Теория параллелизма

Отчет

Задание 8

Выполнила Агапова Александра, 23932 группа

17.04.2025г

Выполнение на CPU

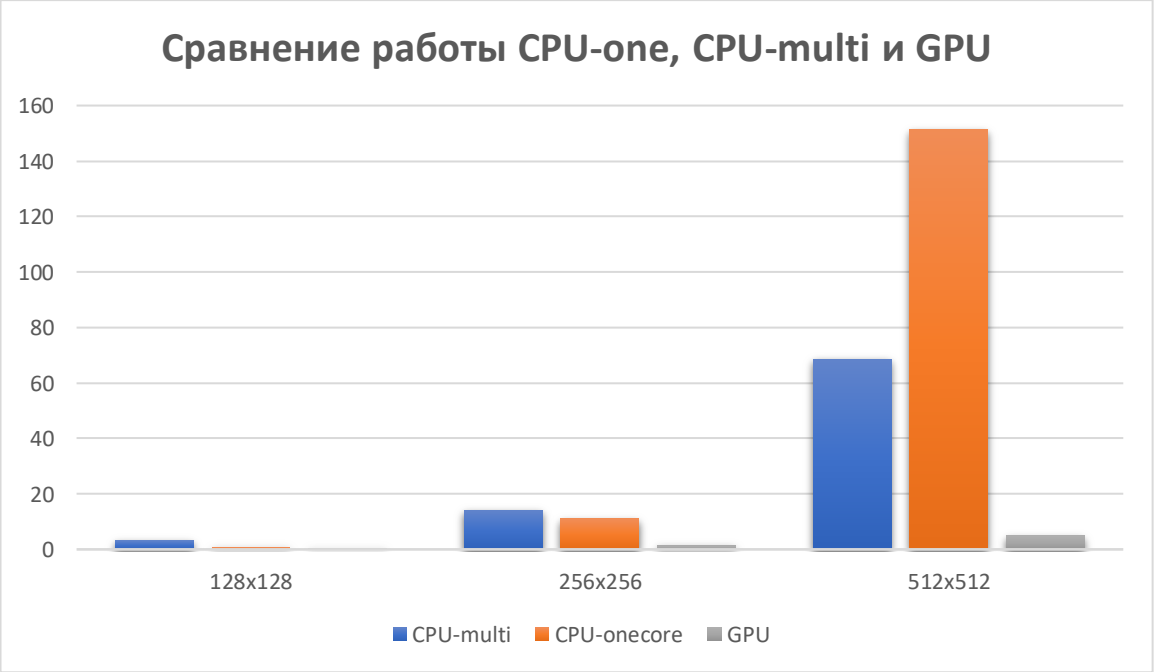
CPU-onecore

Размер сетки	Время выполнения	Точность	Количество итераций
128x128	0.828311	7.53014e-07	1.000.000
256x256	11.0707	9.91166e-07	1.000.000
512x512	151.382	9.92416e-07	1.000.000

CPU-multicore

Размер сетки	Время выполнения	Точность	Количество итераций
128x128	3.38594	7.53014e-07	1.000.000
256x256	14.1547	9.91166e-07	1.000.000
512x512	68.6179	9.92416e-07	1.000.000
1024x1024	461.941	1.37575e-06	1.000.000

Диаграмма сравнения времени работы CPU-one и CPU-multi



Выполнение на GPU (оптимизированный вариант)

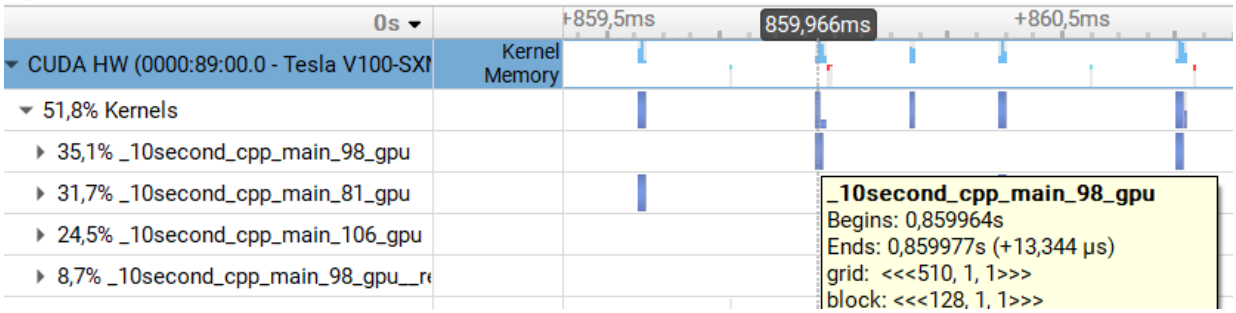
Размер сетки	Время выполнения	Точность	Количество итераций
128x128	0.488777	7.53014e-07	1.000.000
256x256	1.21184	9.91166e-07	1.000.000
512x512	4.74862	9.92416e-07	1.000.000
1024x1024	36.6044	1.37575e-06	1.000.000

