

# Теория параллелизма

## Отчёт

### Уравнение теплопроводности

Выполнил: Лейсле Александр Геннадьевич, группа 21932

08.03.2023

Цели работы:

Выполнить расчёт уравнения теплопроводности на:

1. одном ядре CPU
2. нескольких ядрах CPU
3. GPU

Выполнить сравнение времени выполнения программы с различными входными параметрами для каждого из указанных устройств для вычисления. Произвести оптимизацию кода. Провести профилирование программы с помощью Nsight System.

Используемый компилятор: `pgc++`

Используемый профилировщик: `nsys`

Как производили замер времени работы:

В начале и в конце программы производилась фиксация текущего времени с использованием команд из `chrono`, разница этого времени выводилась в стандартный поток вывода.

### CPU-onecore

Размер сетки	Время выполнения	Точность	Количество итераций
128x128	1.93с	$9.9 * 10^{-7}$	30074
256x256	27.18с	$9.9 * 10^{-7}$	102885
512x512	119.22с	$9.9 * 10^{-7}$	339599

### CPU-multicore

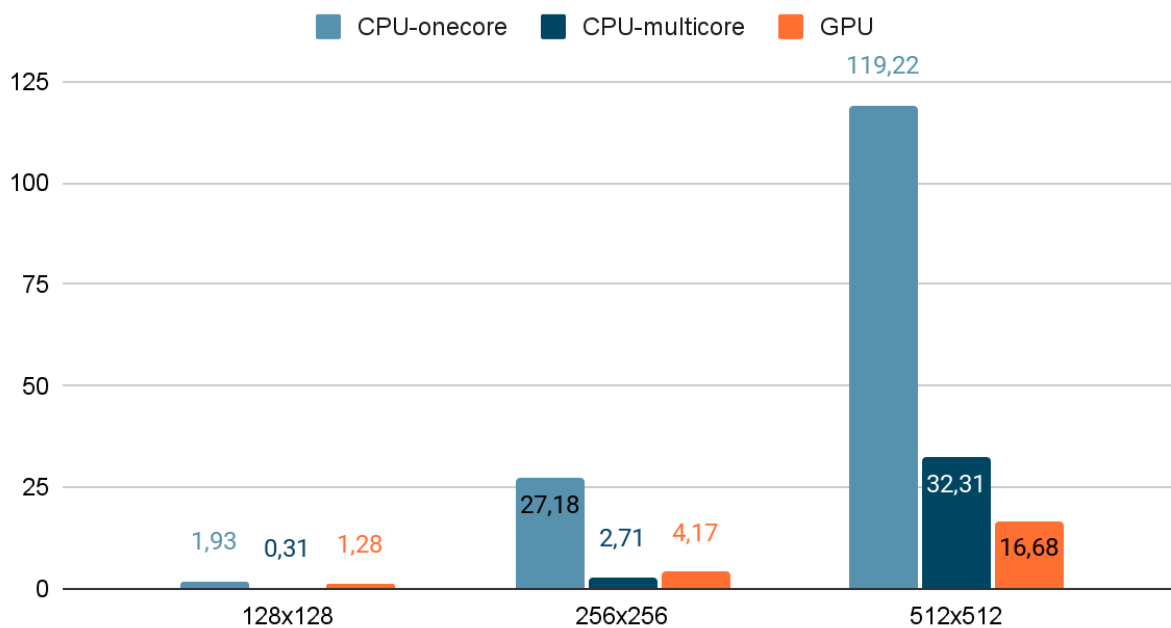
Размер сетки	Время выполнения	Точность	Количество итераций
128x128	0.31с	$9.9 * 10^{-7}$	30074
256x256	2.71с	$9.9 * 10^{-7}$	102885
512x512	32.31с	$9.9 * 10^{-7}$	339599

