

Теория параллелизма

Отчёт

Оптимизированные библиотеки

Выполнил Грищенко Александр Михайлович, 21932

03.2023

1 Цели работы

Реализовать решение уравнение теплопроводности (пятиточечный шаблон) в двумерной области на равномерных сетках.

Перенести программу на GPU используя директивы OpenACC.

Операцию редукции на графическом процессоре реализовать через вызовы функций из библиотеки cuBLAS.

Произвести профилирование программы и оптимизацию кода.

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

2 Используемый компилятор

pgc++ с флагом -Mcudalib=cublas

3 Используемый профилировщик

nsys (NVIDIA Nsight Systems) с флагом -trace=cublas,openacc,nvtx.

4 Как проводился замер времени работы

Для замера времени работы использовалась библиотека chrono.

Замер времени производился несколько раз, затем бралось среднее время.

5 Выполнение на CPU

Данные из предыдущего задания.

5.1 CPU-onecore

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.1	9.5e-07	11136
256*256	1.8	9.8e-07	37376
512*512	25	9.8e-07	120832

5.2 CPU-multicore

Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.5	9.5e-07	11136
256*256	3.5	9.8e-07	37376
512*512	20	9.8e-07	120832
1024*1024	145	9.89e-07	365568

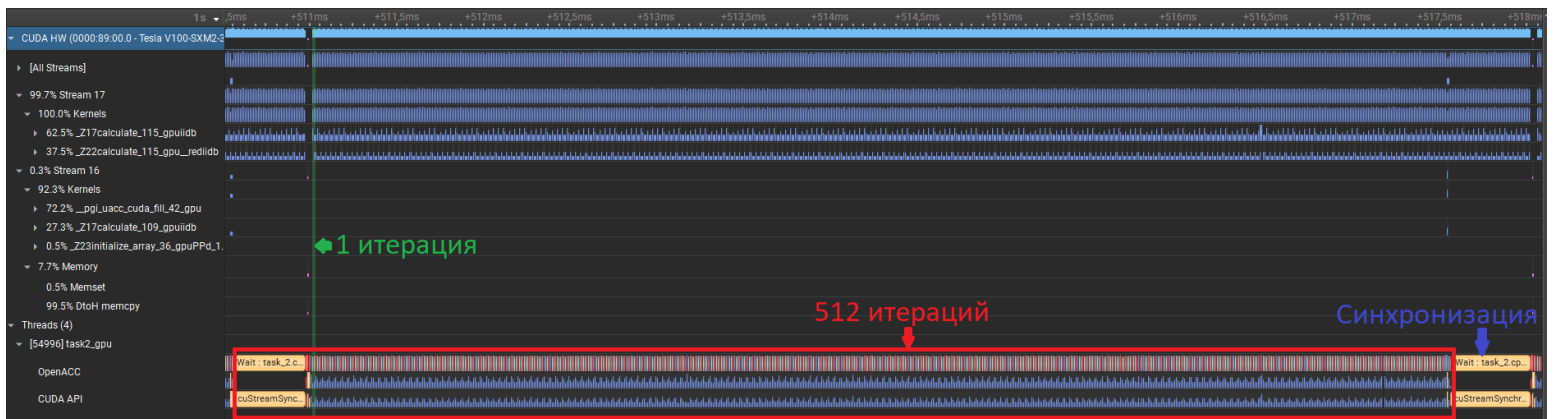
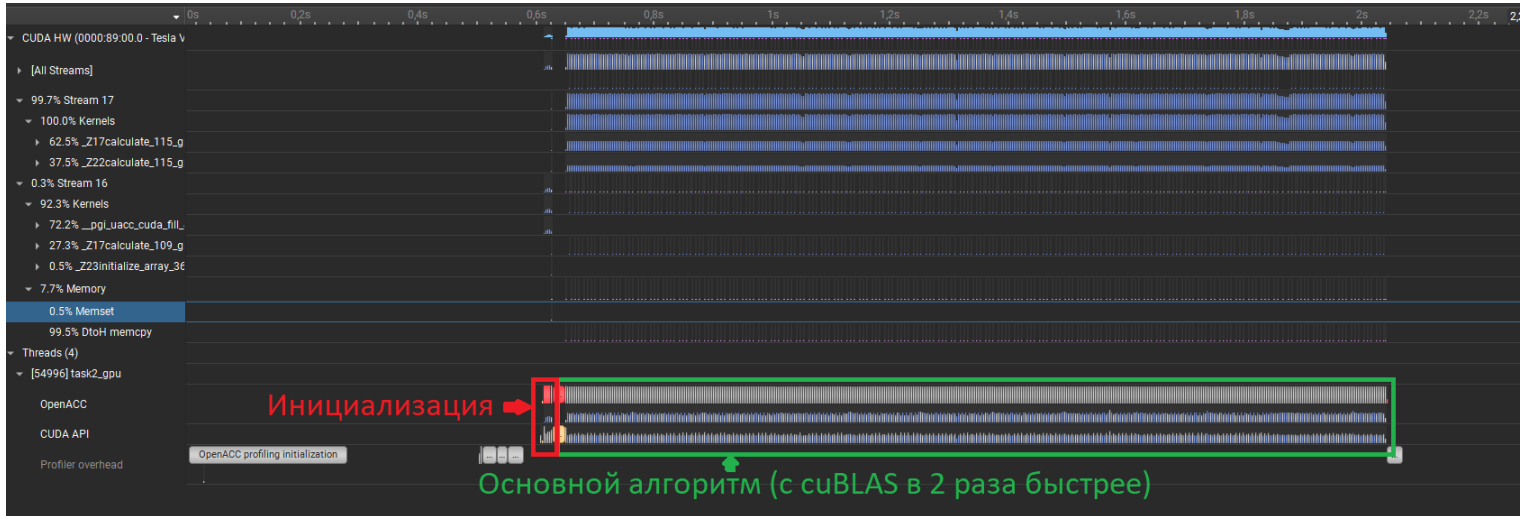
6 Выполнение на GPU

