МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №5
по курсу «Параллельная обработка данных»

Сортировка чисел на GPU. Свертка, сканирование, гистограмма.

Выполнил: Д.В. Коростелев
Группа: 8О-408Б
Преподаватели: К.Г. Крашенинников,
А.Ю. Морозов

Москва, 2022

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.

Вариант 6:

Требуется реализовать карманную сортировку для чисел типа float.
Должны быть реализованы:

- Алгоритм гистограммы, с использованием атомарных операций.
- Алгоритм свертки для любого размера, с использованием разделяемой памяти.
- Алгоритм сканирования для любого размера, с использованием разделяемой памяти. (Можно воспользоваться библиотекой Thrust)
- Алгоритм битонической сортировки для карманов.

Программное и аппаратное обеспечение

- Графический процессор NVIDA GeForce GTX 1050

Графическая память	2 Gb
Разделяемая память на блок	48 Mb
Константная память	64 Mb
Количество регистров на блок	65536
Максимальное количество блоков на процессор	32
Максимальное количество потоков на блок	1024
Количество мультипроцессоров	5

- Процессор Intel Core i7-7700HQ 4x 2.808ГГц

Количество ядер	4
Количество потоков	8
Базовая тактовая частота	2.8 GHz
Максимальная тактовая частота	3.8 GHz
Кеш-память	6 Mb

- Оперативная память DDR4-SODIMM

Объем памяти	8 Gb
Частота	2400 MHz
Форм-фактор	SODIMM
Количество плашек	2

- SSD и HDD накопители

Объем SSD накопителя	128 Gb
Объем HDD накопителя	1 Tb

- Программное обеспечение

Операционная система	Windows 10 Pro
Средство разработки на CUDA (IDE)	Microsoft Visual Studio 2019
Компиляторы	MSVC 2019
Версия CUDA Toolkit	11.4.2
Дополнительный текстовый редактор	Notepad++

Метод решения

Требуется реализовать карманную сортировку с битонической сортировкой карманов на множестве чисел с плавающей точкой. Для этого также требуется реализовать алгоритм гистограммы и скана, а также редукций нахождения минимумов и максимумов в заданных промежутках. Первые два алгоритма будут применятся для того, чтобы раскидать элементы по нужным пакетам (карманам), полученные ключи для каждого элемента (номер кармана, в который попал элемент) сортируем при помощи сортировки подсчетом. В итоге получаем индексы сплитов, которые будут отсортированы, и кроме того будет известно в каких позициях по номеру кармана нужно поставить элемент в конечном массиве, также будут известны индексы начала и конца в конечном массиве элементов каждого кармана. Данный процесс повторяем рекурсивно (в моей реализации будет использован стек), пока не получим сплиты, которые будут помещаться в карманы большего размера, которые будут помещаться в разделяемую память. Затем все карманы сортируем при помощи бионической сортировки.

Результаты

Первой строкой идет кол-во блоков и тредов для алгоритма редукции, второй – для бионической сортировки.

Задача \ Сетка	std::sort	64, 64 128, 128	128, 128 256, 256	256, 256 512, 512
100 элементов	27 мкс	155 мс	199 мс	229 мс
10 000 элементов	5178 мкс	220 мс	195 мс	213 мс
500 000 элементов	243 мс	212 мс	208 мс	188 мс
1 000 000 элементов	447 мс	193 мс	229 мс	195 мс
10 000 000 элементов	4349 мс	283 мс	305 мс	351 мс
100 000 000 элементов	-	4597 мс	5345 мс	6245 мс

Выводы

Крайне сложная лабораторная работа, с крайне сложным вариантом. На реализацию программы и дебаг потребовалось крайне много времени, даже несмотря на то, что по варианту разрешено использовать алгоритм скана из библиотеки thrust. В сумме получился внушительный алгоритм, разобраться в котором довольно тяжело и сложно, однако скорость сортировки значительно увеличилась, по сравнению с однопоточным алгоритмом.