Теория параллелизма

Отчёт

# Уравнения теплопроводности на CUDA

Выполнил Солопов Илья
Группа 21932
04.04.2023

# Введение

## Цели работы

Реализовать решение уравнения теплопроводности (пятиточечный шаблон) в двумерной области на равномерных сетках. С условиями линейной интерполяции между углами области, а также ограниченными значениями точности и максимального числа итераций.

На вход программе через командную строку должны подаваться параметры: точность, размер сетки, количество итераций.

Вывод программы – количество итераций и достигнутое значение ошибки.

Перенести программу на GPU используя CUDA. Операцию редукции (вычисление максимального значения ошибки) реализовать с использованием библиотеки CUB.

Сравнить скорость работы для разных размеров сеток на центральном и графическом процессоре (реализация с CUDA и без).

Произвести профилирование программы и оптимизацию кода.

## Используемый компилятор

Использовался компилятор nvcc.

## Используемый профилировщик

Использовался профилировщик NVIDIA NsightSystems.

## Замер времени работы

Замер времени работы программы производился при помощи библиотеки chrono языка c++.

# Выполнение на CPU

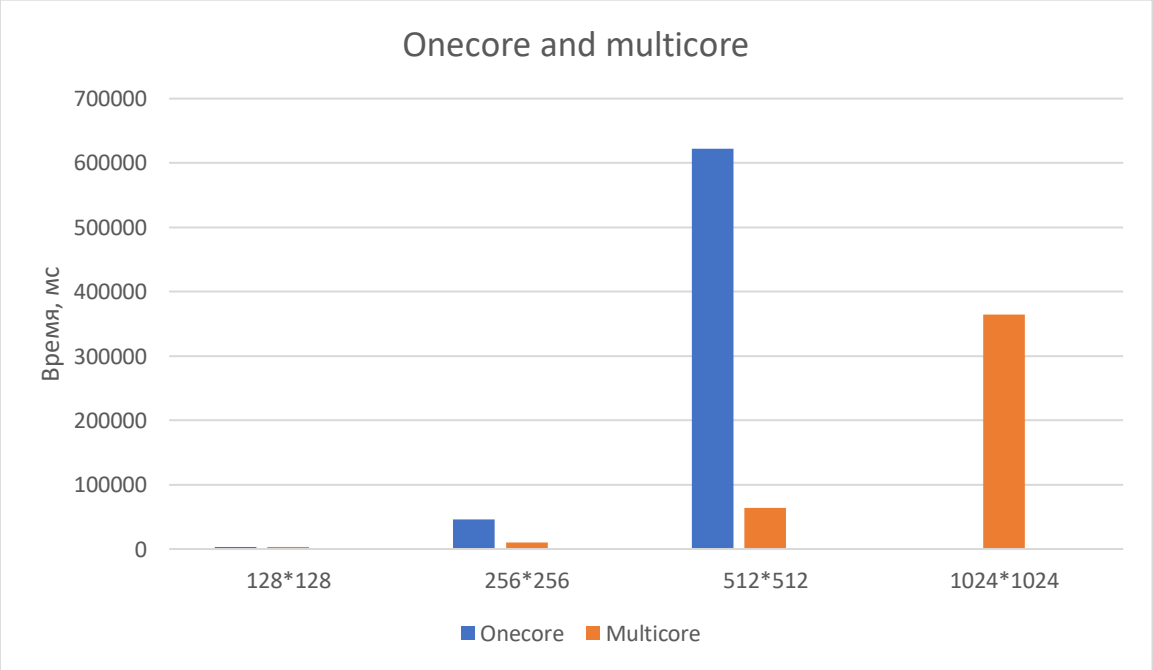
## CPU-onecore

| Размер сетки | Время выполнения, мс | Точность | Количество итераций |
|---|---|---|---|
| 128*128 | 3378.5 | 9.97841e-07 | 30081 |
| 256*256 | 46056.25 | 9.97808e-07 | 102913 |
| 512*512 | 622527.5 | 9.93016e-07 | 339969 |

## CPU-multicore

| Размер сетки | Время выполнения, мс | Точность | Количество итераций |
|---|---|---|---|
| 128*128 | 3431.83 | 9.97841e-07 | 30081 |
| 256*256 | 10365.57 | 9.97808e-07 | 102913 |
| 512*512 | 63836 | 9.93016e-07 | 339969 |
| 1024*1024 | 364049 | 1.373e-06 | 1000000 |

## Диаграмма сравнения времени работы




.