6.1 Этапы оптимизации на сетке 512*512 (количество итераций при профилировании 100 000)

100 итераций (как в прошлом задании) недостаточно, чтобы увидеть преимущество cuBLAS, поскольку перед запуском основного алгоритма происходит долгое создание handle, которое несравнимо с просчетом 100 итераций. Преимущество cuBLAS на большом количестве итераций (больших размерах сетки, в том числе), поэтому было взято 100 000.

Этап №	Время выполнения, с	Точность	Количество операций	Комментарии (что было сделано)
0	1.3	2.6e-06	1e5	Код из предыдущего задания (переделан на одномерные матрицы).
0	5.1	2.6e-06	1e5	Код из предыдущего задания (ошибка считается на каждой итерации).
1	11	2.6e-06	1e5	Вычисление максимальной ошибки через функции из cuBLAS
2	1.1	2.6e-06	1e5	Ошибка считается не каждую итерацию, асинхронность

Этап 0 (без оптимизаций)

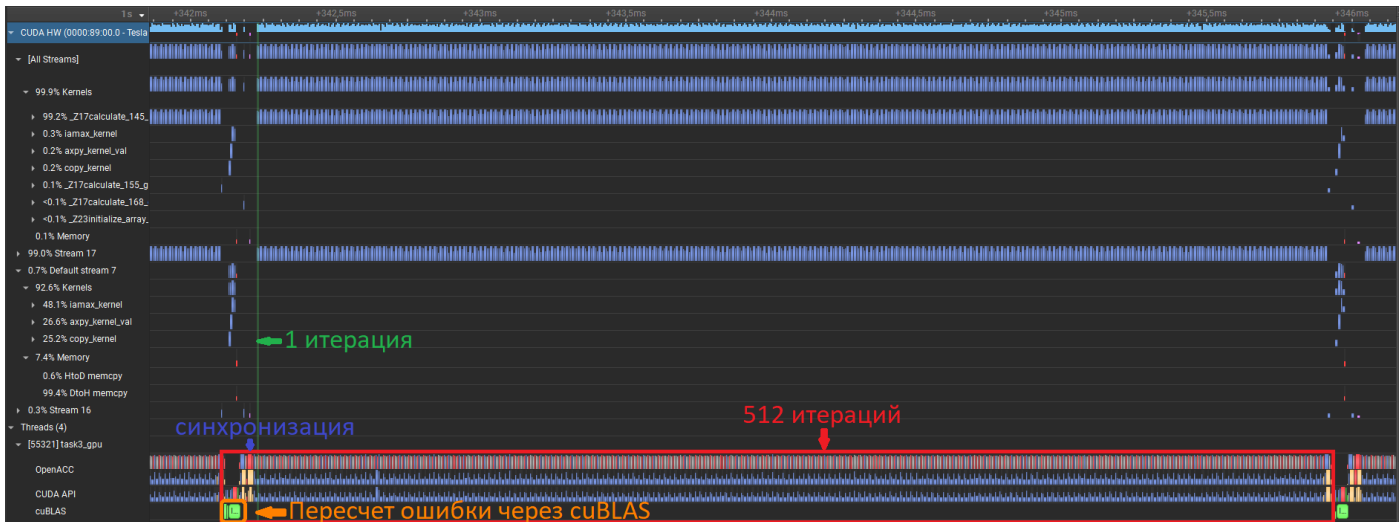
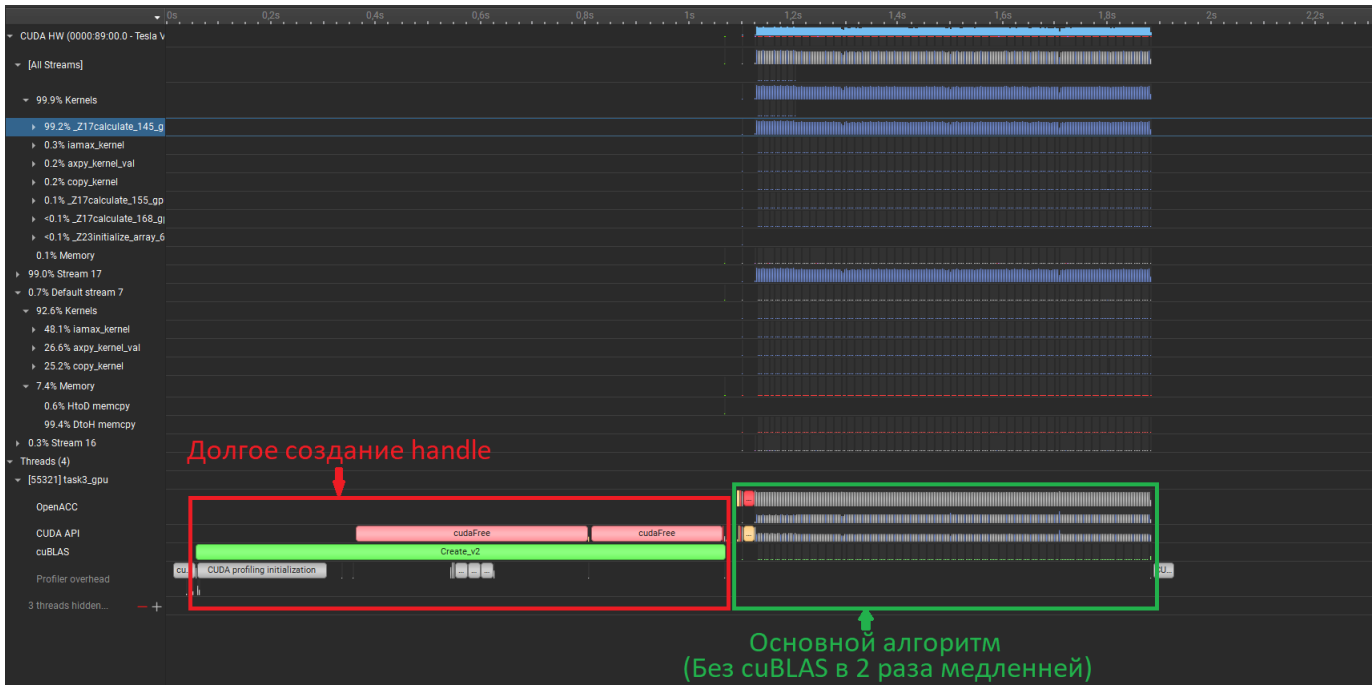


Этап 1

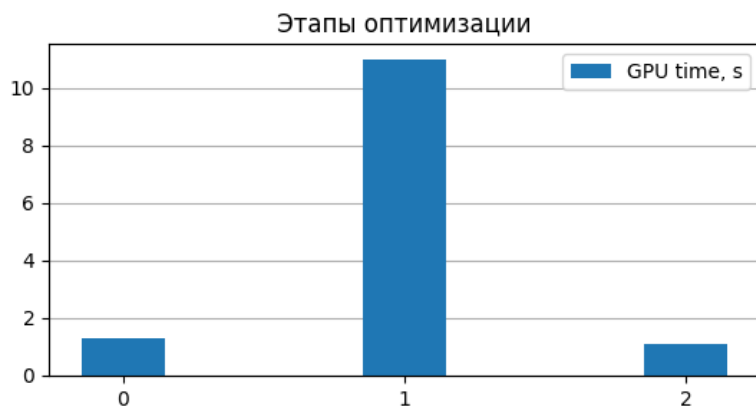


Этап 2

Поскольку ошибка считается через операции с матрицами, которые хорошо оптимизированы для GPU, а не через reduction (которая для каждой переменной создает локальные копии в каждом потоке), программа с cuBLAS работает быстрее, чем без него.



