

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

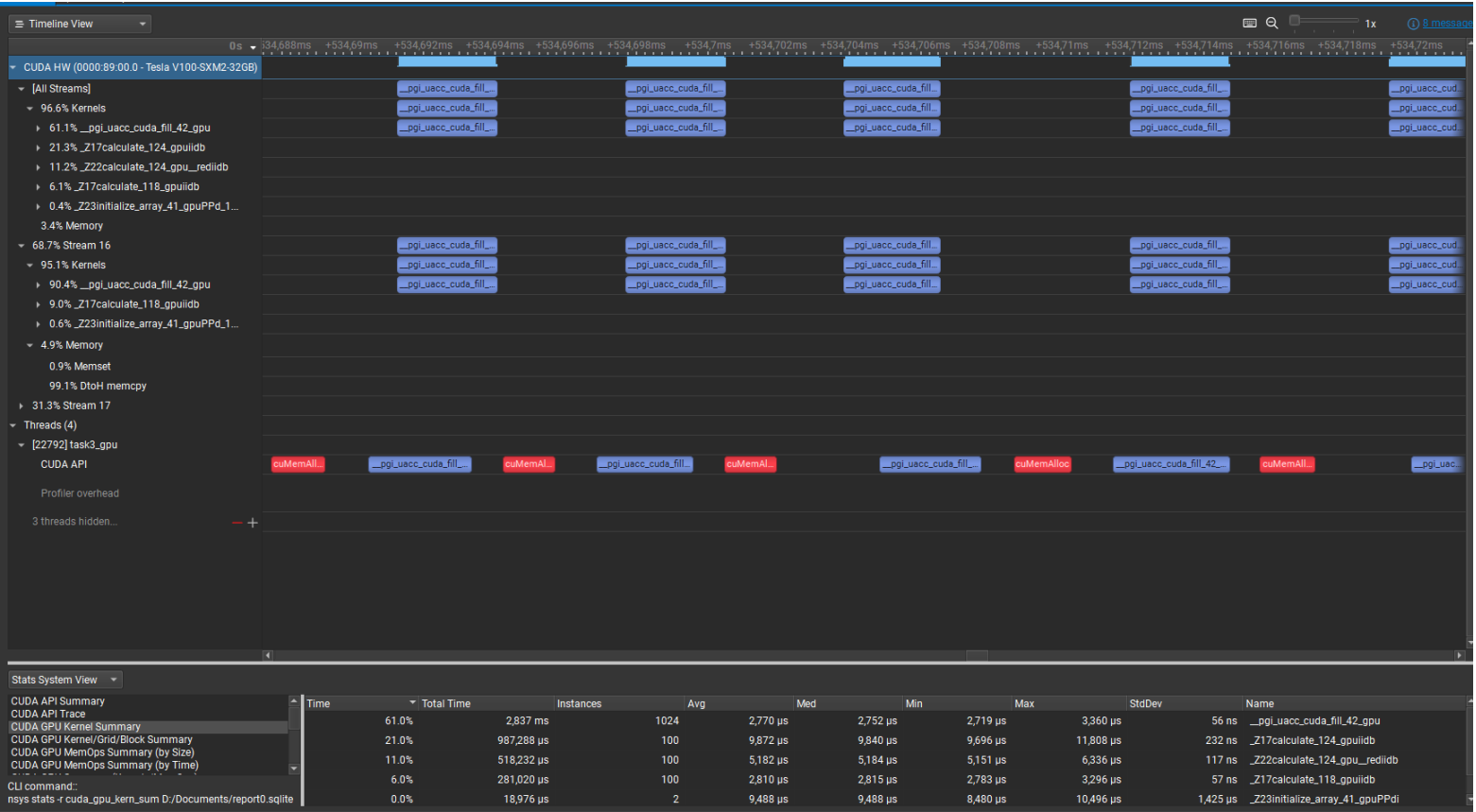
Размер сетки	Время выполнения, с	Точность	Количество операций
128*128	0.5	9.5e-07	11136
256*256	3.5	9.8e-07	37376
512*512	20	9.8e-07	120832
1024*1024	145	9.89e-07	365568

