

Теория параллелизма

Отчет

Лабораторные работы номер 6,7,8

Выполнил 23932, Тишкин А.А.

18.04.2025

Цель работы: реализовать решение уравнение теплопроводности (разностная схема – пятиточечный шаблон) в двумерной области на равномерных сетках (128^2 , 256^2 , 512^2 , 1024^2). Провести профилированию и оптимизацию кода, выполнить сравнительный анализ скорости выполнения на CPU в одном и нескольких потоках и GPU.

Используемый компилятор: g++

Используемый профилировщик: Nsight Systems

Как производили замер времени работы: библиотека <chrono>

Выполнение на CPU

CPU-onecore

Размер сетки	Время выполнения	Точность	Кол-во итераций
128*128	1,05932	0,000001	34542
256*256	14,2217	0,000001	116257
512*512	190,29	0,000001	374821

CPU-multicore

Размер сетки	Время выполнения	Точность	Кол-во итераций
128*128	1,49251	0,000001	34542
256*256	4,24443	0,000001	116257
512*512	43,3714	0,000001	374821
1024*1024	293,913	0,000001	1000000

Время работы onecore и multicore

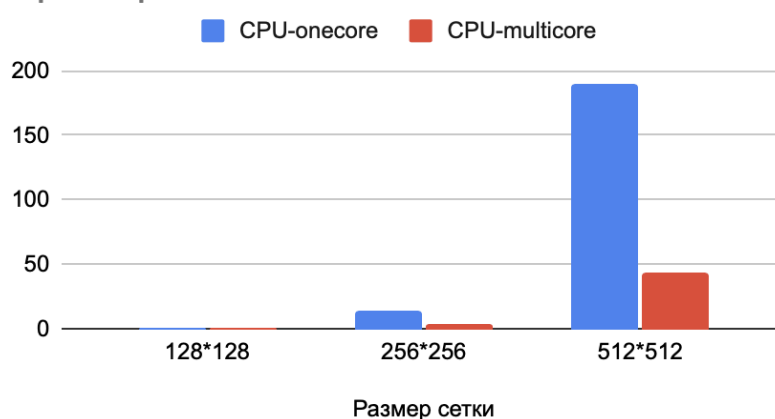


Диаграмма сравнения время работы CPU-one и CPU-multi

Выполнение на GPU

Этапы оптимизации на сетке 512*512

Этап	Время выполнения	Точность	Максимальное количество итераций	Комментарий
1	17,0808	0,000001	1_000_000	Неоптимизированный код
2	12,4292	0,000001	1_000_000	Был оптимизирован swap: вместо поэлементного копирования используется std::swap

(количество итераций при профилировании Nsight Systems до 1000)

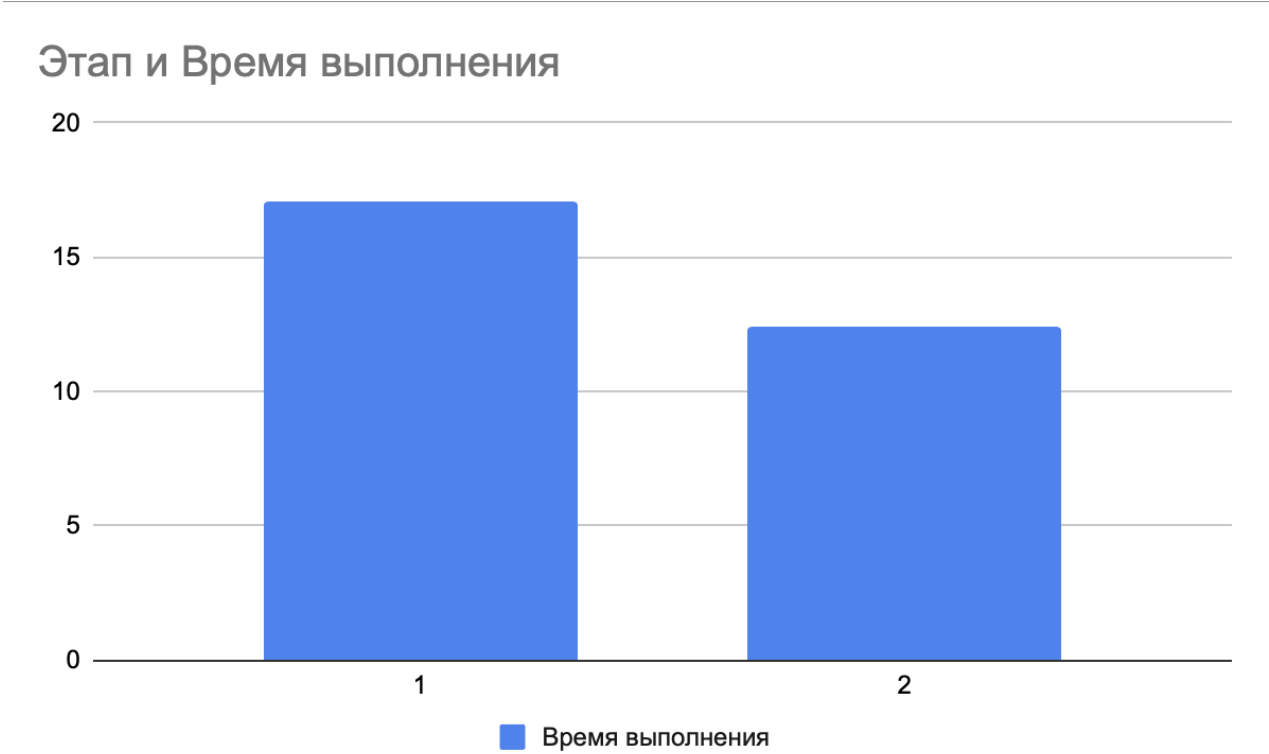
Time (%)	Total Time (ns)	Instances	Avg (ns)	Med (ns)	Min (ns)	Max (ns)	StdDev (ns)	Style	Range
49.7	27694719211	1	27694719211.0	27694719211.0	27694719211	27694719211	0.0	PushPop	loop
36.2	20162540027	374821	53792.5	53083.0	49885	3647625	18603.3	PushPop	calc
13.1	7270930886	374821	19398.4	19012.0	17671	3432831	10065.9	PushPop	swap
1.0	546194302	1	546194302.0	546194302.0	546194302	546194302	0.0	PushPop	init