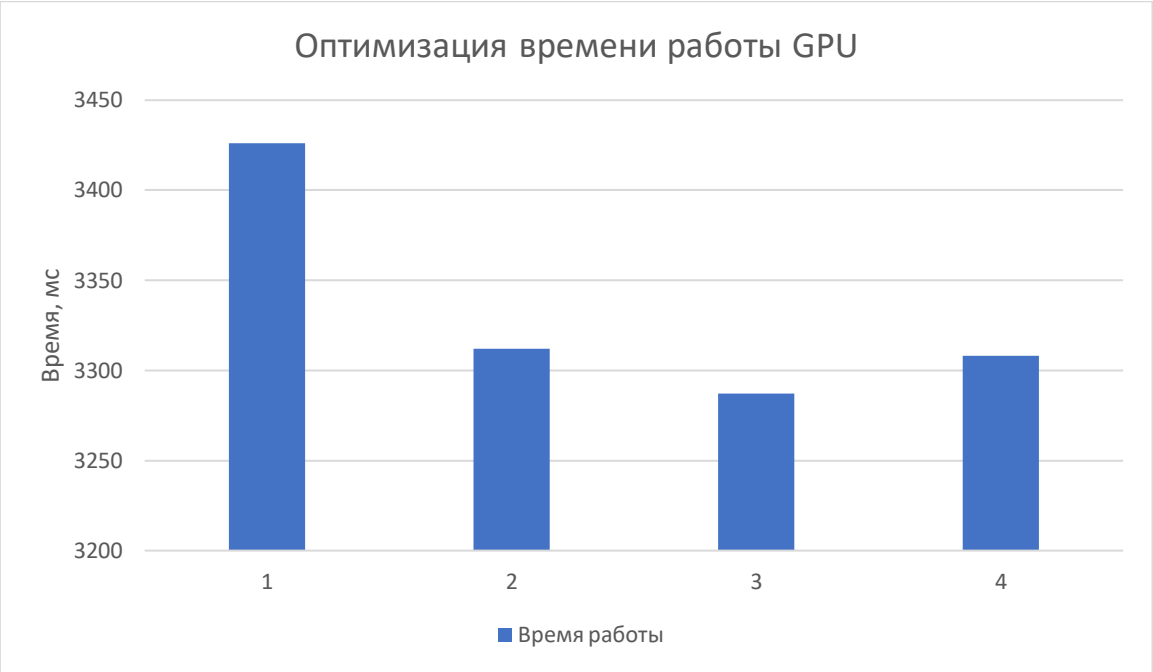
# Выполнение на GPU

## Этапы оптимизации

Оптимизация проводилась на сетке размером 512x512. Количество итераций при профилировании 100.

| Этап № | Время выполнения, мс | Точность | Количество итераций | Комментарии (что было сделано) |
|---|---|---|---|---|
| 1 | 3426 | 0.0207282 | 1000 | Код предыдущего задания (только кублас) |
| 2 | 3312 | 0.0105525 | 1000 | Код на CUDA. Вычисления значения ошибки каждую итерацию. |
| 3 | 3287 | 0.0206635 | 1000 | Вычисление ошибки не на каждой итерации. |
| 4 | 3308 | 0.0206635 | 1000 | Использование CUDA graph. |

