

Теория параллелизма

Отчёт

Оптимизированные библиотеки

Выполнил Солопов Илья

Группа 21932

27.03.2023

ВВЕДЕНИЕ	3
Цели работы.....	3
Используемый компилятор	3
Используемый профилировщик	3
Замер времени работы.....	3
ВЫПОЛНЕНИЕ НА CPU.....	4
CPU-onecore.....	4
CPU-multicore.....	4
Диаграмма сравнения времени работы.....	4
ВЫПОЛНЕНИЕ НА GPU	5
Этапы оптимизации.....	5
Диаграмма оптимизации	5
GPU – вариант с использованием OpenACC	6
GPU – оптимизированный вариант (cuBLAS).....	6
Диаграмма сравнения времени работы.....	7
Диаграмма сравнения времени работы.....	7
ВЫВОД	8
ПРИЛОЖЕНИЕ.....	9
Ссылка на репозиторий	9
Скриншоты массивов.....	9
Скриншоты из профилировщика.....	9

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 директивы OpenACC. Операцию редукции (вычисление максимального значения ошибки) реализовать через вызовы функций из библиотеки cuBLAS.

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

Произвести профилирование программы и оптимизацию кода.

Используемый компилятор

Использовался компилятор PGC++.

Используемый профилировщик

Использовался профилировщик NVIDIA NsightSystems.