неоптимизированная версия

Time (%)	Total Time (ns)	Instances	Avg (ns)	Med (ns)	Min (ns)	Max (ns)	StdDev (ns)	Style	Range
49.6	20544924542	1	20544924542.0	20544924542.0	20544924542	20544924542	0.0	PushPop	loop
48.8	20231010903	374821	53975.1	53154.0	49995	6966536	22771.1	PushPop	calc
1.3	557181351	1	557181351.0	557181351.0	557181351	557181351	0.0	PushPop	init
0.2	88603660	374821	236.4	201.0	109	21188	268.5	PushPop	swap

оптимизированная версия

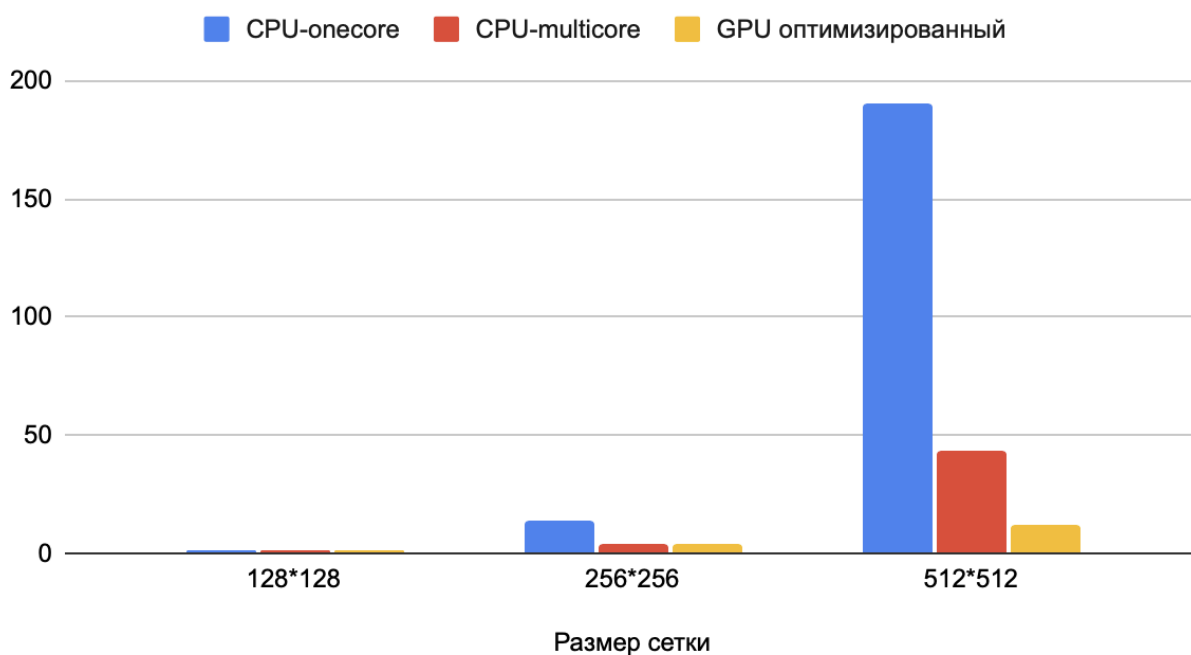
Диаграмма оптимизации



GPU – оптимизированный вариант

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	1,03403	0,000001	34542
256*256	3,57647	0,000001	116257
512*512	12,4283	0,000001	374821
1024*1024	65,9855	0,000001	1000000