## Диаграмма оптимизации



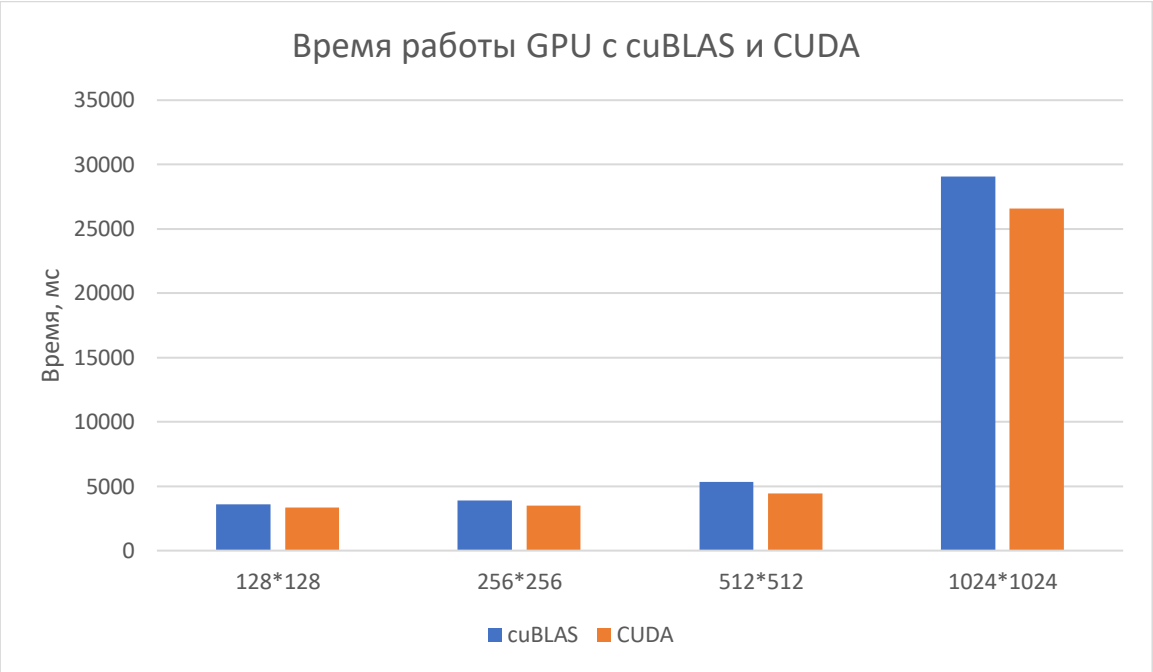

**GPU – вариант с использованием cuBLAS**

| Размер сетки | Время выполнения, мс | Точность | Количество итераций |
|---|---|---|---|
| 128*128 | 3608 | 9.97841e-07 | 30081 |
| 256*256 | 3901 | 9.97808e-07 | 102913 |
| 512*512 | 5340 | 9.93016e-07 | 339969 |
| 1024*1024 | 29048 | 1.373e-06 | 1000000 |