### GPU

Размер сетки	Время выполнения	Точность	Количество итераций
128x128	1.28с	$9.9 * 10^{-7}$	30074
256x256	4.17с	$9.9 * 10^{-7}$	102885
512x512	16.68с	$9.9 * 10^{-7}$	339599
1024x1024	108.86	$1.37 * 10^{-6}$	1000000

## Диаграмма сравнения время работы CPU-onecore, CPU-multicore и GPU

Общее время работы, с



# Выполнение на GPU

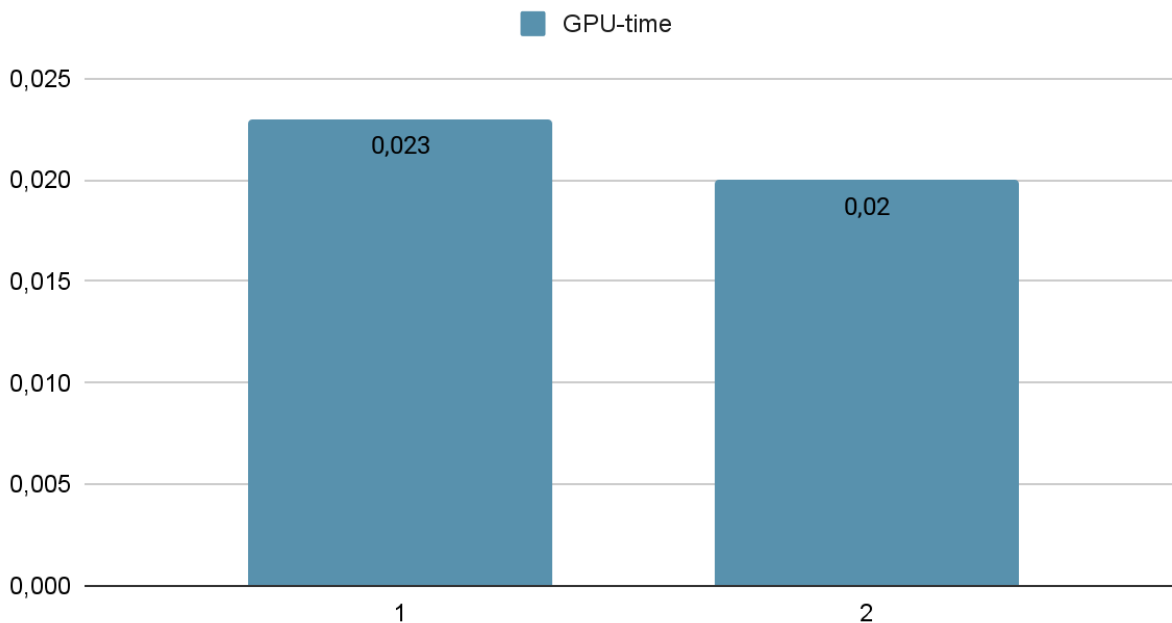
## Этапы оптимизации на сетке 512x512

Этап №	Время выполнения	Точность	Количество итераций	Комментарии (что было сделано)
1	0.023с	0.11	100	<p>Сетка представляется в виде двумерного массива;</p> <p>Под arr и arr_new выделяется память на CPU;</p> <p>Под arr_new выделяется память на GPU с помощью acc data create;</p> <p>arr, max_error и size переносятся на GPU с помощью директивы acc data copy</p> <p>На GPU происходит заполнение границ обеих сеток, этот процесс распараллеливается с помощью acc parallel loop independent</p> <p>В процессе расчета следующего шага сетки применяется редукция</p>

				<p>по функции <code>max</code> для переменной <code>max_error</code> с помощью <code>reduction(max:max_error)</code>);</p> <p>Копирование массивов происходит поэлементно и распараллеливается с помощью <code>acc parallel loop collapse(2) independent</code>;</p> <p>Значение <code>max_error</code> обновляется в памяти CPU в конце каждой итерации с помощью <code>acc update host</code></p>
2	0.020с	0.11	100	<p>Сетка представляется в виде одномерного массива;</p> <p>Обращение к индексам массива происходит с помощью макроса <code>ID</code>, которая переводит индексы двумерного массива в одномерный.</p> <p>Копирование массивов происходит с помощью копирования участка памяти из <code>arr_new</code> в <code>arr</code> на GPU</p>

