EE 147: Lab 3*
Histogram

Due on May 9, 2018

Yixin Yang
862004384

Contents

*Template from www.tedpavlic.com

Problem 1

Use visual profiler to report relevant statistics (e.g. utilization and memory hierarchy related) about the execution of your kernels. Did you find any surprising results?

Solution

Run nvvp to open nvidia visual profiler. The results are shown below.

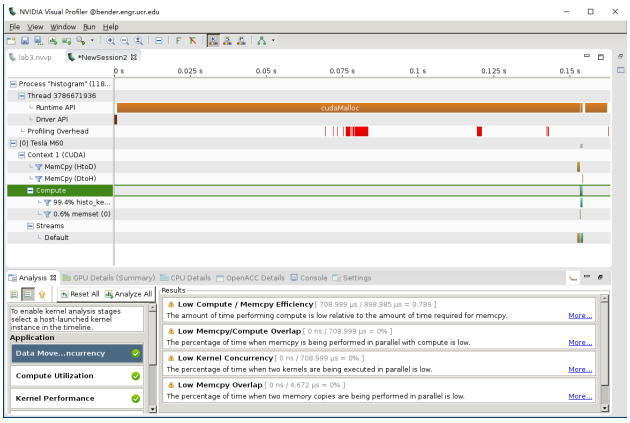

Figure 1: nvvp main window

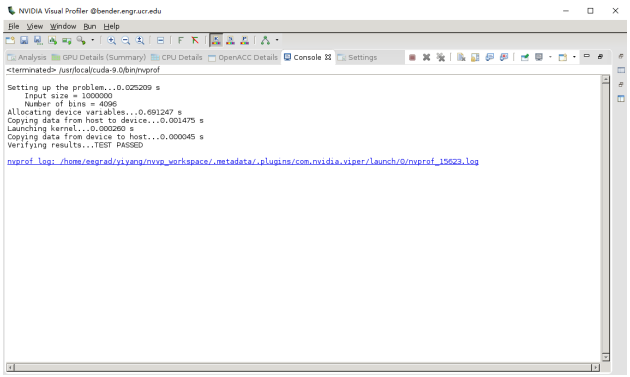

Figure 2: Console

The result of nvvp main window is shown as Fig.1.
The result of Console is shown as Fig.2.
The result of Data Movement and Concurrency is shown as Fig.3.
The result of Compute Utilization is shown as Fig.4.
The result of Kernel Performance is shown as Fig.5.
The result of Dependency Analysis is shown as Fig.6.
The result of Memory Efficiency is shown as Fig.7.

Problem 1 continued on next page...