Замер времени работы

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

Выполнение на CPU

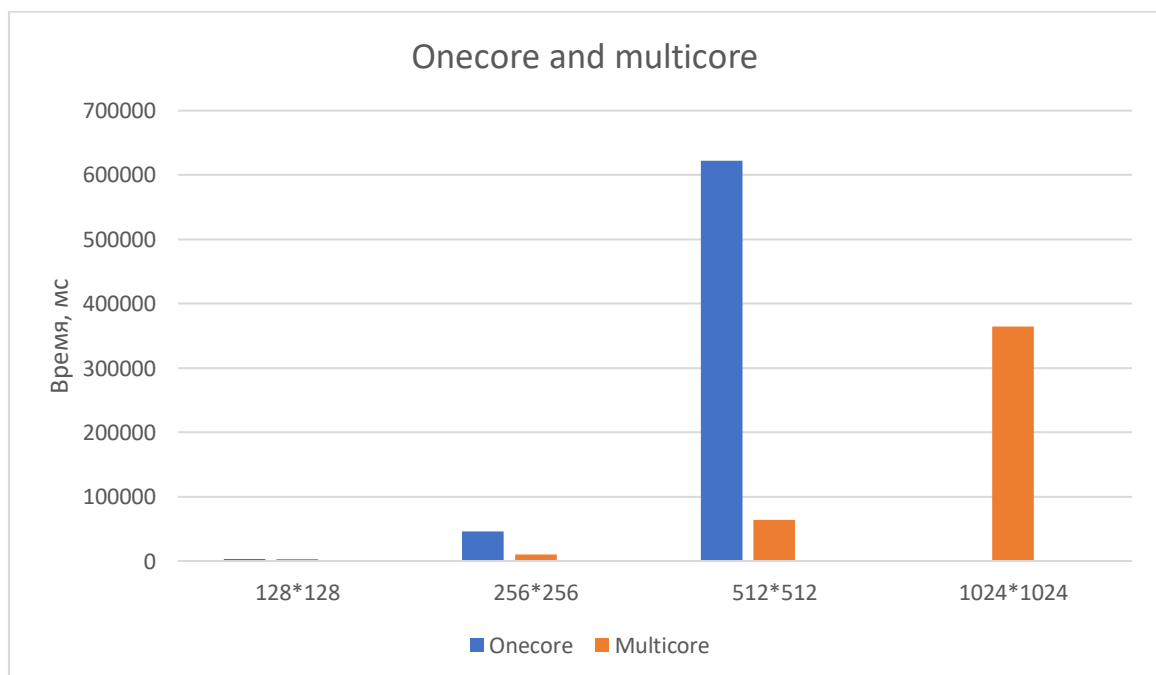
CPU-onecore

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



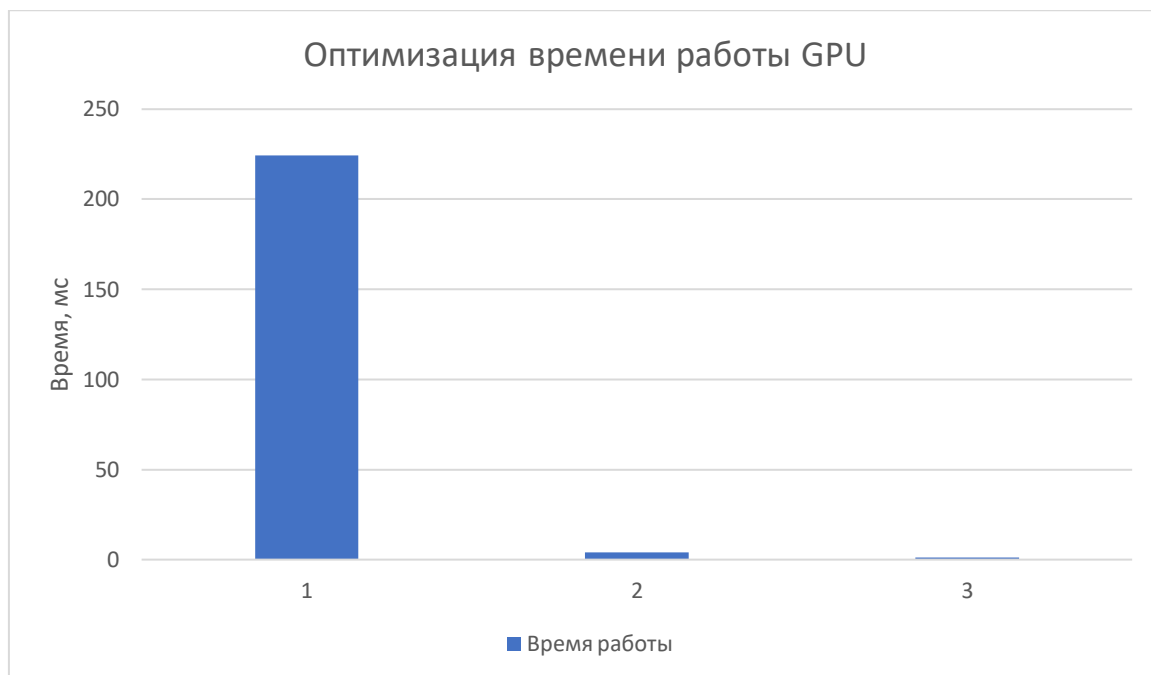
Выполнение на GPU

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	224.2	0.0359466	100	Использованы только директивы openACC.
2	3.91	0.107043	100	Использование функций библиотеки cuBLAS для вычисления максимального значения ошибки каждую итерацию.
3	1.28	0.03596	100	Используется async. Ошибка высчитывается не на каждой итерации.

Диаграмма оптимизации



GPU – вариант с использованием OpenACC

Размер сетки	Время выполнения, мс	Точность	Количество итераций
128*128	228.28	9.9998e-07	30081
256*256	821.27	9.9993e-07	102913
512*512	4092.98	9.99984e-07	339969
1024*1024	44710.47	1.373e-06	1000000

GPU – оптимизированный вариант (cuBLAS)

Размер сетки	Время выполнения, мс	Точность	Количество итераций
128*128	106.04	9.97841e-07	30081
256*256	402.92	9.97808e-07	102913
512*512	2053.32	9.93016e-07	339969
1024*1024	28275.99	1.373e-06	1000000