6.2 Диаграмма оптимизации (по горизонтали номер этапа; по вертикали время работы)



6.3 GPU – оптимизированный вариант (без cuBLAS)

Размер сетки	Время выполнения, с	Точность	Количество опреаций
128*128	0.3	9.5e-07	11136
256*256	0.5	9.8e-07	37376
512*512	1.5	9.8e-07	120832
1024*1024	16.7	9.9e-07	365568

6.4 GPU – оптимизированный вариант (cuBLAS)

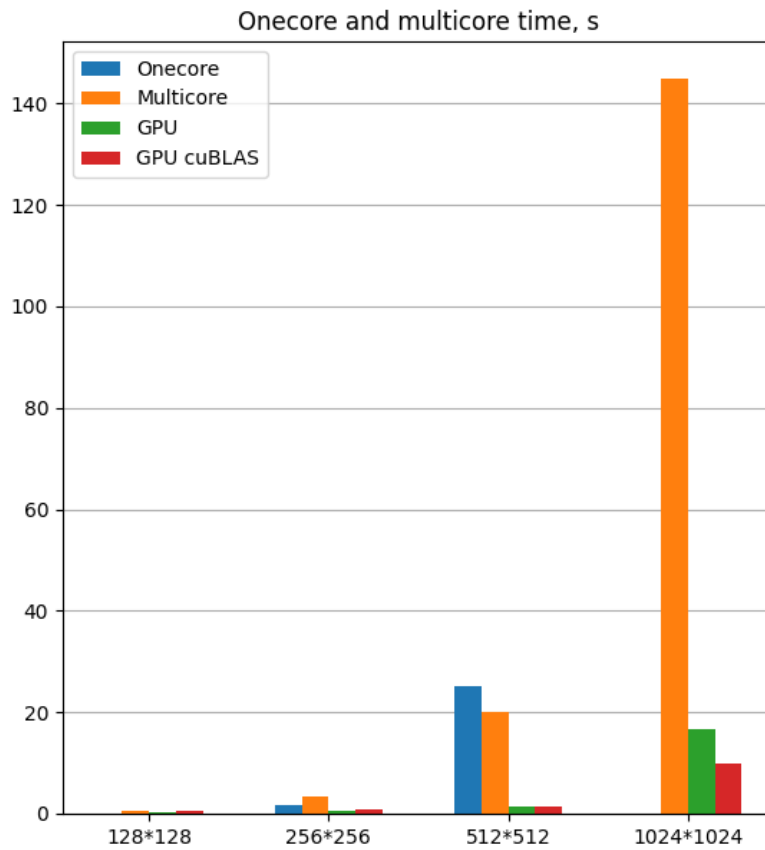
Размер сетки	Время выполнения, с	Точность	Количество опреаций
128*128	0.7	9.5e-07	11136
256*256	0.8	9.8e-07	37376
512*512	1.3	9.8e-07	120832
1024*1024	10	9.9e-07	365568

6.5 Скриншот массива 15*15 после заполнения границ и после работы всей программы

```
--Borders--
10.00 10.71 11.43 12.14 12.86 13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00
10.71 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.71
11.43 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 21.43
12.14 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 22.14
12.86 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 22.86
13.57 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 23.57
14.29 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 24.29
15.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 25.00
15.71 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 25.71
16.43 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 26.43
17.14 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 27.14
17.86 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 27.86
18.57 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 28.57
19.29 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 20.00 29.29
20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43 27.14 27.86 28.57 29.29 30.00

--Result--
10.00 10.71 11.43 12.14 12.86 13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00
10.71 11.43 12.14 12.86 13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71
11.43 12.14 12.86 13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43
12.14 12.86 13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14
12.86 13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86
13.57 14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57
14.29 15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29
15.00 15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00
15.71 16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71
16.43 17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43
17.14 17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43 27.14
17.86 18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43 27.14 27.86
18.57 19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43 27.14 27.86 28.57
19.29 20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43 27.14 27.86 28.57 29.29
20.00 20.71 21.43 22.14 22.86 23.57 24.29 25.00 25.71 26.43 27.14 27.86 28.57 29.29 30.00
```


7 Диаграмма сравнения времени работы CPU-one, CPU-multi, GPU, GPU cuBLAS для разных размеров сеток



8 Вывод

Используя библиотеку cuBLAS можно достичь прироста производительности на больших сетках. Для небольших сеток нет смысла использовать библиотеку, поскольку создание handle и работа с памятью занимают существенное время.

9 Приложение

9.1 Ссылка на GitHub

https://github.com/busyhedg03/ParallelismTheory/tree/master/task_3