6 Выполнение на GPU

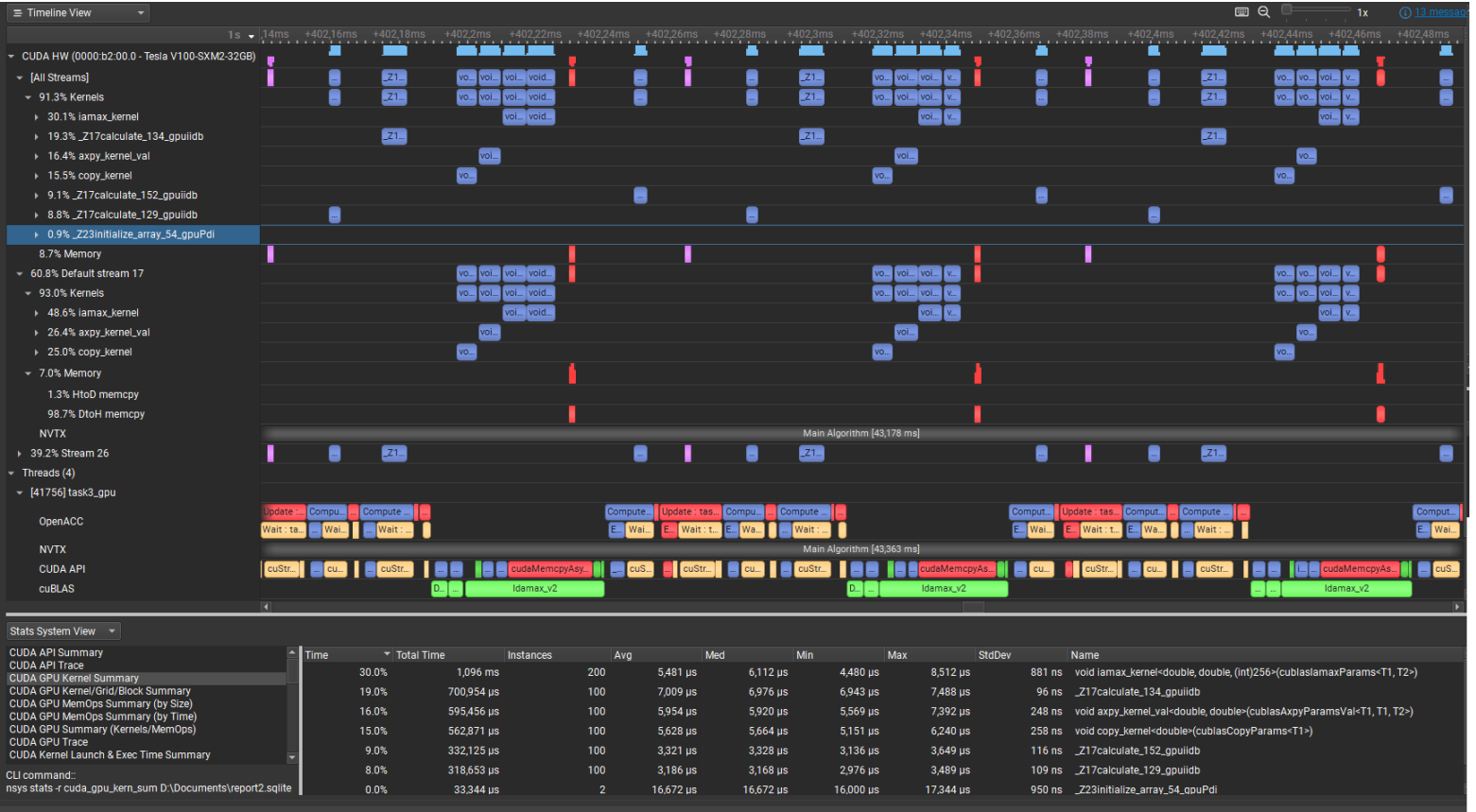
6.1 Этапы оптимизации на сетке 512*512 (количество итераций при профилировании 100)

