

Теория параллелизма

Отчет

Уравнение теплопроводности (Разностная
схема)

Выполнила Рубан Василина Александровна, 22930

31.05.2024

Цель работы:

- Ознакомиться с OpenACC.
- Реализовать решение уравнения теплопроводности на C++ с использованием разностной схемы (пятиточечный шаблон).
- Изучить методы линейной интерполяции для задания граничных условий.
- Оптимизировать программу для работы на CPU и GPU.
- Сравнить производительность программы на CPU и GPU.
- Использовать библиотеку cuBLAS.
- Провести профилирование программы с использованием Nsight Systems.

Используемый компилятор: gpc++.

Используемый профилировщик: "Nsight Systems".

Как производились замеры времени работы: для измерения времени выполнения программы была использована библиотека `std::chrono`.

Выполнение на CPU

CPU-onecore

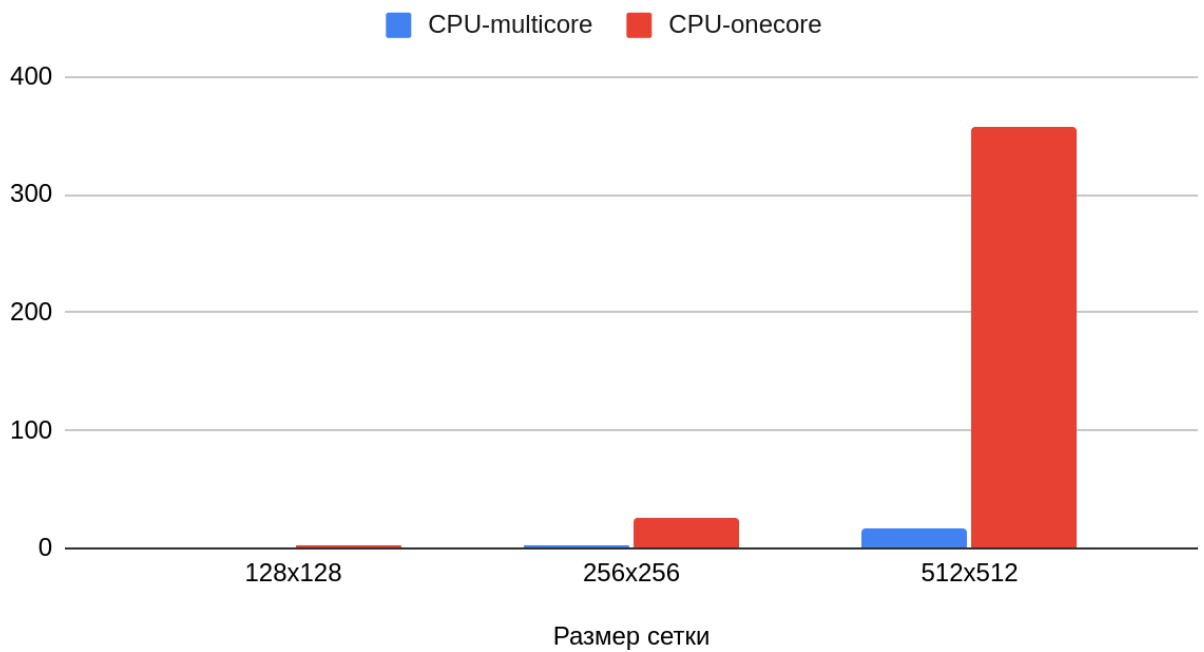
Размер сетки	Время выполнения, в секундах	Точность	Количество итераций
128x128	1.812	1e-6	36700
256x256	26.105	1e-6	125000
512x512	356.595	1e-6	409700

CPU-multicore

Размер сетки	Время выполнения, в секундах	Точность	Количество итераций
128x128	0.331	1e-6	18700
256x256	2.209	1e-6	52700
512x512	17.773	1e-6	121700