Этапы оптимизации на сетке 512x512

Максимальное количество итераций – 1.000.000

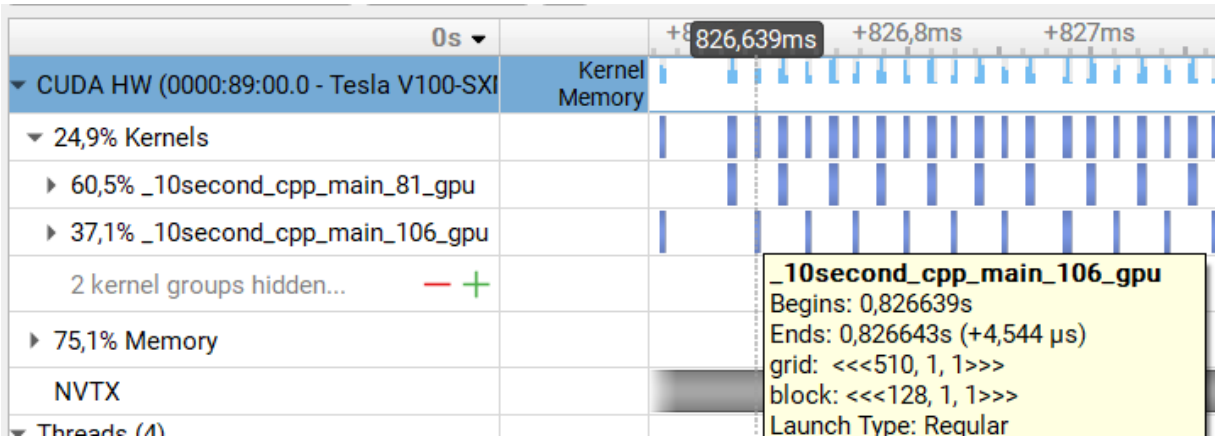
Этап №	Время выполнения	Точность	Комментарии
1	148.368	9.99984e-07	*Начальная версия*
2	13.15386	9.92416e-07	*Подсчет ошибки не на каждой итерации, а каждую 1000 итерацию*
3	8.97609	9.92416e-07	*Вложенные #pragma acc loop внутри #pragma acc parallel loop заменила на collapse(2)*
4	4.74862	9.92416e-07	*Вместо копирования каждого элемента из Anew в A использую swap, который мгновенно меняет местами указатели A и Anew*

1)



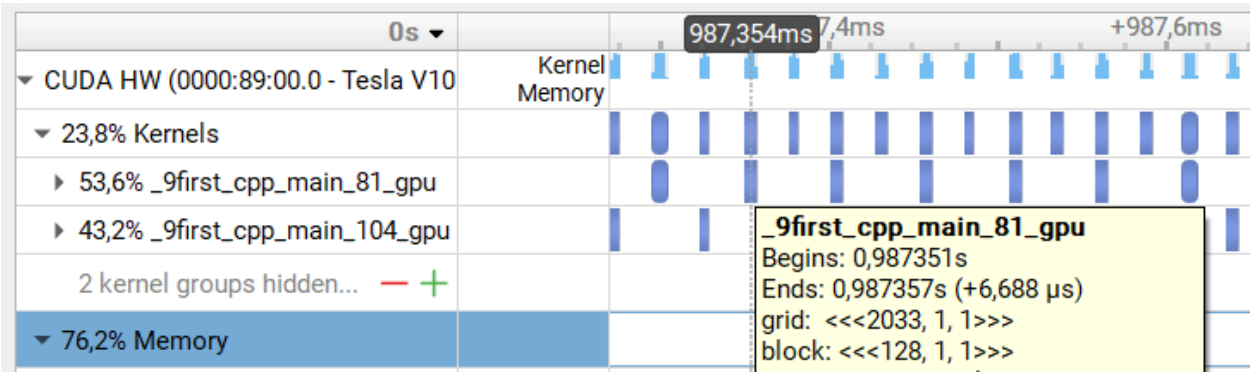
98 строка – подсчет ошибки на каждой итерации