Этап №	Время выполнения, с	Точность	Количество операций	Комментарии (что было сделано)
0	0.23	N/A*	100	Код из предыдущего задания.
0	0.22	0.035	100	Код из предыдущего задания (ошибка считается на каждой итерации, одномерные матрицы).
1	0.6	0.035	100	Вычисление максимальной ошибки через функции из cuBLAS
2	0.6	N/A*	100	Ошибка считается не каждую итерацию, асинхронность

* Из-за того, что период пересчёта ошибки намного меньше размера сетки, нет возможности определить точность на сотой итерации.



Этап 0 (без оптимизаций)



Этап 1

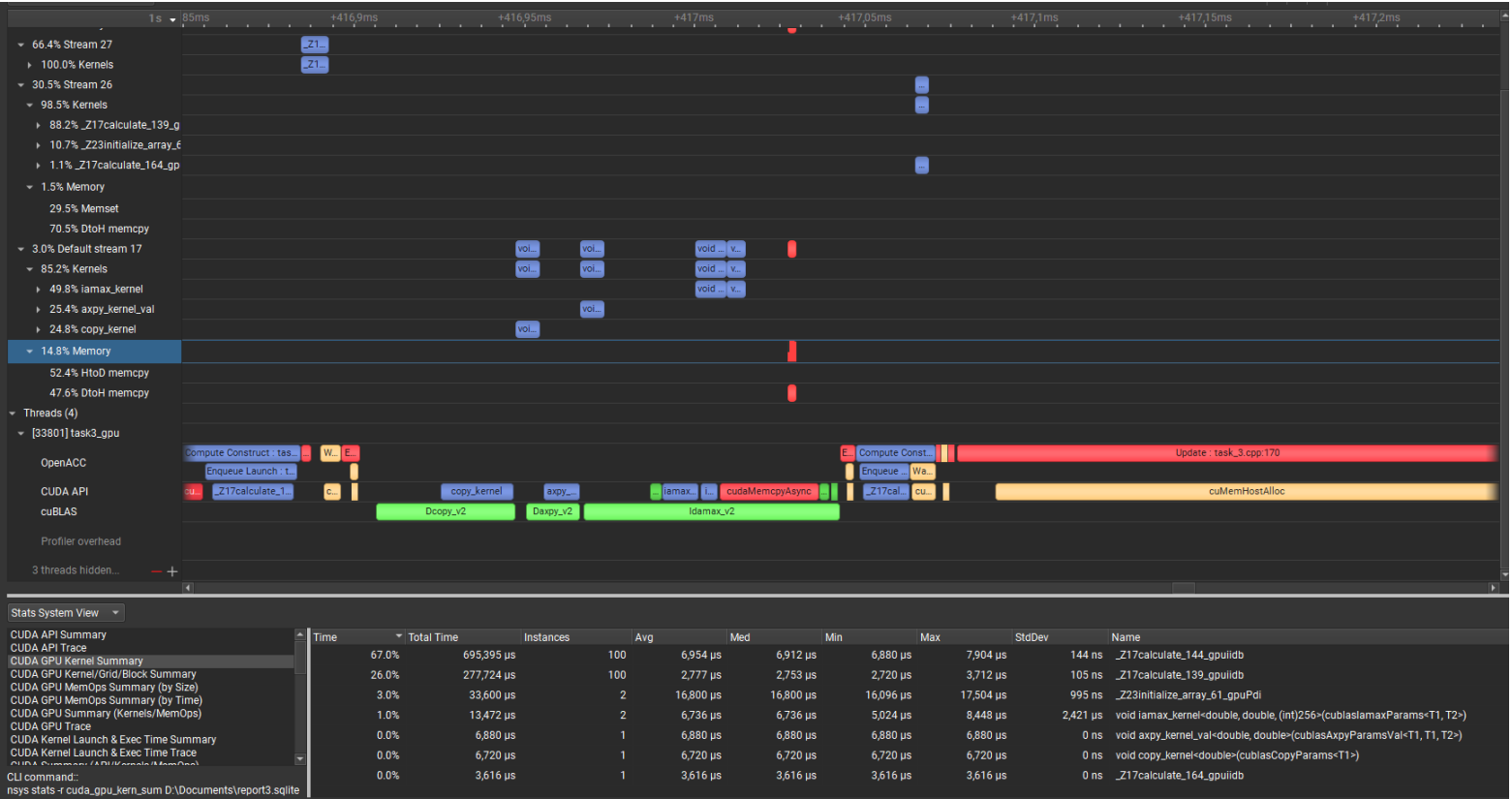
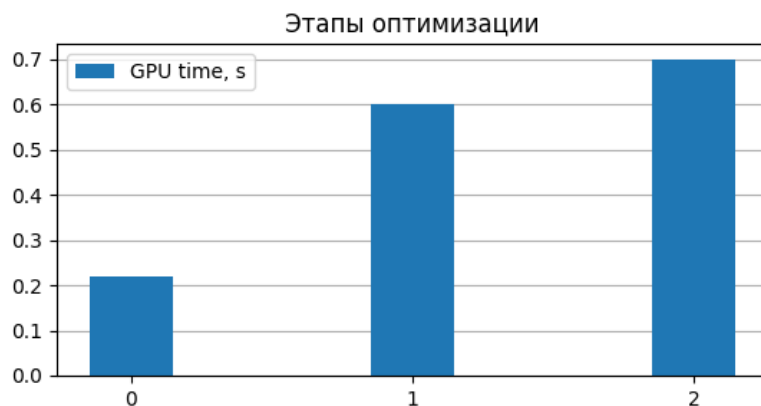