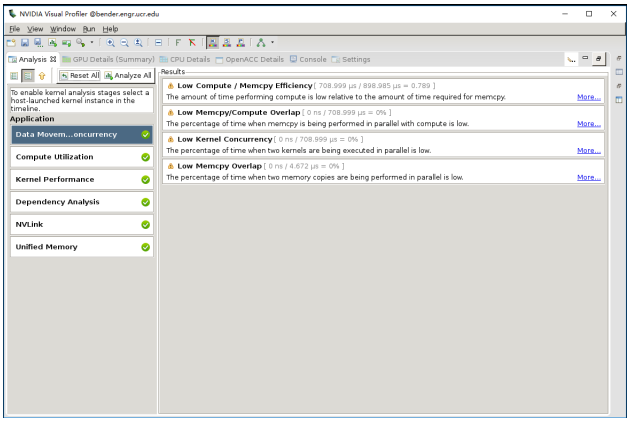

Figure 3: Data Movement and Concurrency

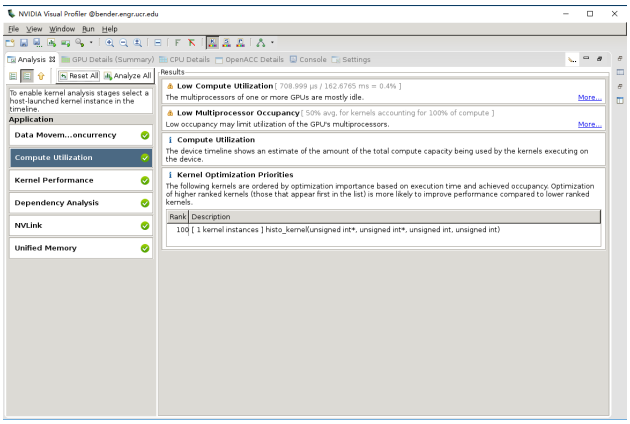

Figure 4: Compute Utilization

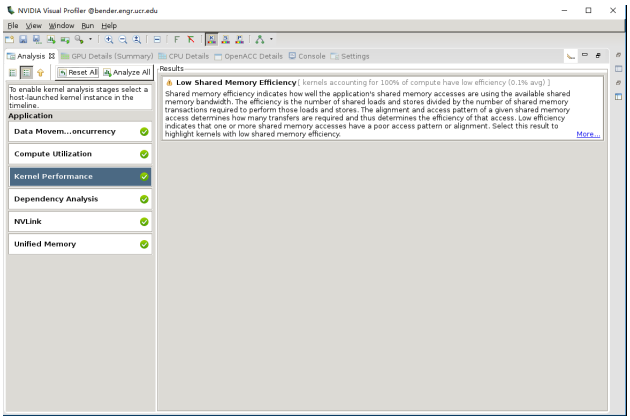

Figure 5: Kernel Performance

I am surprised that the compute utilization is only 0.4 and cudaMalloc occupies most of the execution time. In the data movement and concurrency application, compute/memcpy efficiency, memcpy/compute overlap, kernel concurrency, memcpy throughput, and memcpy overlap are all in a low degree.
In the compute utilization application, compute utilization and multiprocessor occupancy are all in a low degree.

Problem 1 continued on next page. . .

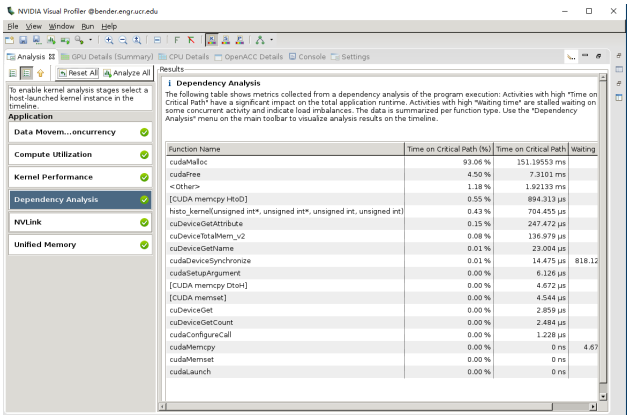

Figure 6: Dependency Analysis

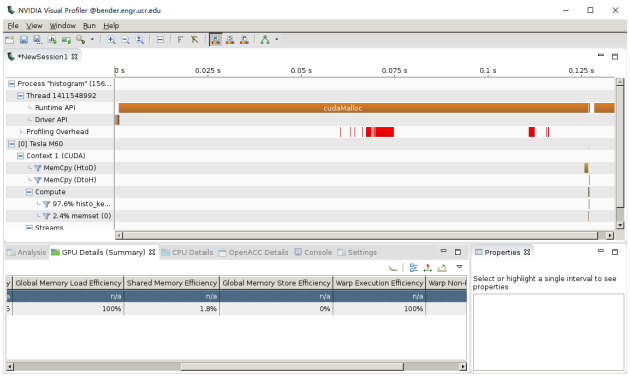

Figure 7: Memory Efficiency

In the kernel performance application, global memory efficiency is in a high degree but shared memory efficiency is in a low degree.
A possible reason for this is that the histogram is relatively easy for GPU to compute. So, memory allocation occupies most parts of time.

Problem 2

What, if any, limitations are there on m and n for your implementation? Explain these limitations and how you may overcome them with a different implementation.
Solution
My code works for a large scale of m and n.
There is no limitation for m.
The maximum value for n is 12288.
Because of hardware limitation, the the maximum size of shared memory is 12288.
One way to overcome this limitation is do not use shared memeory.