

Теория параллелизма

Отчёт Задача №8

Выполнил 23932, Карпачев Дмитрий Александрович

21.05.2025

Цель: попробовать на практике использование cudaAPI на примере решения уравнения теплопроводности разностной схемой, протестировать время работы с использованием cuda ядер и cuda graph.

Компилятор: pgc++

Профилировщик: Nsight Systems

Замеры времени: `std::chrono::high_resolution_clock` (итоговое время в секундах)

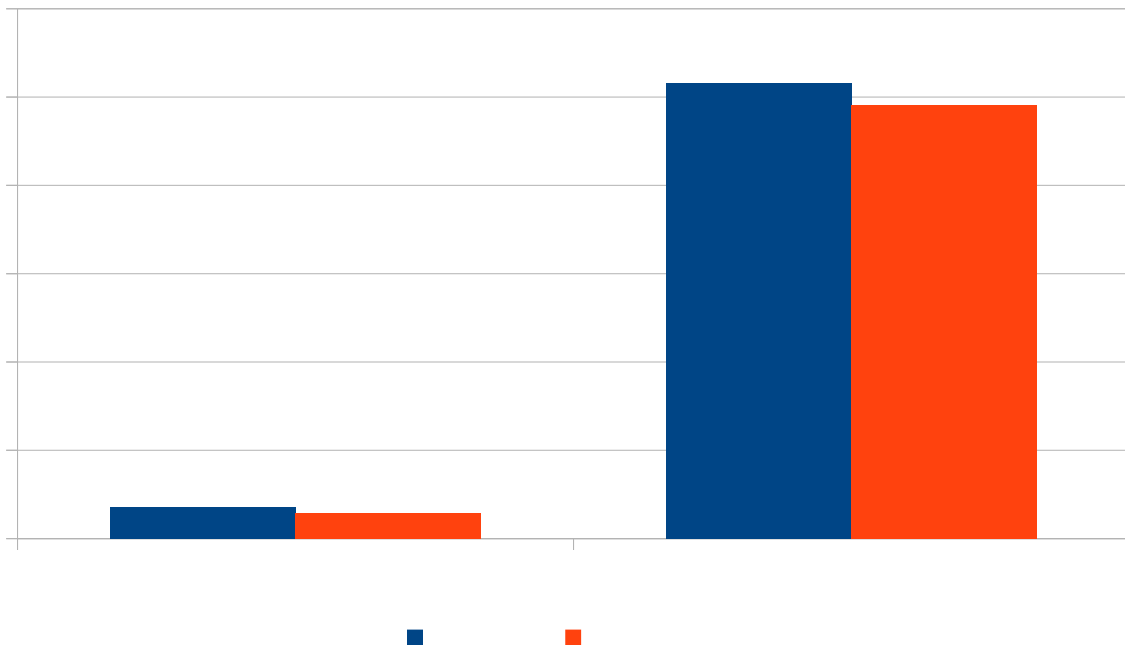
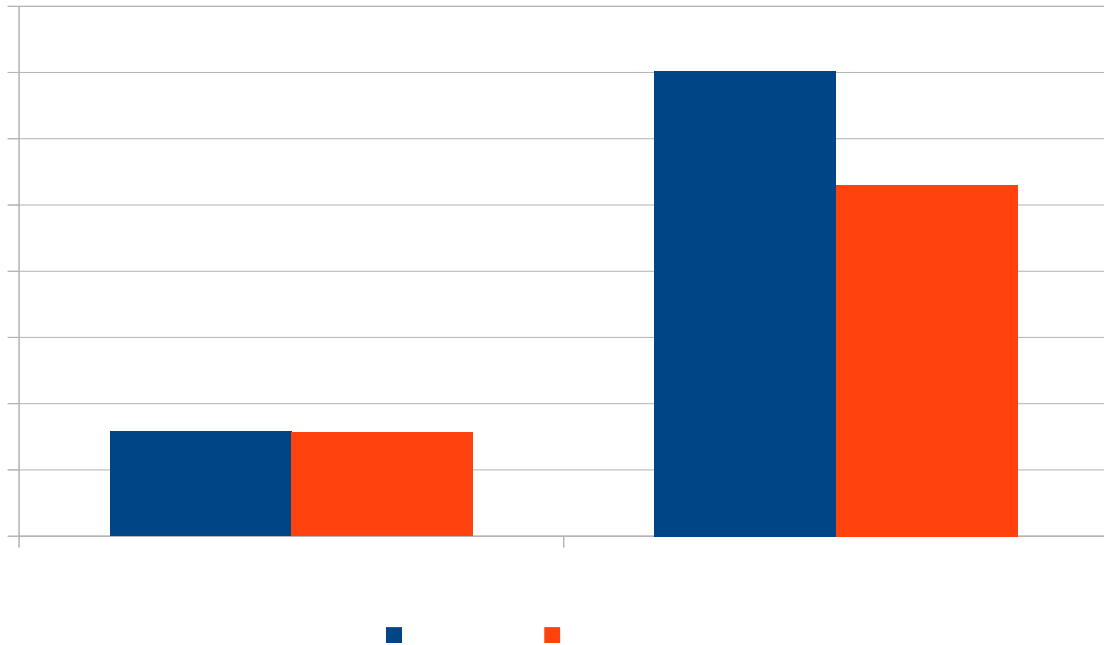
GPU – openACC + cuBLAS

