

Теория параллелизма

Отчет

Лабораторная работа 7

Выполнила Краснопёрова Алёна, 22931
25.05.24

Цель работы:

Реализовать и оптимизировать решение уравнение теплопроводности (разностная схема –пятиточечный шаблон) в двумерной области на равномерных сетках с использованием GPU и директив OpenACC.

Используемый компилятор: gdc++

Используемый профилировщик: “Nsight Systems”.

Замер времени работы: Библиотека “chrono”

Выполнение на CPU

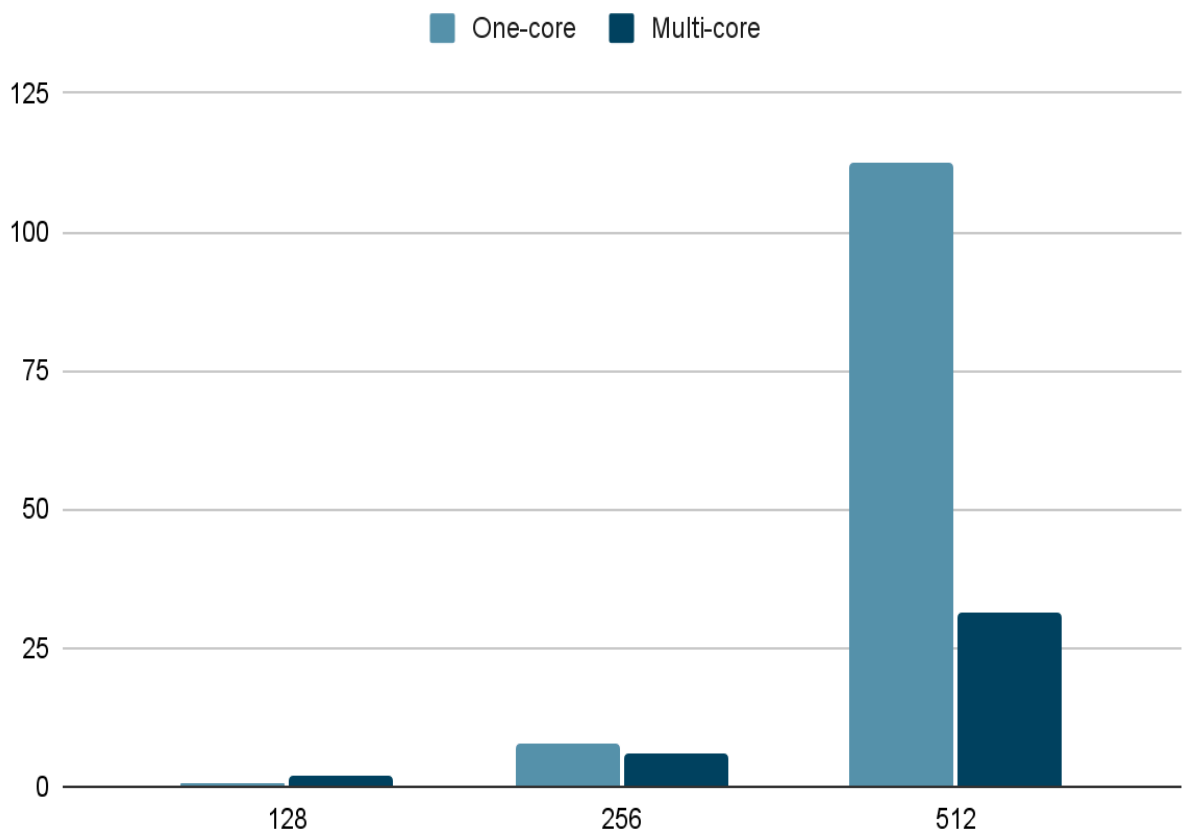
CPU-onecore

Размер сетки	Время выполнения	Точность	Количество операций
128*128	0.51	0.000001	30100
256*256	7.83	0.000001	102900
512*512	112.53	0.000001	339600

CPU-multicore

Размер сетки	Время выполнения	Точность	Количество операций
128*128	1.94	0.000001	30100
256*256	5.87	0.000001	102900
512*512	31.61	0.000001	339600
1024*1024	229.47	>0.000001	1000000