Время работы каждого из вариантов



Cublas

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	4,02982	0,000001	34542
256*256	27,4421	0,000001	116257
512*512	199,858	0,000001	374821

Time (%)	Total Time (ns)	Instances	Avg (ns)	Med (ns)	Min (ns)	Max (ns)	StdDev (ns)	Style	Range
46.5	30375870034	1	30375870034.0	30375870034.0	30375870034	30375870034	0.0	PushPop	loop
46.4	30293998761	116257	260577.8	247420.0	213174	3582288	82193.2	PushPop	calc
6.2	4046197488	116257	34803.9	35406.0	26127	3358387	12516.5	PushPop	cublas
0.8	537094131	1	537094131.0	537094131.0	537094131	537094131	0.0	PushPop	init
0.0	24224488	116257	208.4	182.0	97	22377	328.7	PushPop	swap

```
i.purtov@d2e6a4e2eddd:~/concurrency_theory/task6$ ./task6_gpu 0.000001 10 1000000
10 11.1111 12.2222 13.3333 14.4444 15.5556 16.6667 17.7778 18.8889 20
11.1111 12.2222 13.3333 14.4444 15.5555 16.6666 17.7778 18.8889 20 21.1111
12.2222 16.3546 15.5083 16.5768 16.8794 17.565 17.8676 18.9362 18.0898 22.2222
13.3333 16.9824 18.8187 17.2787 17.6876 18.7914 19.2753 17.6978 19.384 23.3333
14.4444 17.7135 18.8906 20.0683 18.7301 19.8212 18.3966 19.3192 20.5864 24.4444
15.5556 18.2255 19.8519 21.3266 22.4418 18.4404 19.0978 20.5363 22.4263 25.5556
16.6667 19.1889 20.7694 22.2867 21.1744 22.2067 20.7176 20.8792 22.5041 26.6667
17.7778 21.3795 23.1176 21.1063 21.7012 22.2136 23.1107 21.7154 23.4482 27.7778
18.8889 22.6056 21.5385 21.8471 22.525 22.9201 24.1066 25.6136 25.0145 28.8889
20 21.1111 22.2222 23.3333 24.4444 25.5556 26.6667 27.7778 28.8889 30
```

```
i.purtov@d2e6a4e2eddd:~/concurrency_theory/task6$ ./task6_gpu 0.000001 13 1000000
10 10.8333 11.6667 12.5 13.3333 14.1667 15 15.8333 16.6667 17.5 18.3333 19.1667 20
10.8333 11.6667 12.5 13.3333 14.1666 15 15.8333 16.6666 17.5 18.3333 19.1666 20 20.8333
11.6667 15.711 14.7095 15.6346 15.8257 16.292 16.6667 17.0413 17.5076 17.6988 18.6239 17.6223 21.6667
12.5 16.1986 17.8835 16.1475 16.7887 17.5212 17.4178 17.2873 17.9949 18.7152 16.9479 18.4559 22.5
13.3333 16.3808 18.0304 19.2045 17.2355 17.2115 18.0803 19.0521 19.1926 17.1495 18.0271 19.731 23.3333
14.1667 17.1478 18.5114 18.9909 19.6448 17.8192 18.7638 19.8796 18.0765 18.4357 18.7015 20.2088 24.1667
15 17.3198 18.3422 19.2031 20.4075 21.6537 19.8786 17.8605 18.4104 19.361 20.5645 22.1621 25
15.8333 18.078 19.2659 20.3275 21.4302 22.5501 20.7747 18.882 19.352 20.2478 21.2531 22.2787 25.8333
16.6667 19.568 20.2004 21.12 22.1582 20.5983 21.482 22.5641 20.7872 20.6301 21.2316 22.7694 26.6667
17.5 20.4869 22.154 23.288 20.9337 21.3264 22.0594 22.4182 23.1399 21.146 21.6953 23.3591 27.5
18.3333 21.6823 23.2615 21.7041 22.3287 22.2487 21.7894 22.0869 22.9785 24.0626 22.6208 24.2007 28.3333
19.1667 22.9436 21.737 21.6759 21.8844 22.4056 23.151 23.8867 24.1663 25.1578 26.431 25.5661 29.1667
20 20.8333 21.6667 22.5 23.3333 24.1667 25 25.8333 26.6667 27.5 28.3333 29.1667 30
```

Вывод: Скорость выполнения на GPU самая большая за счет большого количества вычислительных ядер. Оптимизация кода важна и дает ощутимое ускорение. Профилирование позволяет найти участки, занимающие больше всего времени. Для того, чтобы обменивать массивы существует `std::swap`, а если его нет под рукой, то нужно передать друг другу значения указателей.