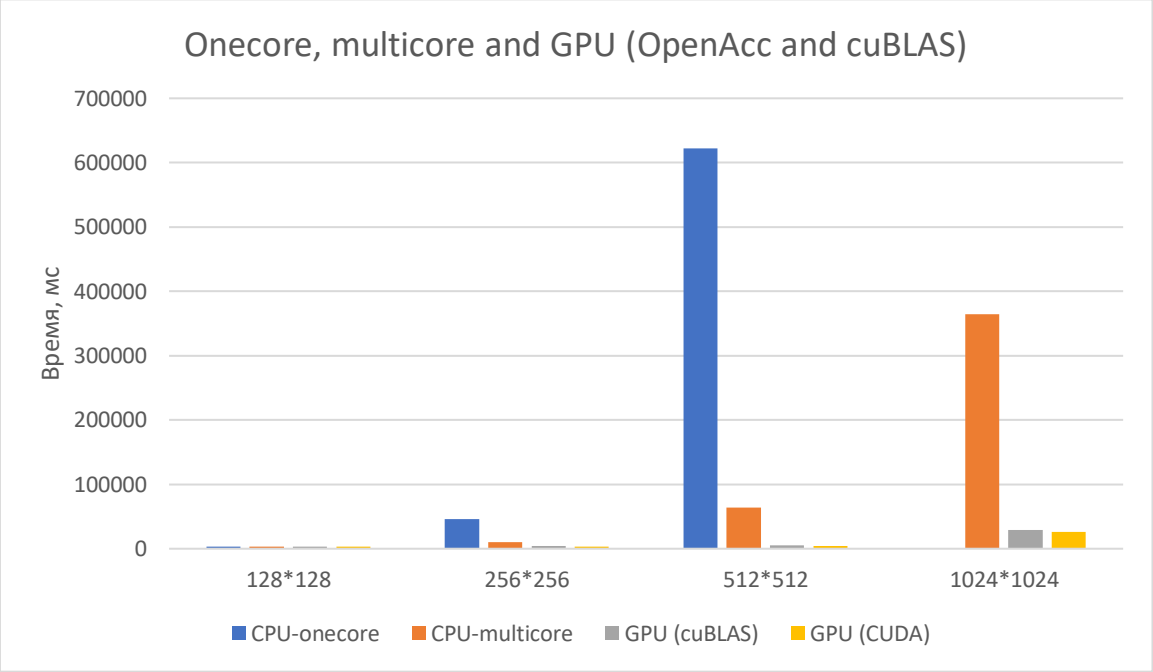
**GPU – вариант с использованием CUDA**

| Размер сетки | Время выполнения, мс | Точность | Количество итераций |
|---|---|---|---|
| 128*128 | 3355 | 9.97535e-07 | 30081 |
| 256*256 | 3518 | 9.97732e-07 | 102913 |
| 512*512 | 4464 | 9.92997e-07 | 339969 |
| 1024*1024 | 26561 | 1.373e-06 | 1000000 |

**Диаграмма сравнения времени работы**




## Диаграмма сравнения времени работы



## Вывод

Полученные результаты говорят нам о том, что использование CUDA благотворно влияет на производительность программы. CUDA даёт прирост на сетках всех размерностей за счёт того, что уменьшается задержка между запусками ядер. Чем больше размерность сетки, тем быстрее выполнение.