Диаграмма сравнения время работы CPU-one и CPU-multi

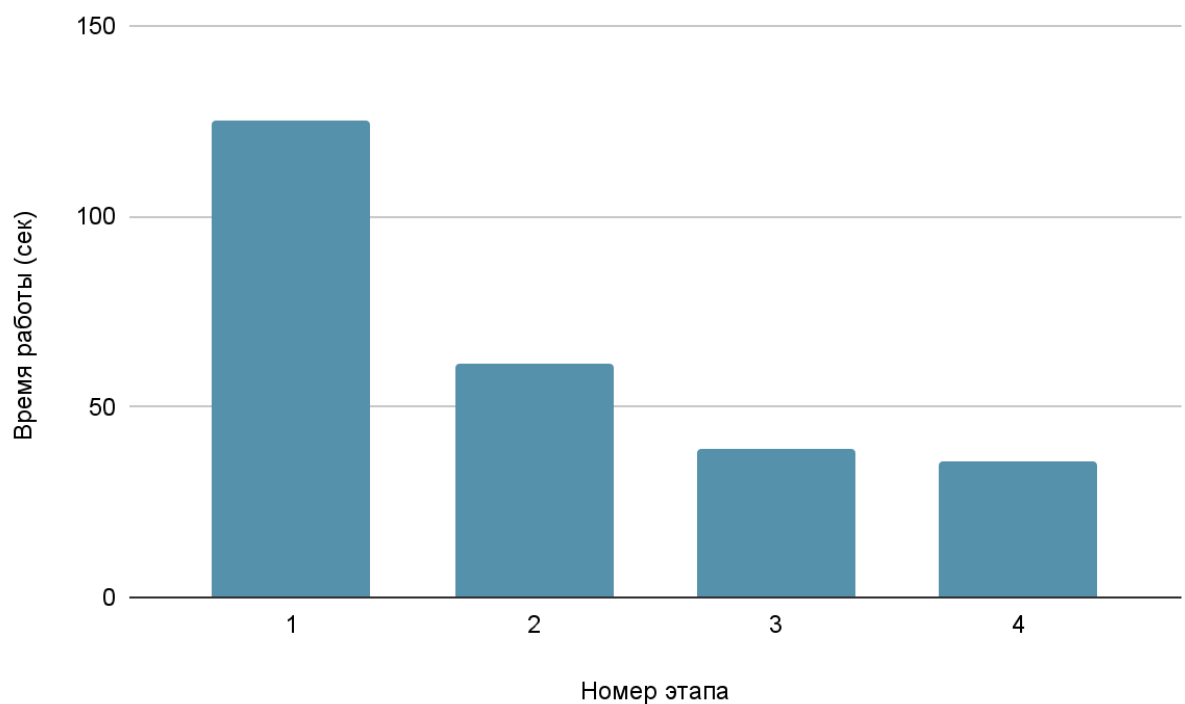


Выполнение на GPU

Этапы оптимизации на сетке 1024*1024

Этап	Время выполнения	Точность	Количество итераций	Комментарии
1	125.13	>0.000001	1000000	Неоптим-ный вариант
2	61.23	>0.000001	1000000	Swap через указатели
3	39.09	>0.000001	1000000	Расчет error каждые 100 операций
4	35.89	>0.000001	1000000	Замена распараллеливания цикла на #pragma acc parallel loop independent collapse(2) vector vector_length(1024) gang num_gangs(40)

Диаграмма оптимизации



GPU - оптимизированный вариант

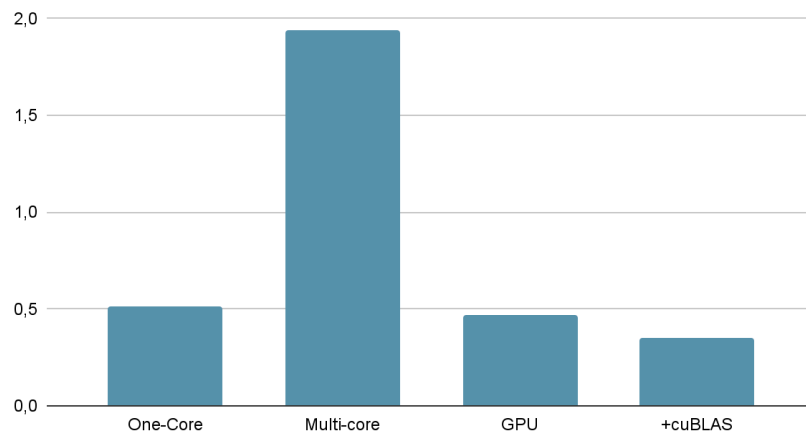
Размер сетки	Время выполнения(с)	Точность	Количество операций
128*128	0.47	0.000001	300100
256*256	1.52	0.000001	102900
512*512	5.11	0.000001	339600
1024*1024	35.89	>0.000001	1000000

