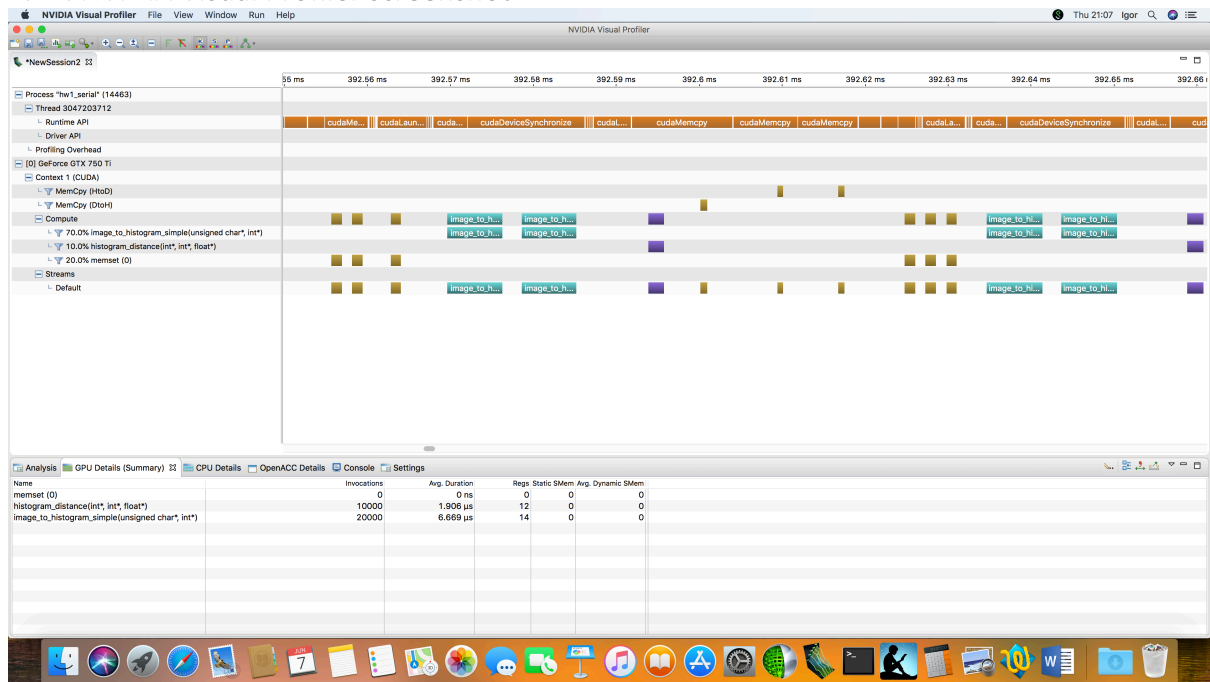


1.
 - 1.1. CUDA version: 5.0
 - 1.2. GPU name: GeForce GTX 750 Ti (GM107)
 - 1.3. #SMs: 5

2.
 - 2.3. atomicAdd is required because multiple threads are running simultaneously and updates histogram array elements. In order to promise correct calculation of histogram each thread should update histogram array in safe manner by atomic access.
 - 2.10. Run time: 0.467 [sec], throughput, 21413 [face comparisons / sec]
 - 2.11. NVIDIA Visual Profiler screenshot



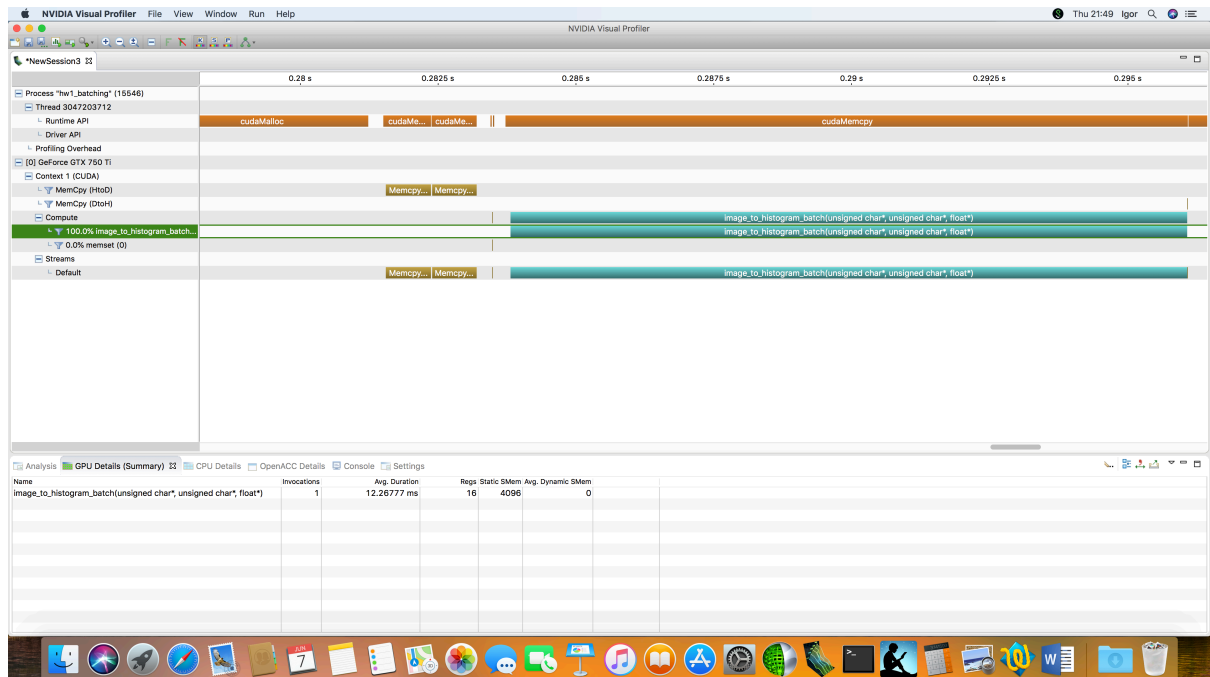
- 2.12. Memcpy time: 7 [usec]. Average per byte: 7 [usec] / 1024 bytes = 0.0068 [usec]

3.
 - 3.3. I load the image to shared memory because shared memory is on-chip memory and variables that reside in this memory can be accessed at very high-speed in a highly parallel manner.
 - 3.4. I compute the histogram in shared memory because each block calculates histogram for 32x32 picture and both image and histogram could reside in shared memory of SM [48K]. At the beginning each thread of the block copies one element of image array into preallocated shared memory, calculates histogram in local_hist[] also located in shared memory and first 256 threads copies histogram into global device memory.
 - 3.5 Run time: 0.396 [sec]. Speedup: 0.467/0.396 = 1.18

4.

4.5. Run time: 0.011 [sec]. Speedup = $0.396/0.011 = 36$

4.6. NVIDIA Visual Profiler screenshot



4.7. Malloc time: 817 [usec]. Average per byte: $817 \text{ [usec]} / 10240000 = 0.00008 \text{ [usec]}$.
==> Not linear with data size.

5. We could eliminate handling of corner cases by splitting image_to_histogram kernel to nine distinct kernels 4 for corner pixels, another 4 for border line cases and one for all internal pixels (internal pixel has all 9 surrounding neighbor pixels).