

Name: Ge Yuhao
NetID: Yuhaoge2@illinois.edu
Section: Z11/Z12

ECE 408/CS483 Milestone 2 Report

1. Show output of rai running Mini-DNN on the basic GPU convolution implementation for batch size of 1k images. This can either be a screen capture or a text copy of the running output. Please do not show the build output. (The running output should be everything including and after the line "*Loading fashion-mnist data...Done*").

```
Test batch size: 1000
Loading fashion-mnist data...Done
Loading model...Done
Conv-GPU==
Layer Time: 65.4718 ms
Op Time: 1.63886 ms
Conv-GPU==
Layer Time: 53.3757 ms
Op Time: 6.25089 ms

Test Accuracy: 0.886
```

```
real    0m9.673s
user    0m9.343s
sys     0m0.300s
```

2. For the basic GPU implementation, list Op Times, whole program execution time, and accuracy for batch size of 100, 1k, and 10k images.

Batch Size	Op Time 1	Op Time 2	Total Execution Time	Accuracy
100	0.174333ms	0.630749ms	1.216s	0.86
1000	1.63886ms	6.25089ms	9.673s	0.886
10000	16.0759ms	62.9824ms	1m34.672s	0.8714

3. List all the kernels that collectively consumed more than 90% of the kernel time and what percentage of the kernel time each kernel did consume (start with the kernel that consumed the most time, then list the next kernel, until you reach 90% or more).

99.9% conv_forward_kernel

4. List all the CUDA API calls that collectively consumed more than 90% of the API time and what percentage of the API time each call did consume (start with the API call that consumed the most time, then list the next call, until you reach 90% or more).

Batch size = 1000:
68.8% cudaMalloc
28.8% cudaMemcpy

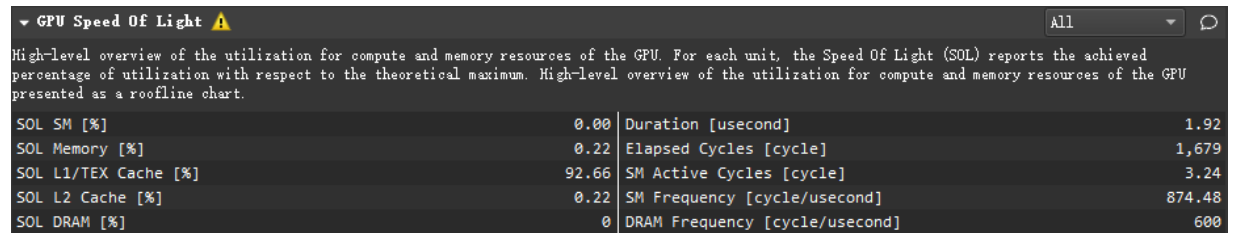
Batch size = 10000:
76.2% cudaMemcpy
16.7% cudaMalloc

5. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both.

The CUDA API is the build in function such as cudaMalloc and cudaMemcpy which are called by the main function.
The Kernel is the cuda function we defined in our code which are also called by the main function.

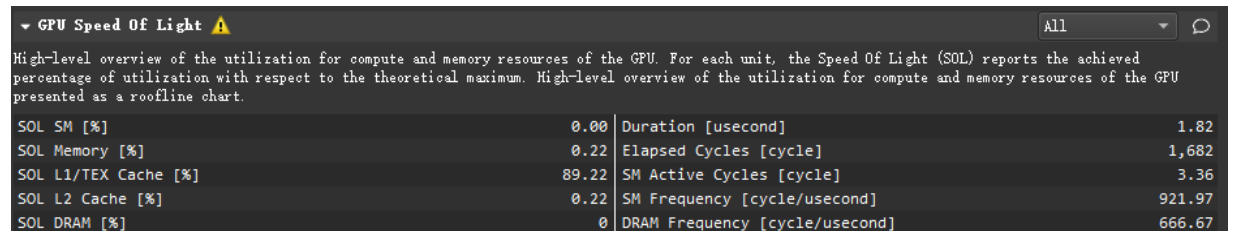
6. Show a screenshot of the GPU SOL utilization

Batch size = 1000:



SOL SM [%]	0.00	Duration [usecond]	1.92
SOL Memory [%]	0.22	Elapsed Cycles [cycle]	1,679
SOL L1/TEX Cache [%]	92.66	SM Active Cycles [cycle]	3.24
SOL L2 Cache [%]	0.22	SM Frequency [cycle/usecond]	874.48
SOL DRAM [%]	0	DRAM Frequency [cycle/usecond]	600

Batch size = 10000



SOL SM [%]	0.00	Duration [usecond]	1.82
SOL Memory [%]	0.22	Elapsed Cycles [cycle]	1,682
SOL L1/TEX Cache [%]	89.22	SM Active Cycles [cycle]	3.36
SOL L2 Cache [%]	0.22	SM Frequency [cycle/usecond]	921.97
SOL DRAM [%]	0	DRAM Frequency [cycle/usecond]	666.67