2)



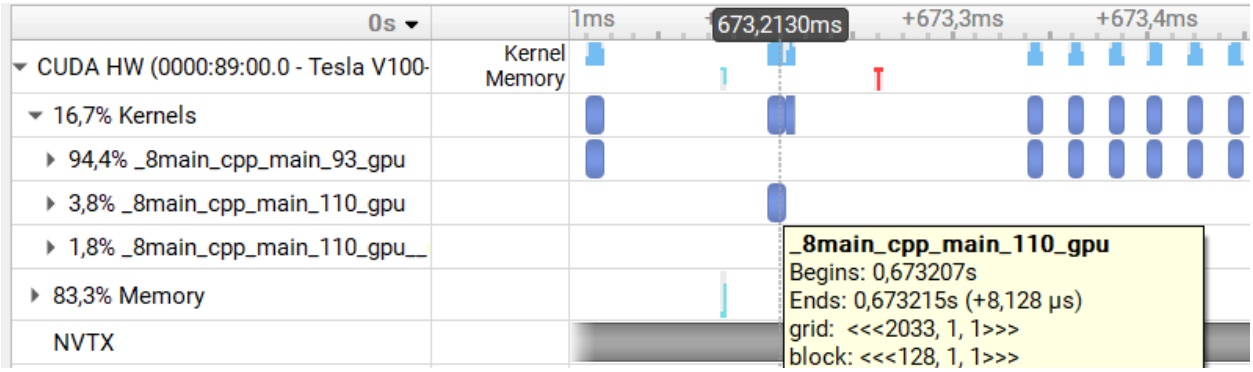
81 строка – #pragma acc parallel loop present(A, Anew) цикл обновления значений в матрице Anew на основе соседних значений из A

3)



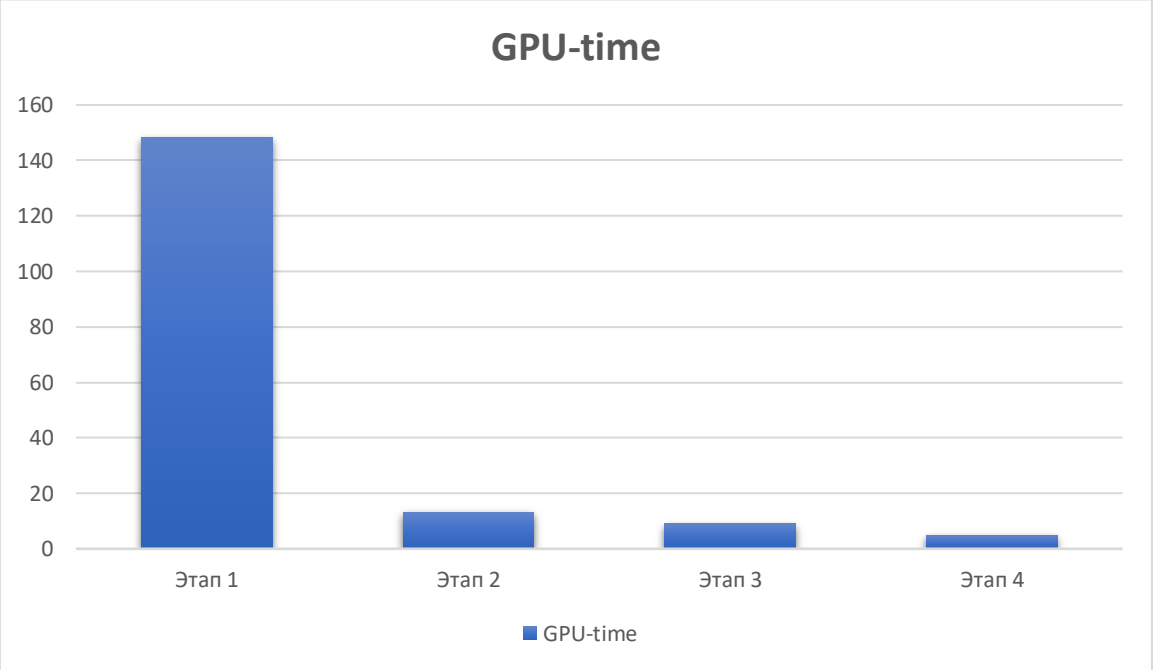
81 строка - #pragma acc parallel loop collapse(2) present(A, Anew) объединила два вложенных цикла в один двухмерный параллельный цикл