Диаграмма оптимизации  
(по горизонтали - номер этапа;  
по вертикали - время работы, с)

Этапы оптимизации

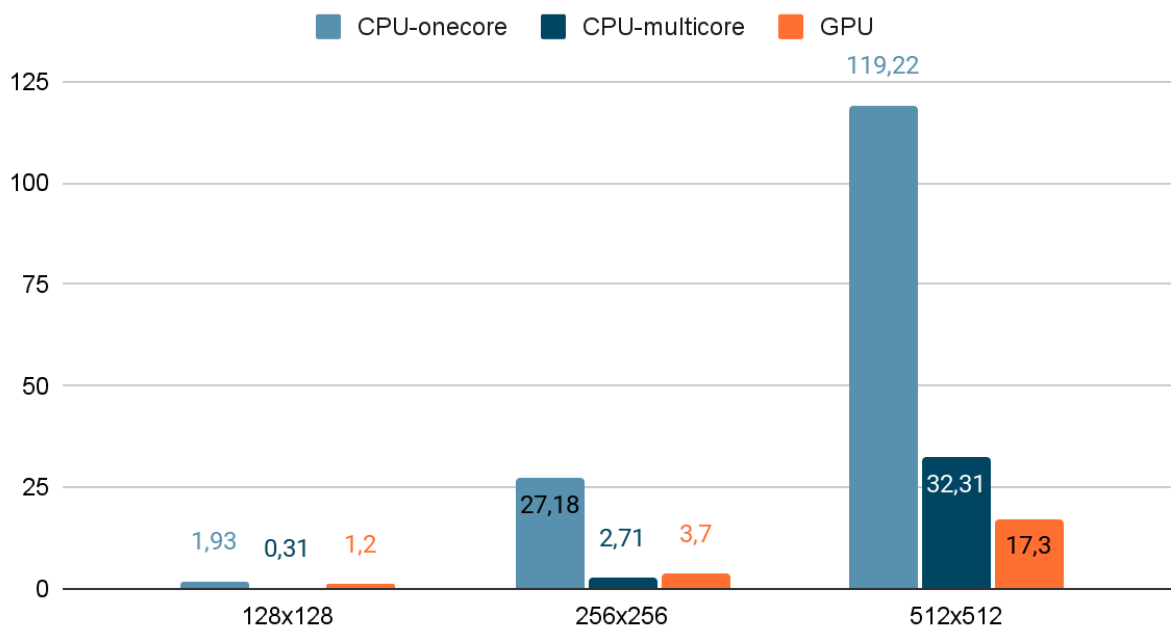


GPU – оптимизированный вариант

Размер сетки	Время выполнения	Точность	Количество итераций
128x128	1.20с	$9.9 * 10^{-7}$	30074
256x256	3.70с	$9.9 * 10^{-7}$	102885
512x512	17.30с	$9.9 * 10^{-7}$	350241
1024x1024	102.64с	$1.37 * 10^{-6}$	1000000

Диаграмма сравнения времени работы CPU-onecore, CPU-multicore, GPU(оптимизированный вариант) для разных размеров сеток

Общее время работы, с





Вывод:

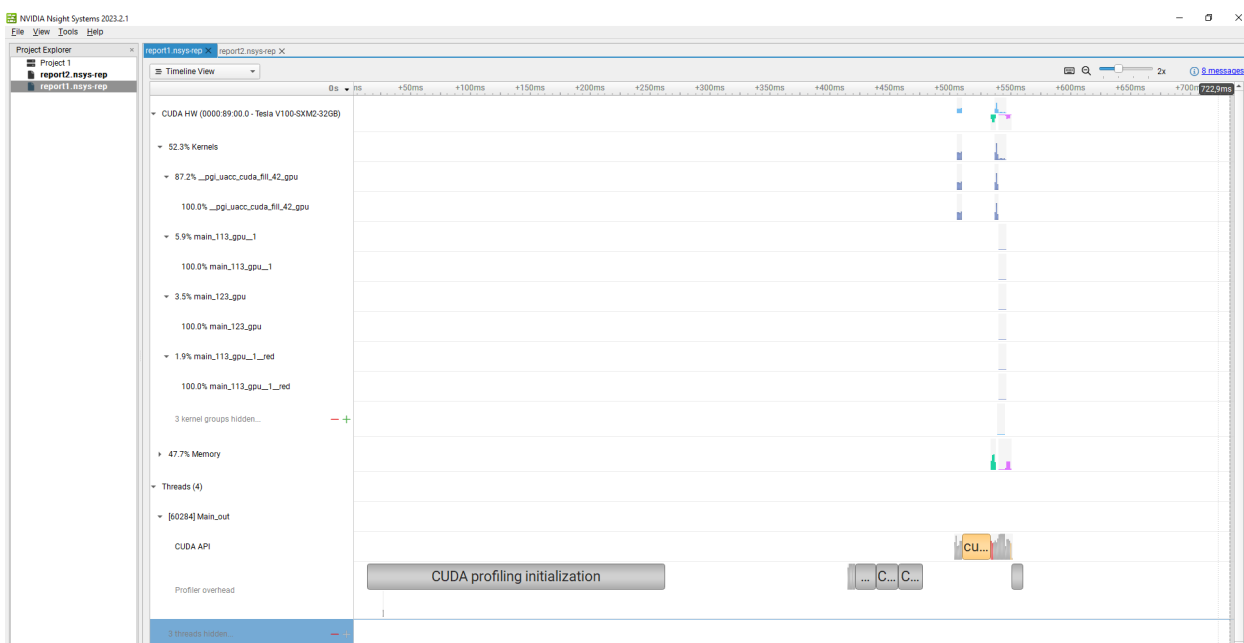
В ходе работы была написана программа для решения уравнения теплопроводности, было произведено сравнение времени выполнения на CPU-onecore, CPU-multicore и GPU, была выполнена оптимизация кода, а также было выполнено профилирование через Nsight System.

