МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

Лабораторная работа №5
по курсу «*Параллельные и распределенные вычисления*»

Сортировка чисел на GPU. Свертка, сканирование, гистограмма..

Выполнил: *Новиков Сергей Сергеевич*
Группа: М8О-311Б-22
Преподаватель: А.Ю. Морозов
Е.Е. Заяц

Москва, 2025

Условие

Цель работы. Ознакомление с фундаментальными алгоритмами GPU: свертка (reduce), сканирование (blelloch scan) и гистограмма (histogram). Реализация одной из сортировок на CUDA. Использование разделяемой и других видов памяти.

Исследование производительности программы с помощью утилиты nvprof (обязательно отразить в отчете).

Вариант 8. Поразрядная сортировка.

Программное и аппаратное обеспечение

GPU Name: AMD Radeon Graphics
Compute Capability: 9.0
Total Global Memory: 9679 MB
Shared Memory Per Block: 64 KB
Constant Memory: 2097151 KB
Registers Per Block: 65536
Max Threads Per Block: 1024
Multiprocessors: 6
CPU Cores: 6
Total RAM: 19358 MB
Total Disk Space: 233 GB
Operating System: Linux
Compiler: GCC 4.2

Метод решения

Решение включает два ключевых этапа: построение гистограммы с атомарными операциями и алгоритм сканирования (префиксной суммы). На первом этапе каждый рабочий элемент обрабатывает часть входного массива, используя атомарные операции для инкремента счётчиков в гистограмме, что гарантирует корректность при параллельном доступе к общим ячейкам. На втором этапе выполняется рекурсивное сканирование: массив разбивается на блоки, обрабатываемые в разделяемой памяти с бесконфликтным, после чего рекурсивно вычисляются префиксные суммы для агрегированных значений блоков, что позволяет обрабатывать массивы произвольного размера. Итоговые позиции элементов определяются на

основе гистограммы и префиксных сумм, обеспечивая распределение данных в отсортированном порядке.

Описание программы

Алгоритм был декомпозирован на следующие составные части:

1. Main – организация ввода, запуска остальных ядер, и вывода. Сначала она запускает ядро, которое заполняет массив гистограммы нулями, а затем осуществляет подсчет значений. Вызывает алгоритм exclusive_scan, на основе результата которого восстанавливает исходный массив в отсортированном порядке.
2. Excludive_scan – осуществляет подсчет префиксной суммы в 3 этапа.
    a. Рассчитывает сумму для блоков длиной равной 2 рабочим группам
    b. В случае необходимости вызывает себя рекурсивно, чтобы рассчитать сумму для корректировки ранее подсчитанных значений, начиная с первого блока
    c. Производит корректировку

Результаты

Конфигурация	128, 8	1024, 32	8192, 128	CPU (1 thread)
10	2010278300	300259000	72248400	16743543000
10000	2005672300	305959300	73906900	16044377400
1000000	2023094900	326498400	76009000	15881732900

Показано время в наносекундах(нс)

Работа с отчетом NVIDIA Nsight Compute

Профилирование программы показало, что программа написана достаточно хорошо (использовалось смещение для избавления от конфликта банков памяти при доступе к разделяемой памяти), но имеет 2 недостатка:

- SM Workload Imbalance
- L1 Slices Workload Imbalance

Оба являются следствием использования статической сетки потоков

Выводы

Данный алгоритм показывает преимущество использования параллельных вычислений, т.к. выигрыш насчитывает десятки раз. Это первый алгоритм, который показал настолько существенную разницу. В ходе разработки часто возникали ошибки, связанные с неправильной итерацией внутри ядер, но при помощи отладчика и возможности запуска ядер на cpu их удалось исправить.