Диаграмма сравнения время работы CPU-one и CPU-multi

CPU-multicore и CPU-onecore



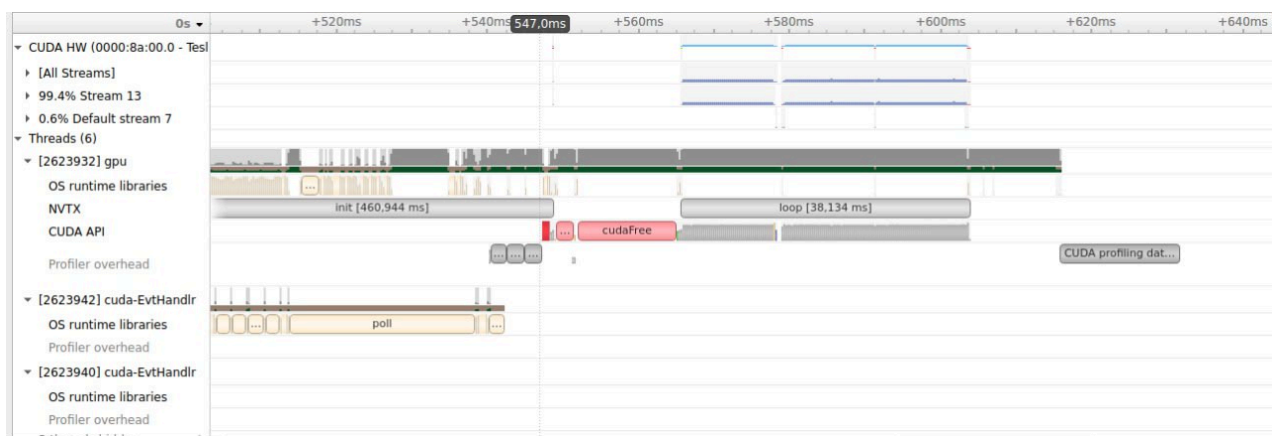
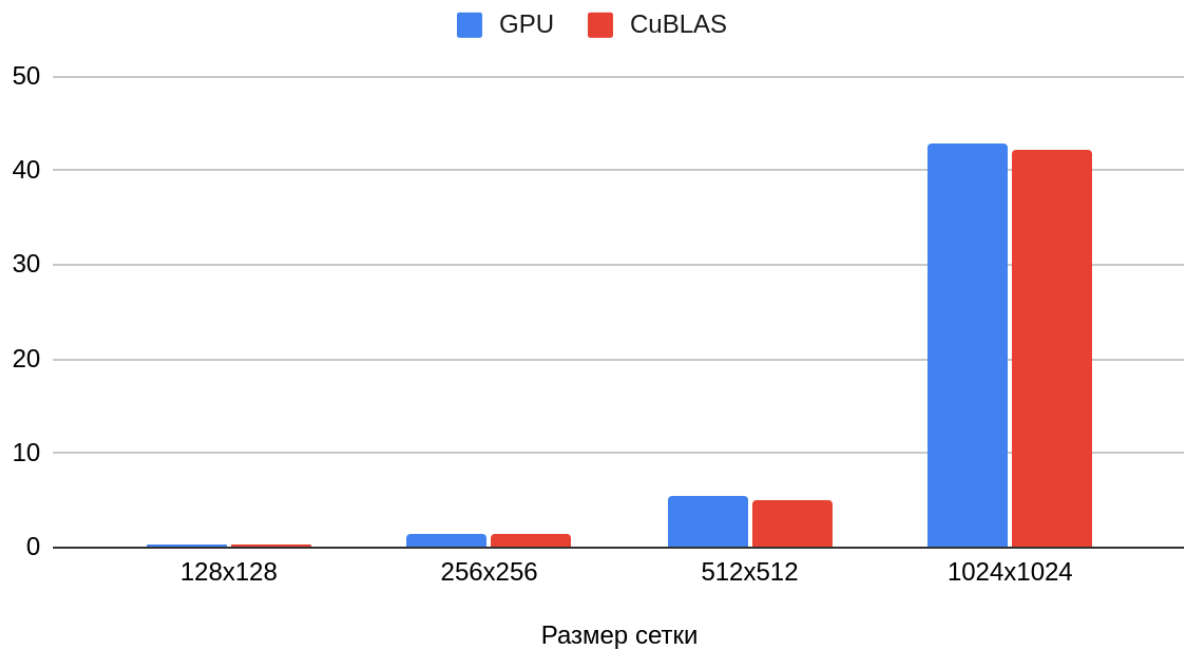
Выполнение на GPU

Размер сетки	Время выполнения, в секундах	Точность	Количество итераций
128x128	0.384	1e-6	36700
256x256	1.400	1e-6	125000
512x512	5.367	1e-6	409800
1024x1024	42.845	1e-6	1274200

Выполнение на GPU с использованием cuBLAS

Размер сетки	Точность	Время выполнения	Количество итераций
128x128	1e-06	0.377	37000
256x256	1e-06	1.39758	125000
512x512	1e-06	5.13732	410000
1024x1024	1e-06	42.141	1275000

GPU и CuBLAS



Вывод:

Использование библиотеки cuBLAS для расчета ошибки привело к небольшому ускорению.