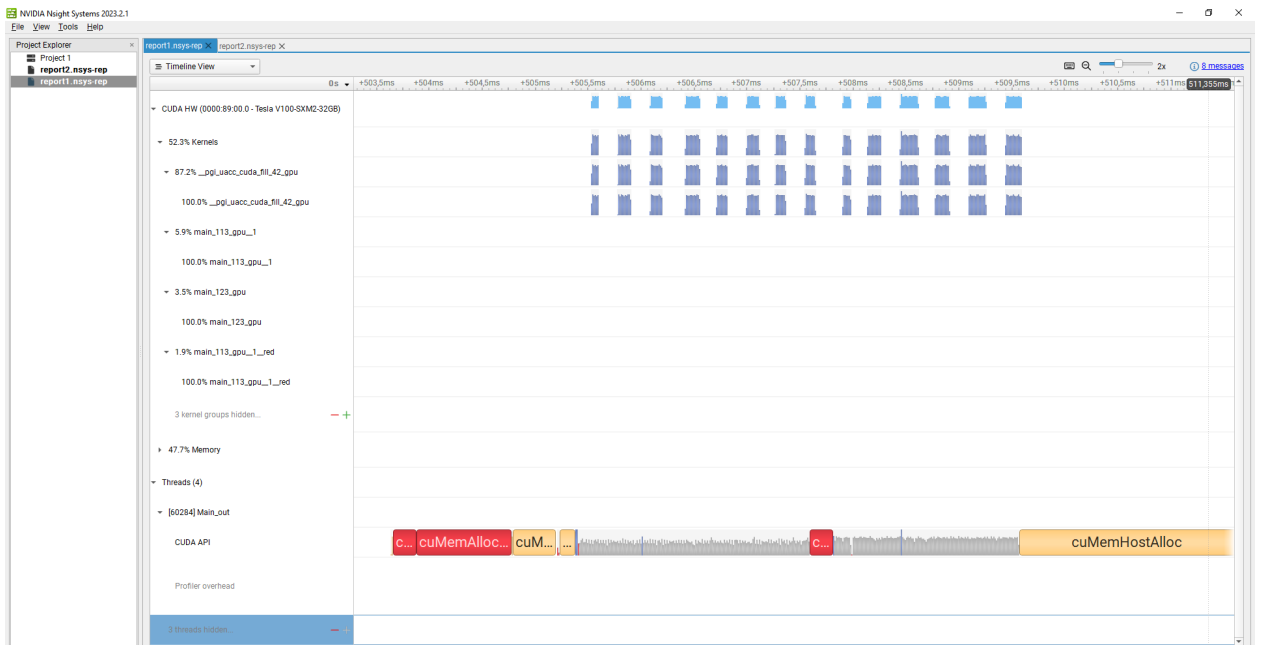
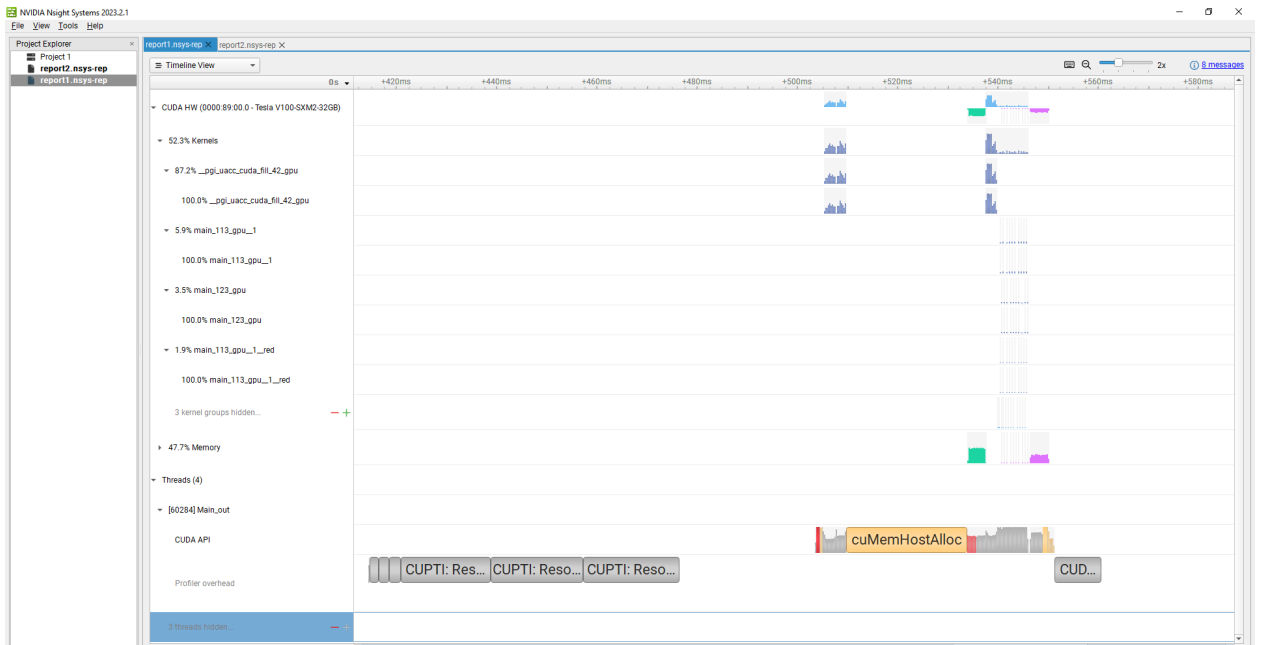
Как видно из полученных данных, с ростом размера сетки общее время меньше всего становится при выполнении на GPU. При выполнении на небольших размерах сетки лучше выполнять программу на CPU-multicore, так как из-за копирования данных время выполнения на GPU будет больше.