Figure 2

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



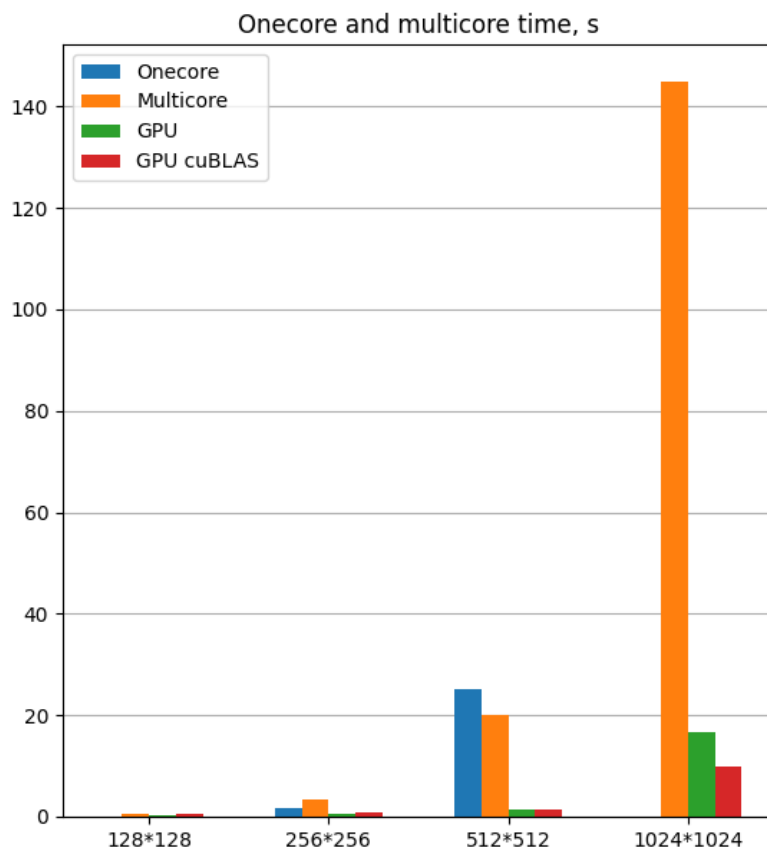
6.3 GPU – оптимизированный вариант (без cuBLAS)

Размер сетки	Время выполнения, с	Точность	Количество операций
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 операций
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

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