+cuBLAS

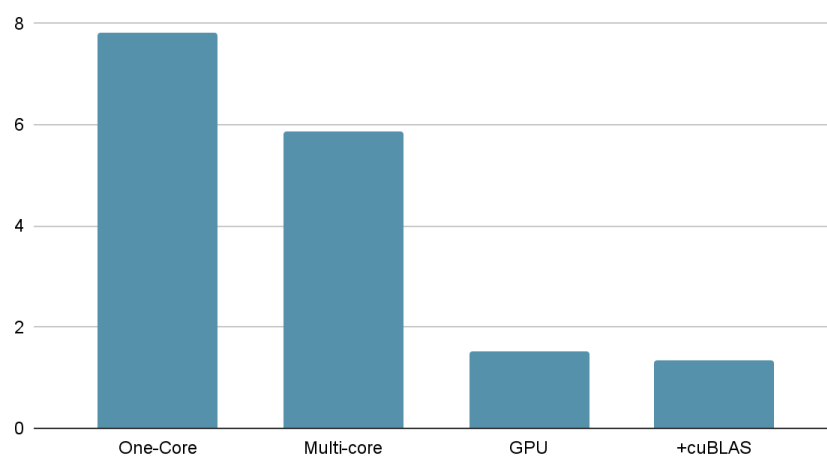
Размер сетки	Время выполнения(с)	Точность	Количество операций
128*128	0.35	0.000001	300100
256*256	1.33	0.000001	102900
512*512	4.91	0.000001	339600
1024*1024	35.81	>0.000001	1000000

Диаграммы сравнения времени работы CPU-one, CPU-multi, GPU(оптимизированный вариант)