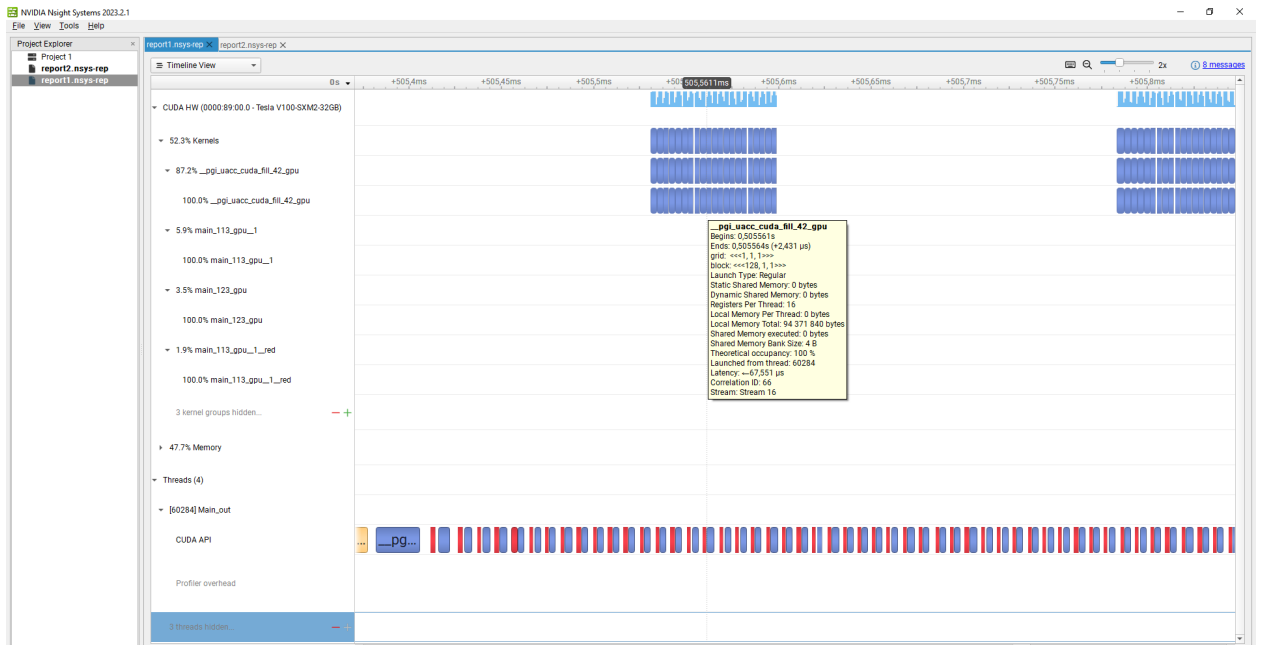
# Приложение

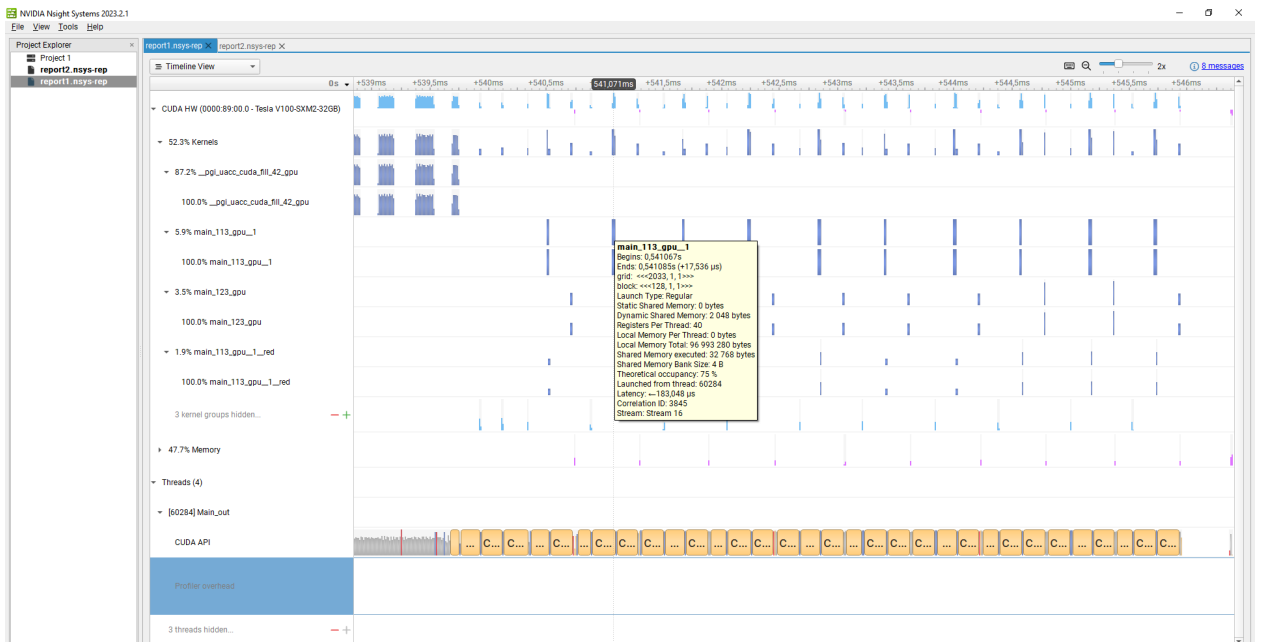