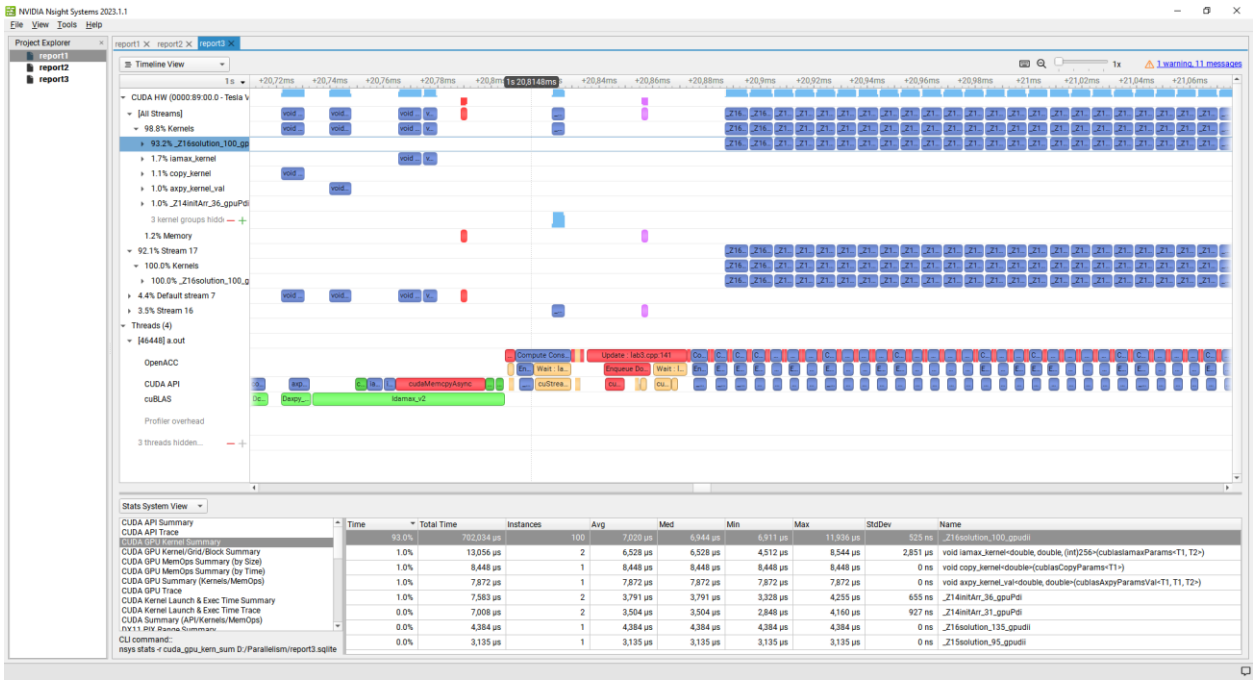
# Приложение

## Ссылка на репозиторий

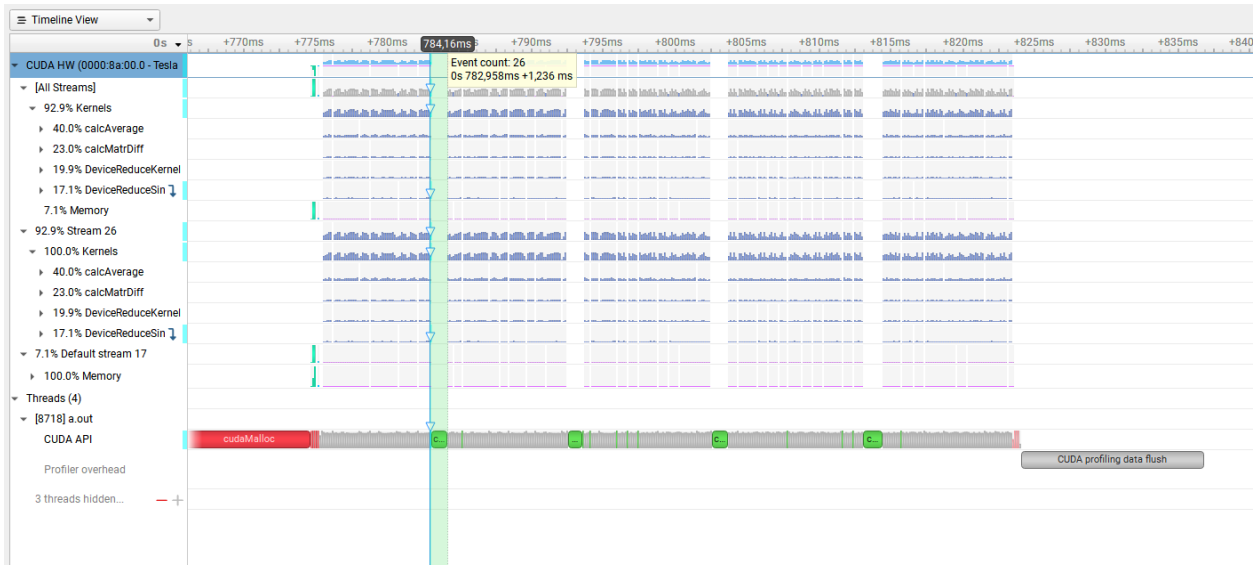
https://github.com/IlS0/Parallelism-tasks