Диаграмма сравнения времени работы

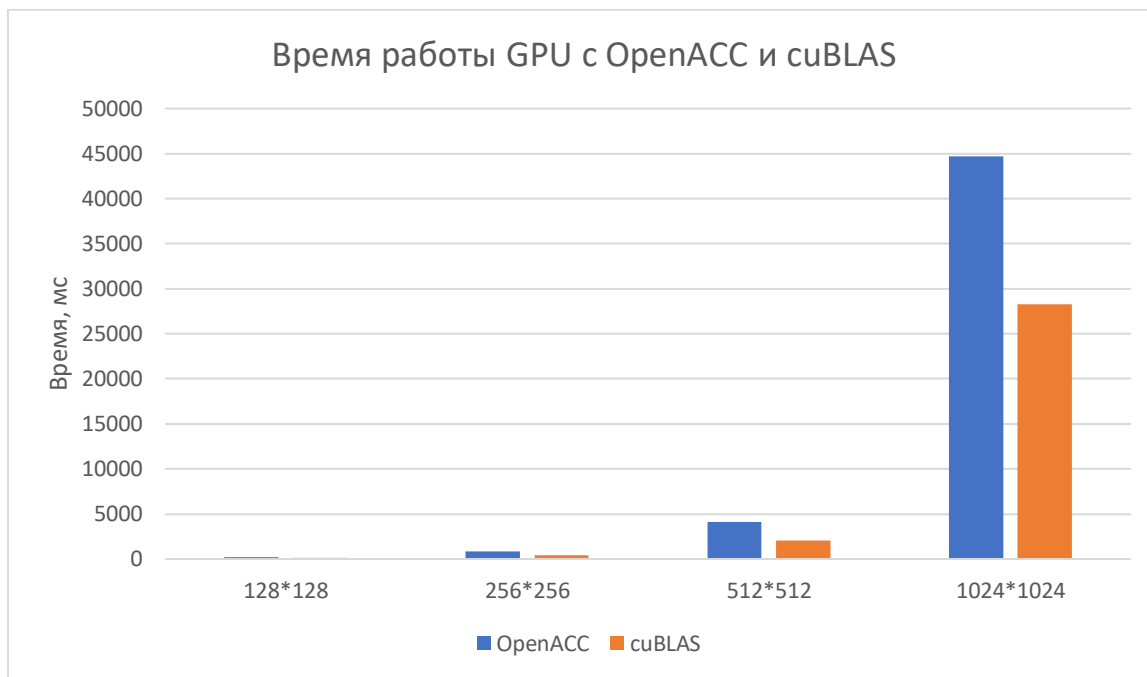
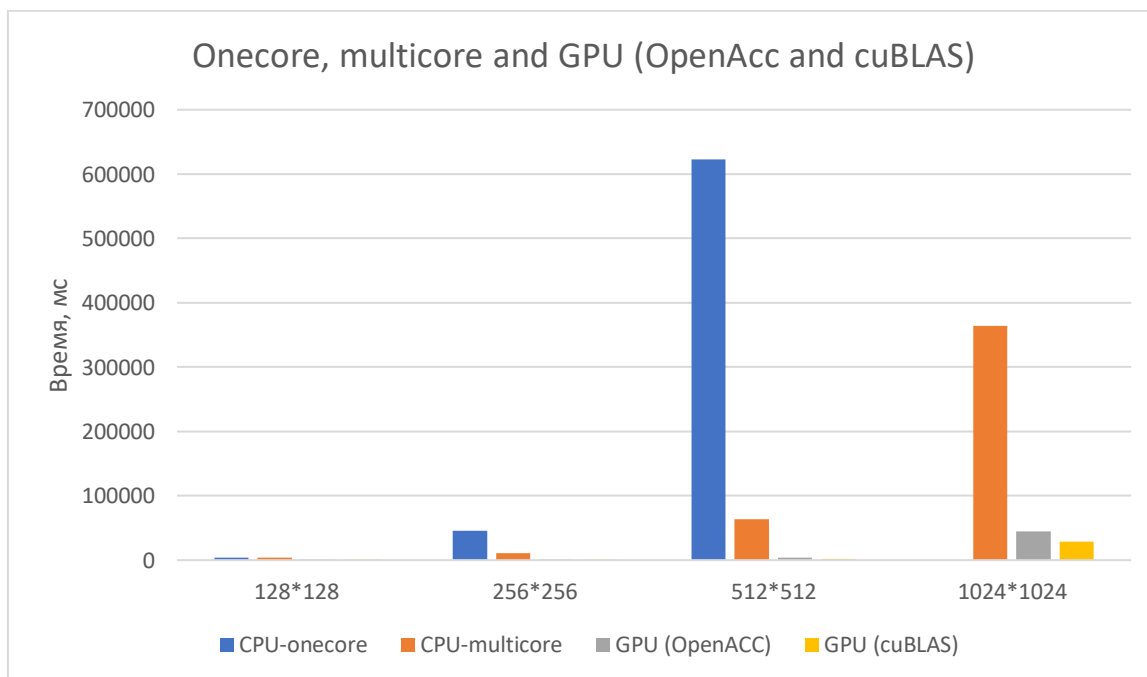


Диаграмма сравнения времени работы



Вывод

Полученные результаты говорят нам о том, что использование библиотеки cuBLAS даёт значительный прирост производительности на сетках всех размерностей.

Приложение

Ссылка на репозиторий

<https://github.com/IIS0/Parallelism-tasks>

Скриншоты массивов

Размер сетки 15*15.