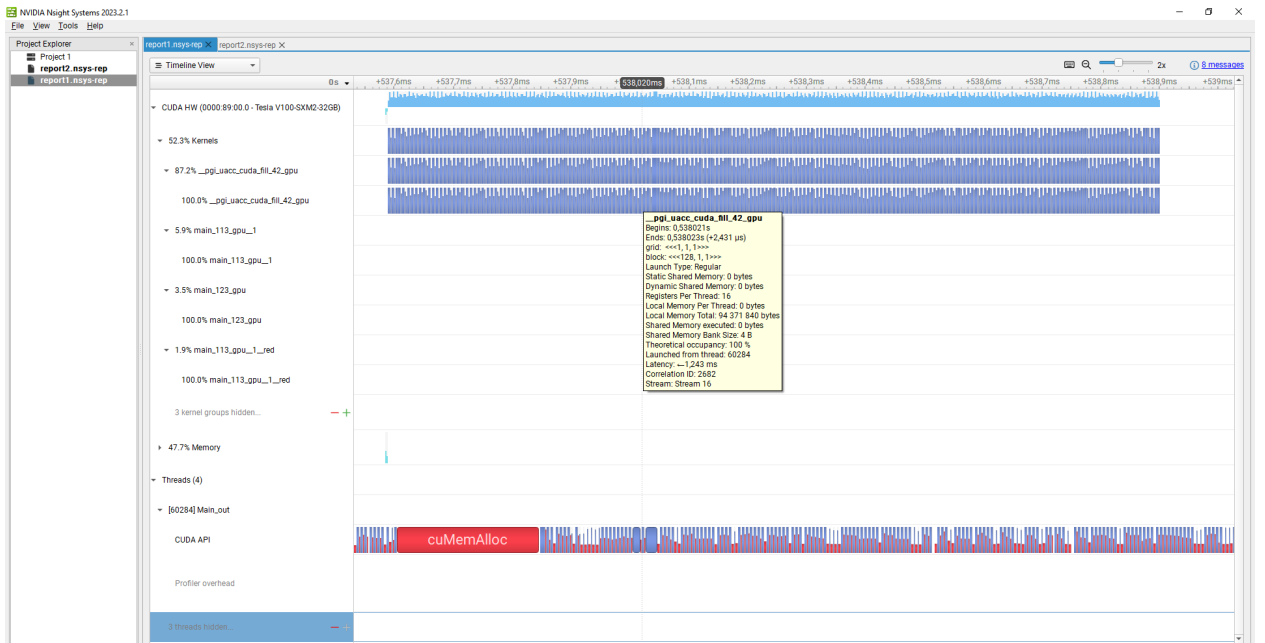
[https://github.com/HerrPhoton/Heat\\_equation](https://github.com/HerrPhoton/Heat_equation)