## Скриншоты из профилировщика

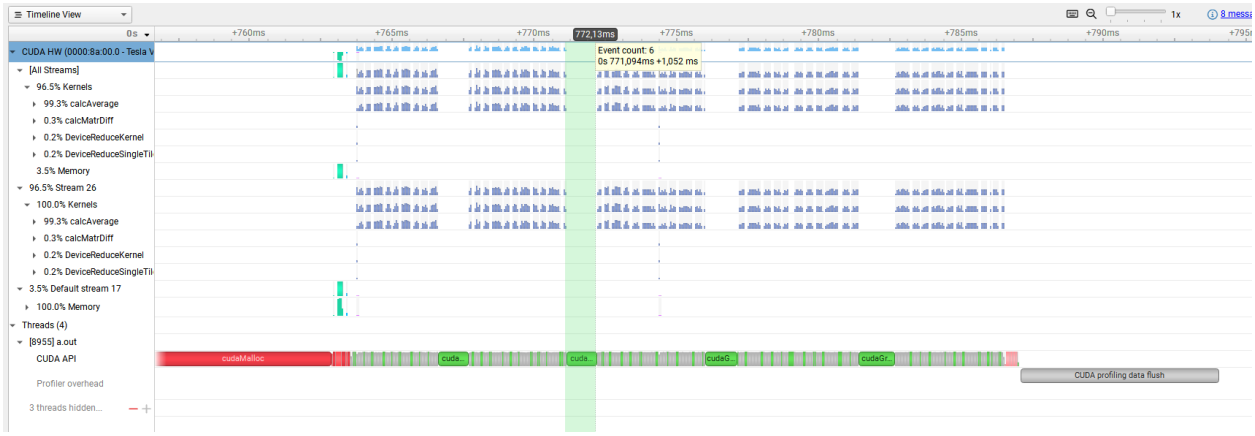
Этап 1



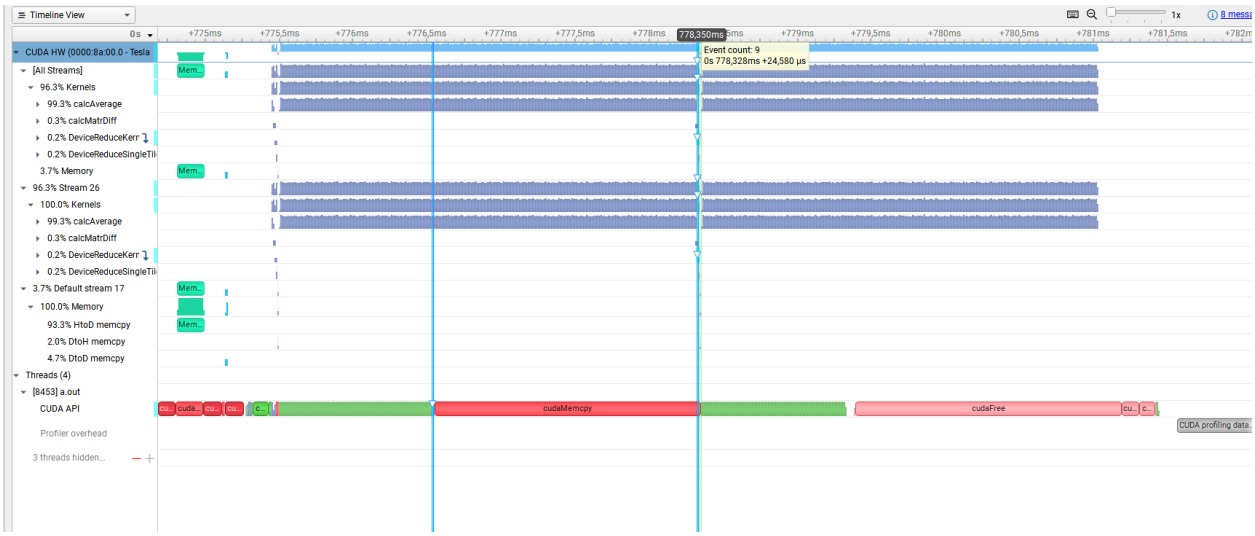
Этап 2




Этап 3


Timeline View
1x
0s
+760ms
+765ms
+770ms
772,13ms
+775ms
+780ms
+785ms
+790ms
CUDA HW (0000:8a:00.0 - Tesla V
[All Streams]
96.5% Kernels
99.3% calcAverage
0.3% calcMatrDiff
0.2% DeviceReduceKernel
0.2% DeviceReduceSingleTil
3.5% Memory
96.5% Stream 26
100.0% Kernels
99.3% calcAverage
0.3% calcMatrDiff
0.2% DeviceReduceKernel
0.2% DeviceReduceSingleTil
3.5% Default stream 17
100.0% Memory
Threads (4)
[8955] a.out
CUDA API
Profiler overhead
3 threads hidden...
Event count: 6
0s 771,094ms +1,052 ms
cudaMalloc
cuda...
cuda...
cudaG...
cudaGr...
CUDA profiling data flush

# Этап 4


Timeline View
1x
0s
+775ms
+775,5ms
+776ms
+776,5ms
+777ms
+777,5ms
+778ms
778,350ms
+779ms
+779,5ms
+780ms
+780,5ms
+781ms
+781,5ms
CUDA HW (0000:8a:00.0 - Tesla
[All Streams]
96.3% Kernels
99.3% calcAverage
0.3% calcMatrDiff
0.2% DeviceReduceKerr
0.2% DeviceReduceSingleTil
3.7% Memory
96.3% Stream 26
100.0% Kernels
99.3% calcAverage
0.3% calcMatrDiff
0.2% DeviceReduceKerr
0.2% DeviceReduceSingleTil
3.7% Default stream 17
100.0% Memory
93.3% HtoD memcpy
2.0% DtoH memcpy
4.7% DtoD memcpy
Threads (4)
[8453] a.out
CUDA API
Profiler overhead
3 threads hidden...
Event count: 9
0s 778,328ms +24,580 µs
cudaMemcpy
cudaFree
CUDA profiling data