**Задача 3.**

Выполнил: Сантаев Павел

# Использование 12 мп фотографии (результат 2ой задачи, для сравнения)

|  |  |  |
| --- | --- | --- |
| Размер блока | Время исполнения ядра | Загрузка |
| 16х16 | 39ms | 78% |
| 8х8 | 48ms | 32% |
| 4х4 | 423ms | 25% |

# Результаты программы с использованием нескольких GPU.

Время исполнения ядра: 23ms и 16ms. В сумме 39 ms.

Загрузка 78% и 96%.

Производительность на первой GPU = 5000 \* 3400 \* 9 / 23ms / 2 = 3.5 ГФлопс

# Оптимизация:

Сначала я сделал inline функция получения и изменения пикселей. Это ничего не дало.

Далее сделал inline вручную. Тоже ничего не дало.

Потом начал оптимизировать сам код, убрал лишние вычисления, лишнее использование памяти.

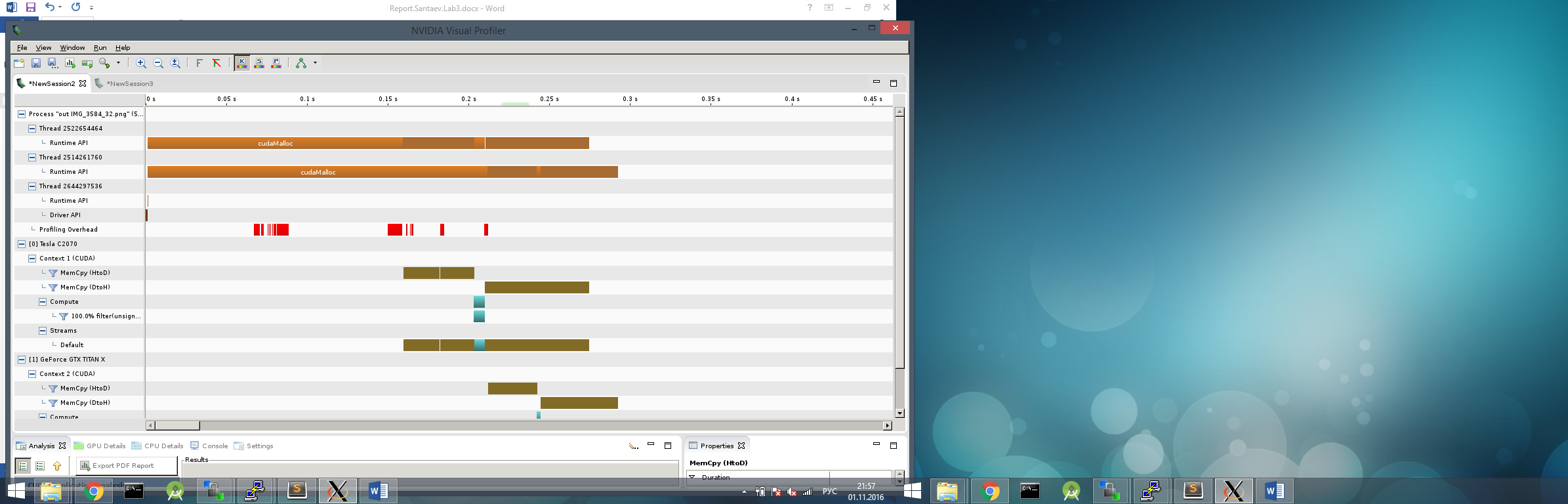
После, время исполнения ядра: 10ms и 10ms. В сумме 30 ms. Ускорение на первом GPU 2.3 раза.

Производительность на первой GPU = 5000 \* 3400 \* 9 / 10ms / 2 = 7.6 ГФлопс.

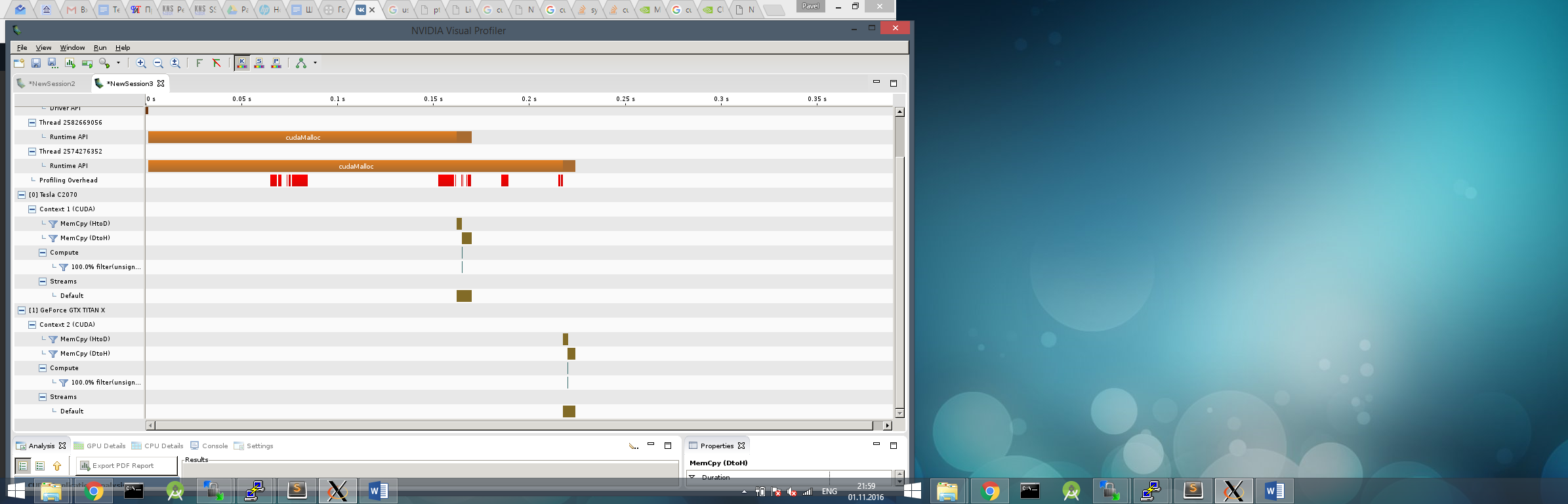
По идее, чтобы суммировать 2 пикселя, производится не 1 операция, а 3., т. к. каждый канал нужно считать отдельно.

# Распараллеливание

Профайл с аргументом IMG\_3584\_32.png



Профайл с аргументом Lenna.png



Оказывается, что потоки и так исполнялись параллельно. На скриншотах видно, что выделение памяти начинается одновременно, но заканчивается в разное время. Из-за этого процесс копирования и сам фильтр запускаются не всегда одновременно.

Убрал я лишнее копирование, из-за этого хотя бы некоторых местах копирование и фильтр стали работать одновременно.

Из-за того, что было лишнее копирование у первого GPU, оно длилось больше. Сейчас разница намного меньше.