

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №5
по курсу «Программирование графических процессоров»
Сортировка чисел на GPU. Свертка, сканирование, гистограмма.

Выполнил: М.С.Гаврилов
Группа: 8О-406Б
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 5.

Сортировка чет-нечет с предварительной битонической сортировкой.

Программное и аппаратное обеспечение

Характеристики графического процессора	
Compute capability:	7.5
Name:	NVIDIA GeForce GTX 1650
Total Global Memory:	4102029312
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:	14

Процессор: Intel(R) Core(TM) i5-11260H @ 2.60GHz

Оперативная память: 7875 Мб

Накопитель: kimtigo SSD 256GB

OS: Linux Mint 21

IDE: Visual Studio Code 1.72.0

compiler: Cuda compilation tools, release 11.8, V11.8.89

Метод решения

Для хранения и обработки массивов используется класс array. Ядра работают одновременно над всем массивом, разделение на блоки учитывается внутри ядра. Каждый блок копируется в разделяемую память, после чего к нему применяются полуфильтры с шагом от n (задается при запуске ядра) до 2. Перед каждым запуском основного ядра, к массиву применяется вспомогательное ядро, так как на первом шаге что сортировки, что слияния, нужен полуфильтр собой конфигурации. После сортировки блоков, массив подвергается блочной сортировке чет-нечет, слияния между блоками производятся с помощью тех же ядер, только размер блоков, поступающий на вход, равен двум размерам сортировочных блоков.

Описание программы

Программа состоит из одного файла, в котором реализован класс array и ядра.

Класс array

Имеет два члена – длина и ссылка на область памяти, где хранятся элементы.

Метод	Описание
<code>array(int len_)</code>	Конструктор, что считывает элементы со stdin
<code>void print_uf()</code>	Выводит элементы массива на stdout.
<code>void upfill()</code>	Приводит размер массива к ближайшей большей степени двойки.
<code>void odd_even_block(int block_size)</code>	Выполняет сортировку чет-нечет с разбиением по блокам размера block size.

Ядра

<code>__global__ void kernel_bitonic_step_and_proceed(int* elements, int n, int init_bs)</code>
<code>__global__ void kernel_bitonic_sort_init(int* elements, int n, int block_size)</code>

Первое ядро выполняет копирование элементов в разделяемую память и затем обрабатывает их в цикле, уменьшая битонический коэффициент на каждом шаге. Число потоков в блоке на мультипроцессоре должно быть равно длине двух блоков, чтобы вычисления можно проводить в полностью в разделяемой памяти.

Второе ядро используется на первом шаге, так как при слияниях и сортировках сравнения на первом шаге выполняются по отличной схеме.

Результаты

1. Сравнение времени работы ядра с различными конфигурациями. Размер сортировочных блоков равен половине размера блоков потоков мультипроцессора.

Размер файла 1091 элементов.

Размерность ядра	Время работы (мс)
<code><<<32,32>>></code>	0.988288
<code><<<32,64>>></code>	0.4736

<<<64,64>>>	0.3736
<<<512,512>>>	0.12304
<<<1024,1024>>>	0.28592

Размер файла 1048590 элементов.

Размерность ядра	Время работы (мс)
<<<64,64>>>	22167.2
<<<512,512>>>	2462.23
<<<1024,1024>>>	1673.47

2. Сравнение времени работы CPU и ядра (конфигурация <<64 64>>)