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	0.079 сек.	7.53244e-07	31000
256*256	0.351 сек.	9.91241e-07	103000
512*512	1.805 сек.	9.92435e-07	340000
1024*1024	25.812 сек.	1.36929e-06	1000000

GPU - cudaAPI

Размер сетки	Время выполнения	Точность	Количество итераций
128*128	0.076 сек.	7.53244e-07	31000
256*256	0.263 сек.	9.91241e-07	103000
512*512	1.439 сек.	9.92435e-07	340000
1024*1024	24.563 сек.	1.36929e-06	1000000

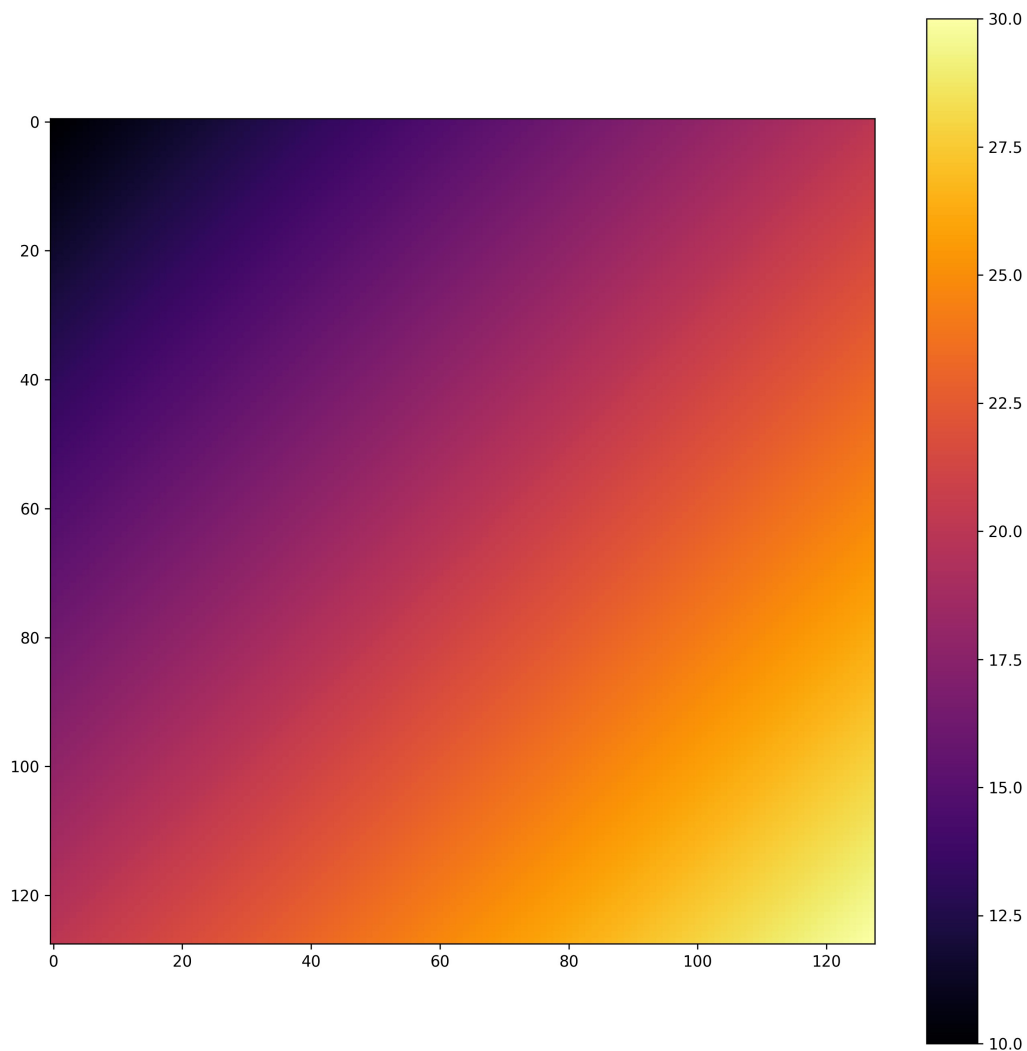
Сравнение реализаций openACC + cuBLAS и cudaAPI



Вывод матрицы 10*10

10.000	11.111	12.222	13.333	14.444	15.556	16.667	17.778	18.889	20.000
11.111	12.222	13.333	14.444	15.556	16.667	17.778	18.889	20.000	21.111
12.222	13.333	14.444	15.556	16.667	17.778	18.889	20.000	21.111	22.222
13.333	14.444	15.556	16.667	17.778	18.889	20.000	21.111	22.222	23.333
14.444	15.556	16.667	17.778	18.889	20.000	21.111	22.222	23.333	24.444
15.556	16.667	17.778	18.889	20.000	21.111	22.222	23.333	24.444	25.556
16.667	17.778	18.889	20.000	21.111	22.222	23.333	24.444	25.556	26.667
17.778	18.889	20.000	21.111	22.222	23.333	24.444	25.556	26.667	27.778
18.889	20.000	21.111	22.222	23.333	24.444	25.556	26.667	27.778	28.889
20.000	21.111	22.222	23.333	24.444	25.556	26.667	27.778	28.889	30.000

Визуализация матрицы 128*128 после расчётов



Вывод:

Использование `cuda graph` даёт эффект схожий с предыдущими реализациям на `async` и `cuBLAS`, но использование `cudaAPI` даёт больше

контроля над распараллеливанием программы и позволяет проводить дальнейшую оптимизацию для улучшения временных показателей.