Массив после инициализации

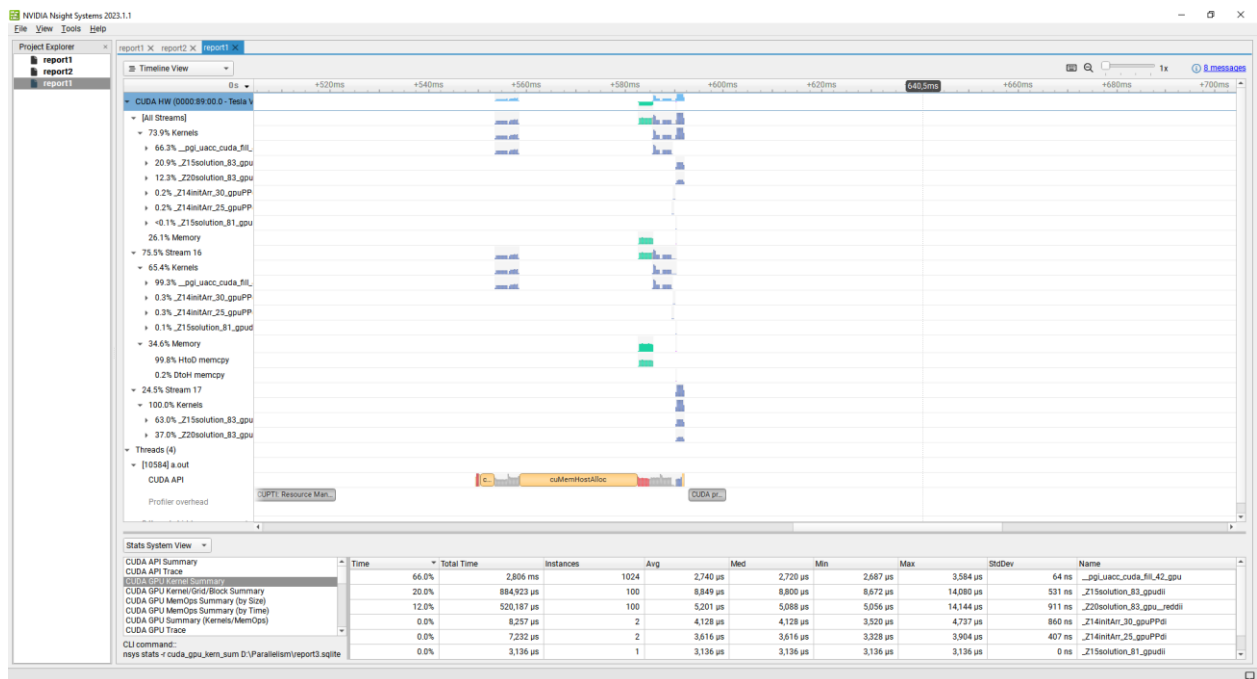
```
10.000000 10.714286 11.428571 12.142857 12.857143 13.571429 14.285714 15.000000 15.714286 16.428571 17.142857 17.857143 18.571429 19.285714 20.000000
10.714286 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 20.714286
11.428571 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 21.428571
12.142857 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 22.142857
12.857143 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 22.857143
13.571429 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 23.571429
14.285714 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 24.285714
15.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 25.000000
15.714286 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 25.714286
16.428571 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 26.428571
17.142857 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 27.142857
17.857143 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 27.857143
18.571429 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 28.571429
19.285714 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 29.285714
20.000000 20.714286 21.428571 22.142857 22.857143 23.571429 24.285714 25.000000 25.714286 26.428571 27.142857 27.857143 28.571429 29.285714 30.000000
```

