>>.

Далее приведены замеры времени работы ядер с четырьмя конфигурациями для updateMatrix, а также замеры работы без использования технологий CUDA (на CPU):

1. Конфигурация <<<dim3(2, 2), dim3(4, 4)>>>

n	Время (в мс)
10	1.46637
100	7.41914
1000	2077.89
10000	Не дождался

2. Конфигурация <<<dim3(4, 4), dim3(8, 8)>>>

n	Время (в мс)
10	1.67968
100	4.90275
1000	212.468
10000	175112

3. Конфигурация <<<dim3(8, 8), dim3(16, 16)>>>

n	Время (в мс)

10	1.77261
100	5.01923
1000	120.909
10000	24118.8

4. Конфигурация <<<dim3(16, 16), dim3(32, 32)>>>

n	Время (в мс)
10	1.65552
100	4.93194
1000	111.124
10000	17493.2

5. CPU

n	Время (в мс)
10	0.002
100	1.127
1000	1012.929
10000	1564499.222

Исследование производительности

Исследование производительности производилось с помощью утилиты ncu.

Запуск:

```
~/st/MAI-PGP-PDP/lab4 main !2 ?2 sudo /usr/local/NVIDIA-Nsight-Compute/ncu --metrics l1tex__t_sectors_pipe_lsu_mem_global_op_ld,l1tex__t_sectors_pipe_lsu_mem_global_op_st,smsp__branch_targets_threads_divergent,sm__sass_inst_executed_op_local,l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld,l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st --print-summary per-gpu ./a.out < tests/t1000.txt
```

Вывод для swapRows и updateMatrix:

```
swapRows(double *, int, int, int) (32, 1, 1)x(32, 1, 1), Invocations 993
  Section: Command line profiler metrics
  ------------------------------------------------------------ ----------- -------- -------- --------
  Metric Name                                                  Metric Unit  Minimum  Maximum  Average
  ------------------------------------------------------------ ----------- -------- -------- --------
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                      0,00     0,00     0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                      0,00     0,00     0,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.avg                sector        35,71    35,71    35,71
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.max                sector        64,00    64,00    64,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.min                sector         0,00     0,00     0,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.sum                sector 2 000,00 2 000,00 2 000,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.avg                sector        35,71    35,71    35,71
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.max                sector        64,00   105,00    95,33
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.min                sector         0,00     0,00     0,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.sum                sector 2 000,00 2 000,00 2 000,00
  sm__sass_inst_executed_op_local.avg                             inst         0,00     0,00     0,00
  sm__sass_inst_executed_op_local.max                             inst         0,00     0,00     0,00
  sm__sass_inst_executed_op_local.min                             inst         0,00     0,00     0,00
  sm__sass_inst_executed_op_local.sum                             inst         0,00     0,00     0,00
  smsp__branch_targets_threads_divergent                      branches         0,00     0,00     0,00
  ------------------------------------------------------------ ----------- -------- -------- --------

updateMatrix(double *, int, int) (16, 16, 1)x(32, 32, 1), Invocations 999
  Section: Command line profiler metrics
  ------------------------------------------------------------ ----------- ------- ---------- ----------
  Metric Name                                                  Metric Unit Minimum    Maximum    Average
  ------------------------------------------------------------ ----------- ------- ---------- ----------
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.avg                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.max                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.min                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.avg                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.max                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.min                      0,00       0,00       0,00
  l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                      0,00       0,00       0,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.avg                sector        0,07  11 167,39   3 634,49
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.max                sector        4,00  13 103,00   4 257,22
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.min                sector        0,00  10 014,00   3 000,22
  l1tex__t_sectors_pipe_lsu_mem_global_op_ld.sum                sector        4,00 625 374,00 203 531,69
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.avg                sector        0,02   5 012,84   1 627,00
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.max                sector        1,00   5 822,00   1 901,32
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.min                sector        0,00   4 518,00   1 329,26
  l1tex__t_sectors_pipe_lsu_mem_global_op_st.sum                sector        1,00 280 719,00  91 112,12
  sm__sass_inst_executed_op_local.avg                             inst         0,00       0,00       0,00
  sm__sass_inst_executed_op_local.max                             inst         0,00       0,00       0,00
  sm__sass_inst_executed_op_local.min                             inst         0,00       0,00       0,00
  sm__sass_inst_executed_op_local.sum                             inst         0,00       0,00       0,00
  smsp__branch_targets_threads_divergent                      branches         0,00     512,00     241,91
  ------------------------------------------------------------ ----------- ------- ---------- ----------
```

Выводы

В ходе лабораторной работы ознакомился с библиотекой алгоритмов для параллельных расчетов Thrust и утилитой для профилирования ncu, также узнал, как считать определитель верхнетреугольной матрицы. Реализовал вычисление детерминанта матрицы через метод Гаусса с выбором главного элемента по столбцу. Профилирование даёт возможность провести анализ работы программы для выявления некоторых проблем и помогает оптимизировать программу. С помощью ncu вывел такие метрики, как количество обращений к глобальной памяти, к локальной памяти; количество дивергенций; количество конфликтов банков памяти. Не уверен насчёт количества обращений к глобальной памяти, но, судя по остальным метрикам, всё работает нормально. Хоть и дивергенции есть, но кажется, что такое маленькое количество не критично.

Из результатов сравнения можно увидеть, что CPU выигрывает по времени на тестах n=10, n=100; но с n=1000 начинает проигрывать даже самой маленькой конфигурации. Т.е. на больших данных эффективнее использовать GPU.