Размер теста	Время на CPU (мс)	Время на GPU (мс)
1 000 x 1 000 pix	181	0.3736

3. Примеры работы программы

До сортировки:
785 744 484 755 36 51 171 434 180 592 344 843 375 425 392 281 569 167 978 143 655 331 196 704 650 664 193 520 637 342 921 537 481 923 497 376 111 498 740 570 726 753 810 593 53 381 393 396 515 593 265 429 109 351 836 36 704 566 806 189 944 957 545 749 696 559 644 547 611 223 909 308 818 469 356 407 167 714 402 467 873 712 923 189 929 926 568 198 868 963 159 805 746 632 606 729 903 243 110 746 520 537 165 216 248 396 903 644 131 45 510 118 891 972 871 318 849 342 270 135 110 214 61 528 59 119 41 325 910 208 318 54 21 313 620 79 624 365 218 495 490 25 266 493 936 156 109 544 806 751 824 701 446 87 935 351 98 193 481 369 412 629 540 152 804 142 851 127 442 81 845 754 132 556 848 880 34 761 645 499 472 26 502 969 84 991 981 96 816 422 719 677 690 19 57 14 291 713 361 41 317 655 641 44 376 798 431 319 487 386 336 575 790 63 271 764 815 742 712 420 669 328 474 513 912 834 532 753 179 665 815 486 450 840 772 285 280 240 816 298 328 40 581 532 674 549 27 588 784 795 183 812 849 292 162 822 366 401 101 778 596 199 339 804 465 795 953 301 938 90 729 753 100 906 557 194 770 965 826 395 23 15 530 127 820 421 127 298 460 456 517 96 477 539 455 861 802 89 205 212 647 115 364 605 773 41 287 521 231 742 956 887 684 369 898 516 245 609 874 318 953 912 771 314 960 164 379 663 789 394 98 971 975 339 841 220 201 140 571 769 481 409 562 450 270 306 46 965 62 776 834 243 494 17 239 874 451 346 21 204 641 861 511 360 458 44 902 183 289 76 521 26 376 749 20 683 406 609 974 361 173 331 903 879 547 730 538 279 559 614 455 227 159 443 868 300 368 604 61 527 365 593 794 981 811 803 864 401 605 471 299 584 987 778 138 20 768 116 131 633 932 288 680 852 263 571 442 472 16 811 686 266 109 610 125 920 651 635 328 638 961 260 643 185 416 804 754 368 629 415 490 872 418 712 625 254 761 14 512 458 915 554 607 98 321 920 794 196 265 100 803 105 458 447 713 478 339 962 431 381 489 738 935 532 904 948 913 755 45 251 670 372 313 744 586 470 152 722 626 226 523 310 378 34 202 448 522 678 910 755 10 730 62 141 331 981 712 927 96 890 931 867 821 228 730 551 542 510 875 393 524 329 497 171 68 801 415 553 328 530 247 480 258 998 178 17 85 651 200 527 654 715 50 86 412 572 654 628 92 561 650 644 632 584 533 840 281 534 907 455 306 144 975 992 639 291 558 470 150 574 697 188 460 662 923 75 739 778 844 729 779 185 221 900 247 977 477 736 657 577 140 493 568 465 454 636 9 392 274 939 324 382 803 608 385 684 422 106 339 128 261 326 75 157 988 457 136 19 146 369 765 955 163 641 611 37 779 12 225 301 749 809 417 771 317 504 926 179 307 861 595 893 905 682 149 701 702 250 915 357 860 435 796 674 679 51 706 701 552 47 998 466 297 104 317 806 313 623 505 996 921 961 435 17 976 86 559 489 469 855 744 937 675 191 943 191 746 516 670 431 107 158 273 319 348 487 455 77 597 266 548 968 34 601 331 740 178 82 332 945 511 794 96 534 383 453 252 947 2 468 397 732 285 784 617 741 687 30 575 314 714 924 567 605 228 451 836 366 486 773 417 942 709 688 218 828 47 756 541 60 744 978 97 245 894 390 598 826 218 678 733 75 944 715 536 173 208 519 277 335 78 751 160 30 990 572
После сортировки:

```

785 | 2 9 10 12 14 14 15 16 17 17 17 19 19 20 20 21 21 23 25 26 26 27 30 30 34 34 34 36 36 37 40 41 41 41 44 44
45 45 46 47 47 50 51 51 53 54 57 59 60 61 61 62 62 63 68 75 75 75 76 77 78 79 81 82 84 85 86 86 87 89 90 92 96
96 96 96 97 98 98 98
100 100 101 104 105 106 107 109 109 109 110 110 111 115 116 118 119 125 127 127 127 128 131 131 132 135
136 138 140 140 141 142 143 144 146 149 150 152 152 156 157 158 159 159 160 162 163 164 165 167 167 171
171 173 173 178 178 179 179 180 183
183 185 185 188 189 189 191 191 193 193 194 196 196 198 199 200 201 202 204 205 208 208 212 214 216 218
218 218 220 221 223 225 226 227 228 228 231 239 240 243 243 245 245 247 247 248 250 251 252 254 258 260
261 263 265 265 266 266 266 270 270
271 273 274 277 279 280 281 281 285 285 287 288 289 291 291 292 297 298 298 299 300 301 301 306 306 307
308 310 313 313 313 314 314 317 317 318 318 318 319 319 321 324 325 326 328 328 328 328 329 331 331
331 331 332 335 336 339 339 339 339
342 342 344 346 348 351 351 356 357 360 361 361 364 365 365 366 366 368 368 369 369 369 372 375 376 376
376 378 379 381 381 382 383 385 386 390 392 392 393 393 394 395 396 396 397 401 401 402 406 407 409 412
412 415 415 416 417 417 418 420 421
422 422 425 429 431 431 431 434 435 435 442 442 443 446 447 448 450 450 451 451 453 454 455 455 455 455
456 457 458 458 458 460 460 465 465 466 467 468 469 469 470 470 471 472 472 474 477 477 478 480 481 481
481 484 486 486 487 487 489 489 490
490 493 493 494 495 497 497 498 499 502 504 505 510 510 511 511 512 513 515 516 516 517 519 520 520 521
521 522 523 524 527 527 528 530 530 532 532 532 533 534 534 536 537 537 538 539 540 541 542 544 545 547
547 548 549 551 552 553 554 556 557
558 559 559 559 561 562 566 567 568 568 569 570 571 571 572 572 574 575 575 577 581 584 584 586 588 592
593 593 593 595 596 597 598 601 604 605 605 605 606 607 608 609 609 610 611 611 614 617 620 623 624 625
626 628 629 629 632 632 633 635 636
637 638 639 641 641 641 643 644 644 644 645 647 650 650 651 651 654 654 655 655 657 662 663 664 665 669
670 670 674 674 675 677 678 678 679 680 682 683 684 684 686 687 688 690 696 697 701 701 701 702 704 704
706 709 712 712 712 712 713 713 714
714 715 715 719 722 726 729 729 729 730 730 730 732 733 736 738 739 740 740 741 742 742 744 744 744 744
746 746 746 749 749 749 751 751 753 753 753 754 754 755 755 755 756 761 761 764 765 768 769 770 771 771
772 773 773 776 778 778 778 779 779
784 784 789 790 794 794 794 795 795 796 798 801 802 803 803 803 804 804 804 805 806 806 806 809 810 811
811 812 815 815 816 816 818 820 821 822 824 826 826 828 834 834 836 836 840 840 841 843 844 845 848 849
849 851 852 855 860 861 861 861 864
867 868 868 871 872 873 874 874 875 879 880 887 890 891 893 894 898 900 902 903 903 903 904 905 906 907
909 910 910 912 912 913 915 915 920 920 921 921 923 923 923 924 926 926 927 929 931 932 935 935 936 937
938 939 942 943 944 944 945 947 948
953 953 955 956 957 960 961 961 962 963 965 965 968 969 971 972 974 975 975 976 977 978 978 981 981 981
987 988 990 991 992 996 998 998