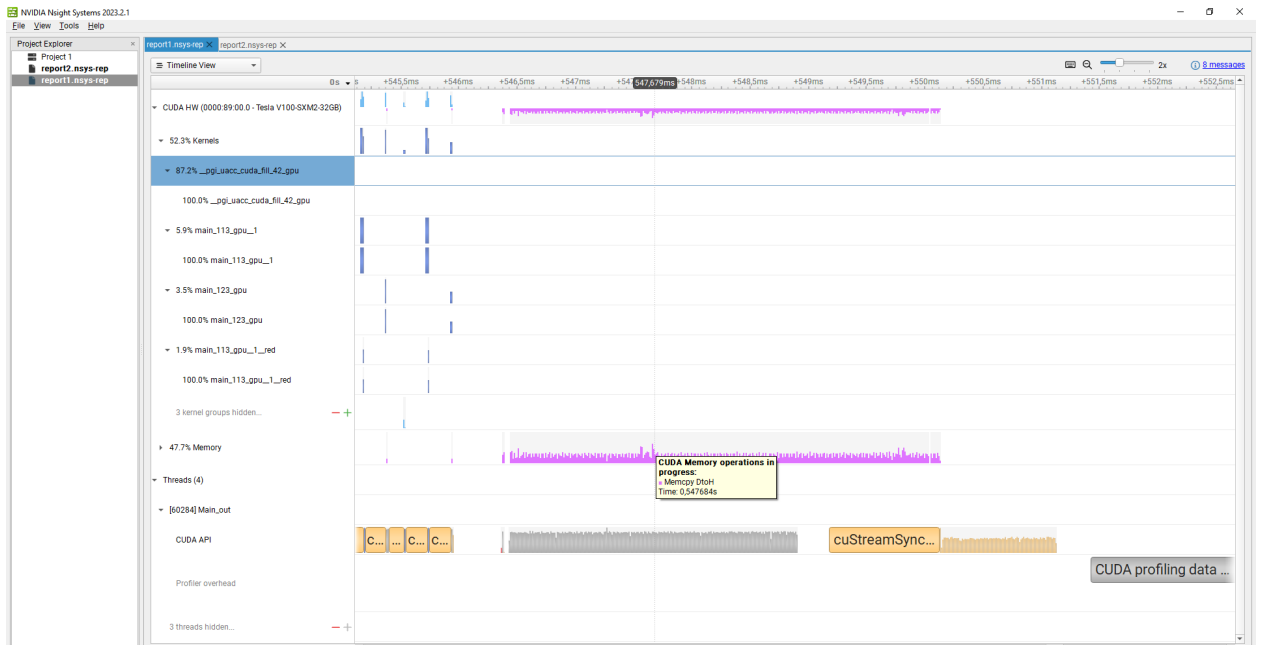
## Профилирование кода на этапе №1











# Профилирование кода на этапе №2