4)



После замены копирования элементов на swap пропала kernel копирования

Диаграмма оптимизации

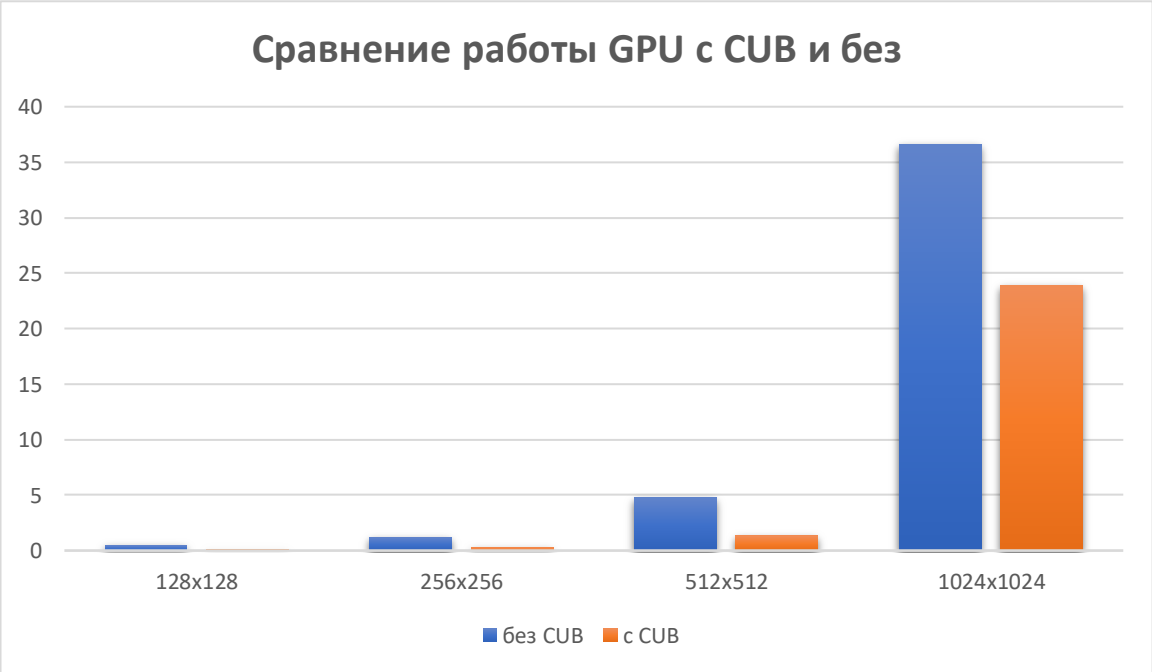



```
Посчитанная матрица 10 x 10
0, ошибка = 14.4444
10      11.1111 12.2222 13.3333 14.4444 15.5556 16.6667 17.7778 18.8889 20
11.1111 12.2222 13.3333 14.4444 15.5556 16.6667 17.7778 18.8889 20      21.1111
12.2222 13.3333 14.4444 15.5556 16.6667 17.7778 18.8889 20      21.1111 22.2222
13.3333 14.4444 15.5556 16.6667 17.7778 18.8889 20      21.1111 22.2222 23.3333
14.4444 15.5556 16.6667 17.7778 18.8889 20      21.1111 22.2222 23.3333 24.4444
15.5556 16.6667 17.7778 18.8889 20      21.1111 22.2222 23.3333 24.4444 25.5556
16.6667 17.7778 18.8889 20      21.1111 22.2222 23.3333 24.4444 25.5556 26.6667
17.7778 18.8889 20      21.1111 22.2222 23.3333 24.4444 25.5556 26.6667 27.7778
18.8889 20      21.1111 22.2222 23.3333 24.4444 25.5556 26.6667 27.7778 28.8889
20      21.1111 22.2222 23.3333 24.4444 25.5556 26.6667 27.7778 28.8889 30
1001 итераций
Ошибка: 0
Время: 0.193013 секунд
```

```
Посчитанная матрица 13 x 13
0, ошибка = 14.5833
10      10.8333 11.6667 12.5    13.3333 14.1667 15      15.8333 16.6667 17.5    18.3333 19.1667 20
10.8333 11.6667 12.5    13.3333 14.1667 15      15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333
11.6667 12.5    13.3333 14.1667 15      15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667
12.5    13.3333 14.1667 15      15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667 22.5
13.3333 14.1667 15      15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333
14.1667 15      15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667
15      15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667 25
15.8333 16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667 25      25.8333
16.6667 17.5    18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667 25      25.8333 26.6667
17.5    18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667 25      25.8333 26.6667 27.5
18.3333 19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667 25      25.8333 26.6667 27.5    28.3333
19.1667 20      20.8333 21.6667 22.5    23.3333 24.1667 25      25.8333 26.6667 27.5    28.3333 29.1667
20      20.8333 21.6667 22.5    23.3333 24.1667 25      25.8333 26.6667 27.5    28.3333 29.1667 30
1001 итераций
Ошибка: 3.55271e-15
Время: 0.181927 секунд
```