Массив после работы программы

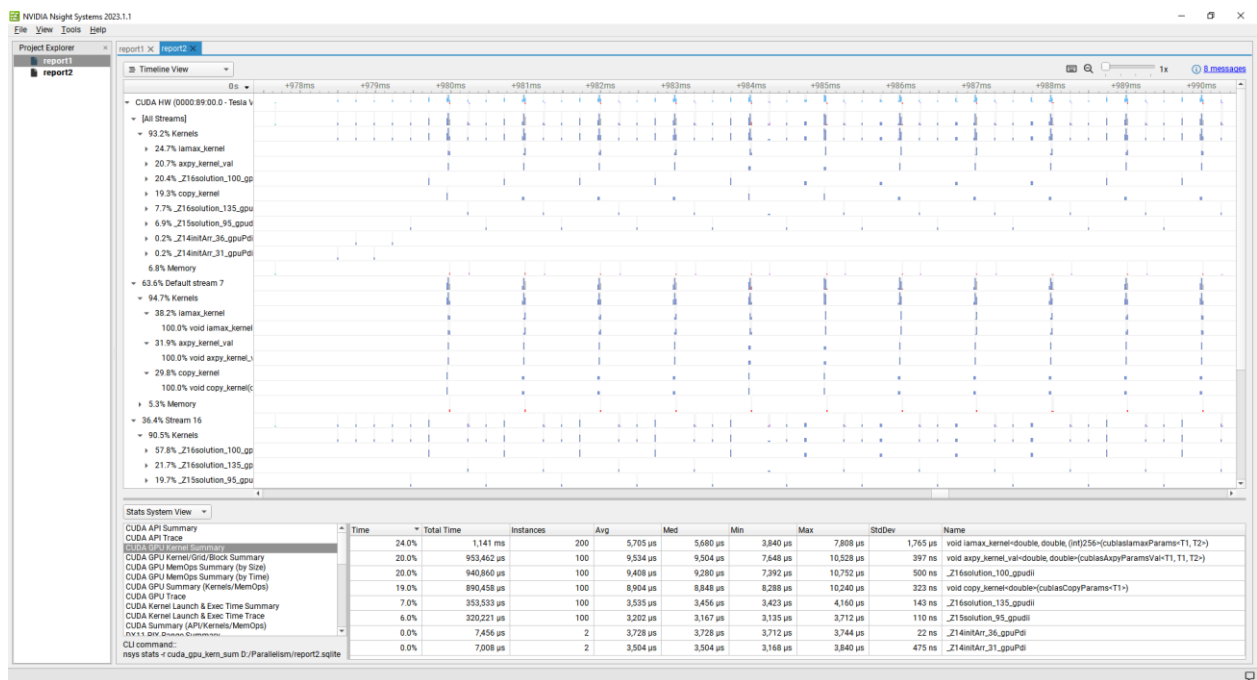
```
10.000000 10.714286 11.428571 12.142857 12.857143 13.571429 14.285714 15.000000 15.714286 16.428571 17.142857 17.857143 18.571429 19.285714 20.000000
10.714286 11.428570 12.142854 12.857138 13.571423 14.285707 14.999992 15.714278 16.428564 17.142850 17.857137 18.571424 19.285711 19.999998 20.714286
11.428571 12.142854 12.857136 13.571419 14.285702 14.999986 15.714271 16.428556 17.142842 17.857129 18.571417 19.285705 19.999993 20.714282 21.428571
12.142857 12.857138 13.571419 14.285701 14.999983 15.714266 16.428550 17.142835 17.857122 18.571409 19.285697 19.999986 20.714276 21.428567 22.142857
12.857143 13.571423 14.285702 14.999983 15.714264 16.428547 17.142831 17.857116 18.571402 19.285690 19.999979 20.714269 21.428560 22.142851 22.857143
13.571429 14.285707 14.999986 15.714266 16.428547 17.142829 17.857112 18.571397 19.285684 19.999972 20.714261 21.428552 22.142844 22.857136 23.571429
14.285714 14.999992 15.714271 16.428550 17.142831 17.857112 18.571396 19.285680 19.999967 20.714255 21.428545 22.142836 22.857128 23.571421 24.285714
15.000000 15.714278 16.428556 17.142835 17.857116 18.571397 19.285680 19.999965 20.714252 21.428540 22.142830 22.857121 23.571413 24.285707 25.000000
15.714286 16.428564 17.142842 17.857122 18.571402 19.285684 19.999967 20.714252 21.428538 22.142827 22.857116 23.571407 24.285700 24.999992 25.714286
16.428571 17.142850 17.857129 18.571409 19.285690 19.999972 20.714255 21.428540 22.142827 22.857115 23.571404 24.285695 24.999986 25.714279 26.428571
17.142857 17.857137 18.571417 19.285697 19.999979 20.714261 21.428545 22.142830 22.857116 23.571404 24.285693 24.999983 25.714274 26.428565 27.142857
17.857143 18.571424 19.285705 19.999986 20.714269 21.428552 22.142836 22.857121 23.571407 24.285695 24.999983 25.714272 26.428562 27.142852 27.857143
18.571429 19.285711 19.999993 20.714276 21.428560 22.142844 22.857128 23.571413 24.285700 24.999986 25.714274 26.428562 27.142851 27.857139 28.571429
19.285714 19.999998 20.714282 21.428567 22.142851 22.857136 23.571421 24.285707 24.999992 25.714279 26.428565 27.142852 27.857139 28.571427 29.285714
20.000000 20.714286 21.428571 22.142857 22.857143 23.571429 24.285714 25.000000 25.714286 26.428571 27.142857 27.857143 28.571429 29.285714 30.000000
```

Скриншоты из профилировщика

Этап 1



Этап 2



Этап 3

