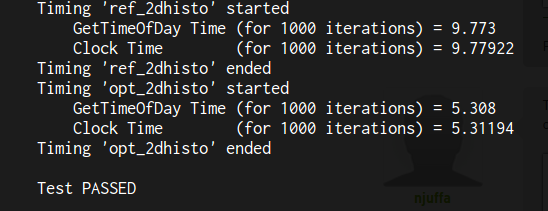
EECS468 Lab3 Histogram Report

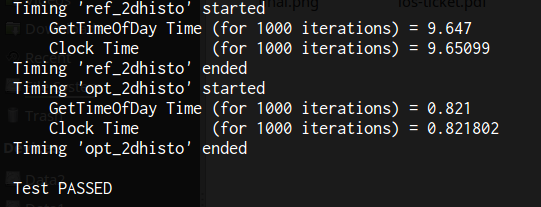
Qinglin Li, Xiangyu Ji

In this assignment, we tried to solve histogram counting problem with CUDA.

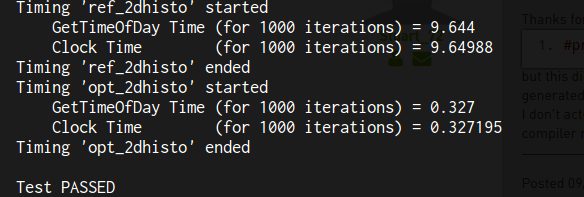
We first build a naïve implementation using atomic add and only one global histogram. We spent two hours implementing this naïve version because we didn’t notice the width is padded so we spent a lot of time figuring the right address of input elements. Here is the result, we tested our program on a GTX 980M GPU.



Next, we use privatization for speed histogram writing. We build sub-histograms for each thread block in shared memory and write sub-histograms into global histogram. We faced some difficulty finding the correct address because of width padding. We spent 20 minutes based on the naïve implementation. We achieved a 6X speedup, details can be seen in the following screenshot.



Finally, we reduce the global memory traffic by letting each thread process N(N=64) elements instead of process one element. So the write bandwidth reduced by a factor by N. The difficulty was to find a optimal N, we tried several settings (16, 32, 64) and decided to go with 64. We spent 20 minutes on this optimization. We achieved a 2.5X speedup.



We also tested the correctness in Wilkinson Lab’s GTX 680.