```

4. Исследование с помощью nvprof.

==14388== Profiling result:

Type	Time(%)	Time	Calls	Avg	Min	Max	Name
GPU activities:	51.43%	237.08us	12	19.756us	19.554us	20.642us	kernel_bitonic_sort_init(int*, int, int)
	47.83%	220.47us	11	20.042us	19.745us	20.930us	kernel_bitonic_step_and_proceed(int*, int, int)
	0.37%	1.6960us	1	1.6960us	1.6960us	1.6960us	[CUDA memcpy DtoH]
	0.37%	1.6960us	1	1.6960us	1.6960us	1.6960us	[CUDA memcpy HtoD]
API calls:	94.14%	846.19ms	2	423.10ms	5.8670us	846.19ms	cudaEventCreate
	5.55%	49.892ms	1	49.892ms	49.892ms	49.892ms	cuDevicePrimaryCtxRelease
	0.08%	719.16us	23	31.267us	10.756us	228.31us	cudaLaunchKernel
	0.07%	634.58us	2	317.29us	276.22us	358.36us	cudaEventSynchronize
	0.05%	407.25us	2	203.62us	70.889us	336.36us	cudaMemcpy
	0.04%	399.91us	101	3.9590us	2.4450us	18.089us	cuDeviceGetAttribute
	0.03%	275.73us	1	275.73us	275.73us	275.73us	cudaMalloc
	0.01%	96.310us	4	24.077us	7.8220us	46.444us	cudaEventRecord
	0.01%	91.911us	2	45.955us	43.022us	48.889us	cudaEventElapsedTime
	0.01%	57.689us	1	57.689us	57.689us	57.689us	cuModuleUnload
	0.00%	24.444us	3	8.1480us	2.9330us	18.089us	cuDeviceGetCount
	0.00%	10.267us	2	5.1330us	4.8890us	5.3780us	cuDeviceGet
	0.00%	8.8000us	1	8.8000us	8.8000us	8.8000us	cuDeviceGetName
	0.00%	4.4000us	1	4.4000us	4.4000us	4.4000us	cuModuleGetLoadingMode

0.00%	4.4000us	1	4.4000us	4.4000us	4.4000us	cuDeviceGetUuid
0.00%	3.9110us	1	3.9110us	3.9110us	3.9110us	cuDeviceTotalMem
0.00%	3.4220us	1	3.4220us	3.4220us	3.4220us	cuDeviceGetLuid

Вывод

В ходе выполнения этой лабораторной работы я ознакомился с методами сортировки массивов на мультипроцессоре и реализовал алгоритм сортировки чет-нечет с предварительной битонической сортировкой. Такие алгоритмы часто применяются для сортировки на графических процессорах. Я получил опыт работы с разделяемой памятью на CUDA.

Основной трудностью при выполнении работы была невозможность заставить windows powershell работать с бинарным вводом-выводом, не искажая его. Пришлось использовать bash и делать отдельную версию программы, считывающие данные из файла, а не со stdin.

Версия на мультипроцессоре, оказалась быстрее, чем на CPU, уменьшение затраченного времени при увеличении размерности сетки наблюдалось.