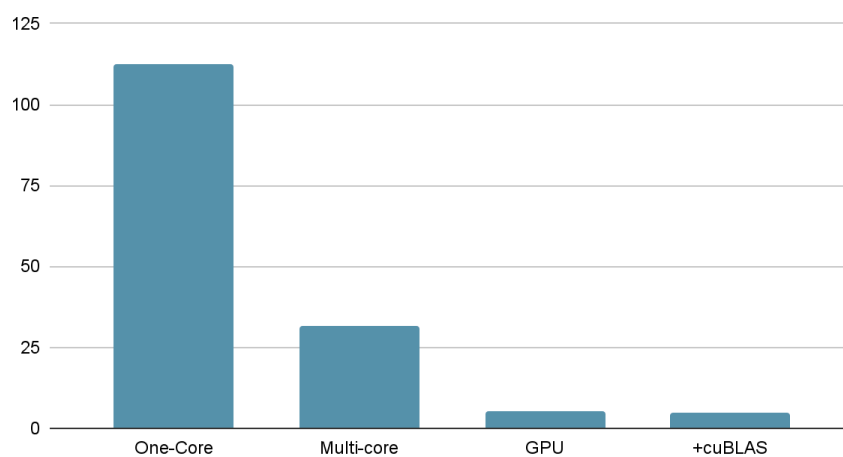
128x128

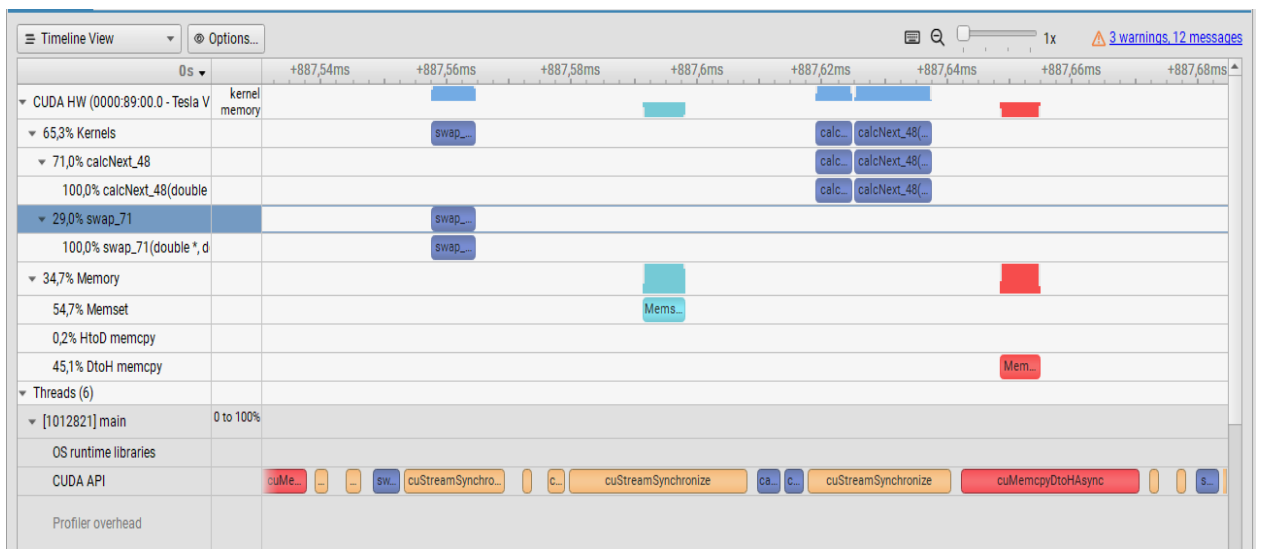
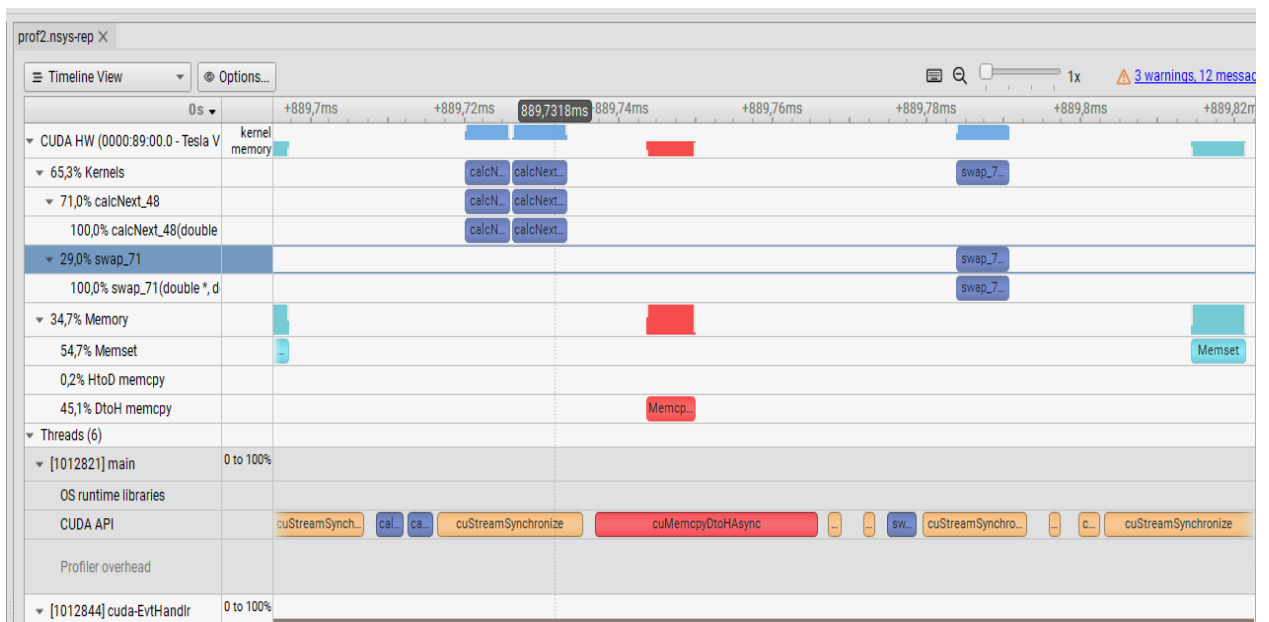


256x256



512x512





Вывод:

а) Для матриц большого размера лучше использовать оптимизированный вариант на GPU, так как он работает быстрее остальных.

Для ускорения были произведены следующие манипуляции:

1. Замена процесса обмена массивов на swar указателей
2. Расчет ошибки с определенной периодичностью, а не каждую итерацию (раз в 100 итераций)
3. Замена распараллеливания цикла следующей директивой:

```
#pragma acc parallel loop independent collapse(2) vector vector_length(n)
gang num_gangs(n)
```

б) На матрицах небольшого размера разница между директивами openACC и cuBLAS наименьшая, однако всегда лучше cuBLAS