Реализация с библиотекой CUB

Размер сетки	Время выполнения	Точность	Количество итераций
128x128	0.0727762	7.53014e-07	1.000.000
256x256	0.278514	9.91166e-07	1.000.000
512x512	1.33679	9.92416e-07	1.000.000
1024x1024	23.8688	1.36929e-06	1.000.000

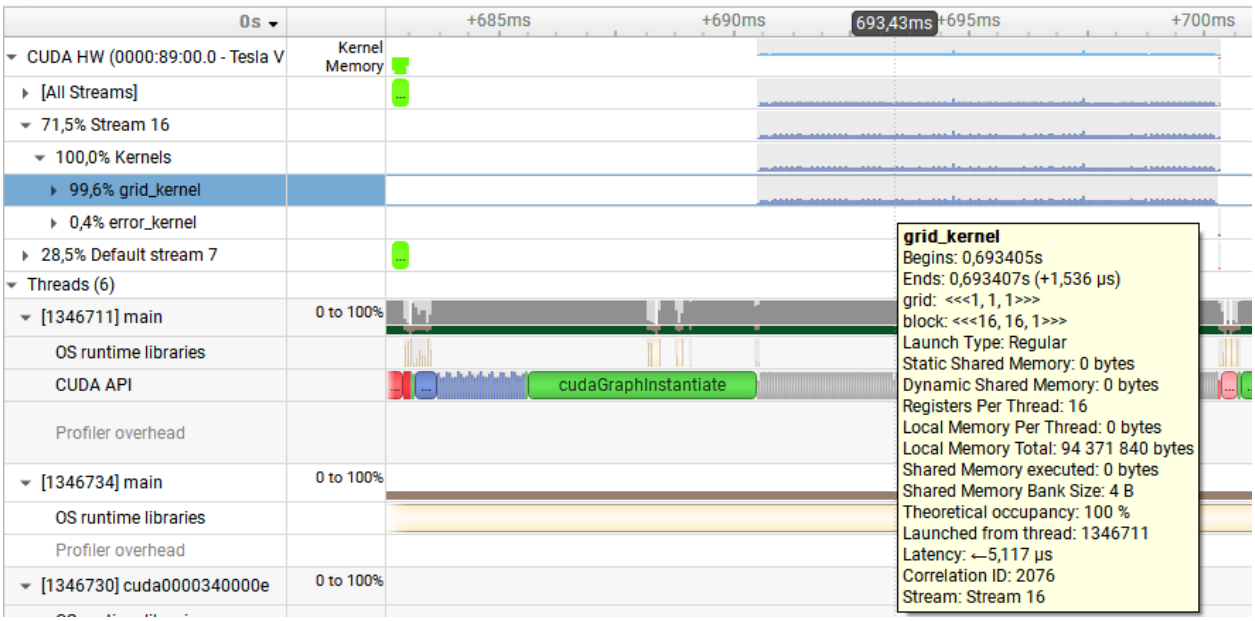


Этапы оптимизации на сетке 512x512

Максимальное количество итераций – 1.000.000

Этап №	Время выполнения	Точность	Комментарии
1	4.59136	9.84603e-07	*Начальная версия*
2	1.19765	9.92435e-07	*Единый крупный запуск ядра, убрала лишние* cudaStreamSynchronize()

1)



Вычисления ядра grid_kernel были раздроблены на множество мелких запусков (по одному блоку на запуск), каждый из которых синхронизировался вручную с CPU через cudaStreamSynchronize(). Даже несмотря на то, что каждый запуск занимал доли миллисекунды , общее количество привело к 99% загрузке GPU по времени

2)