Name: *Sheng-Wei Huang*
NetID: *Swhuang3*
Section: *AL2*

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: 99.5834 ms
Op Time: 1.62754 ms
Conv-GPU==
Layer Time: 81.0777 ms
Op Time: 6.26372 ms

Test Accuracy: 0.886


real    0m9.761s
user    0m9.418s
sys     0m0.308s
```

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	*0.176298ms*	*0.635443ms*	*1.156 s*	*0.86*
1000	*1.62754ms*	*6.26372ms*	*9.761 s*	*0.886*
10000	*16.0021ms*	*62.2321ms*	*95.227 s*	*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).

conv_forward_kernel: 100.0% time

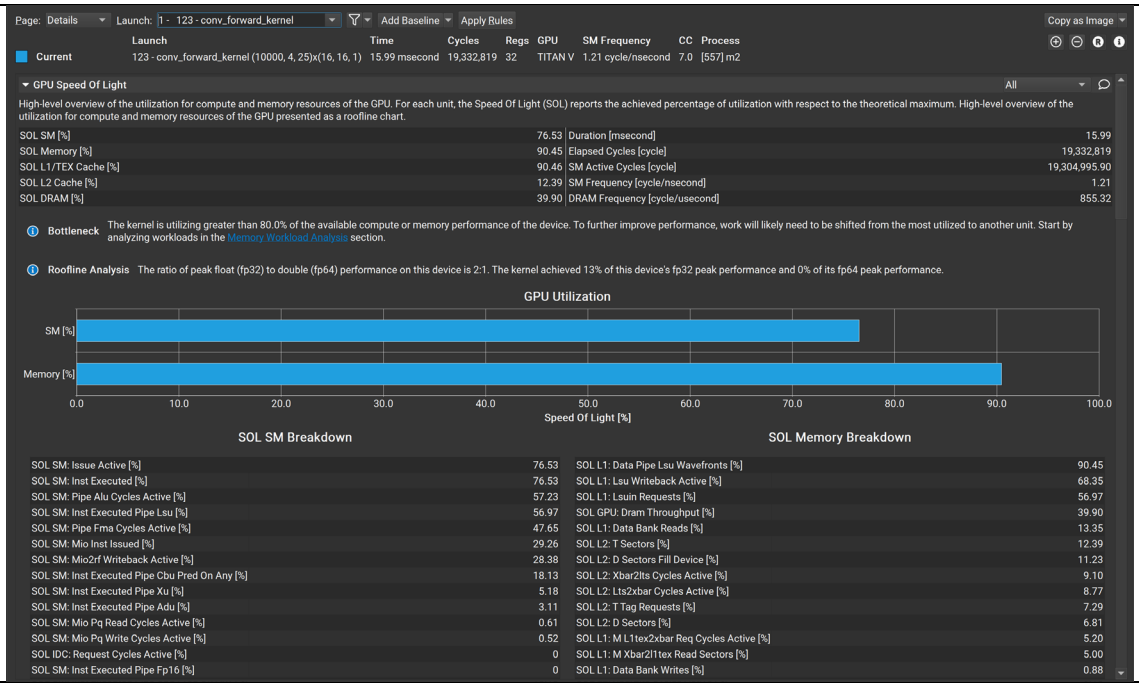
4. List all the CUDA API calls that collectively consumed more than 90% of the API time and what percentage of the API time each call did consume (start with the API call that consumed the most time, then list the next call, until you reach 90% or more).

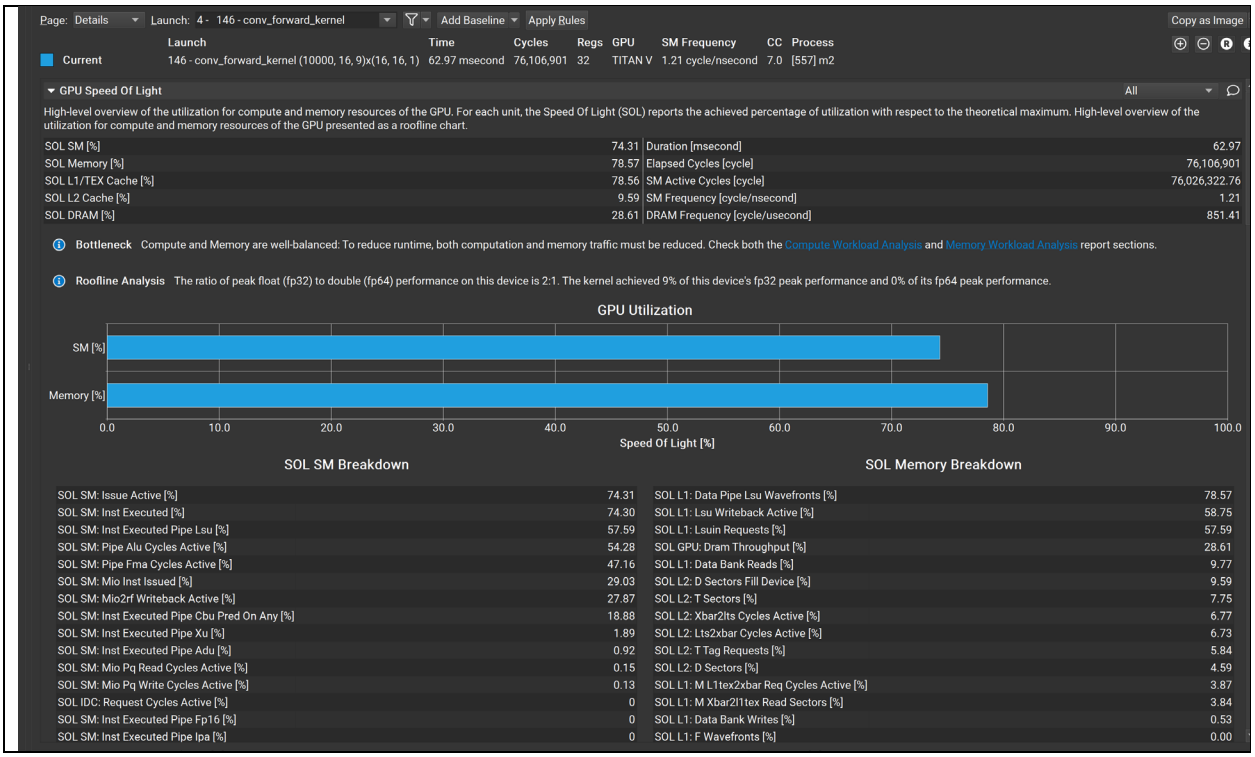
cudaMemcpy: 76.9%
cudaMalloc: 16.3%

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

Cuda kernels are user defined function with __global__ prefix, which can directly access GPU memory and use GPU to calculate, like __global__ conv_forward_kernel() we defined in this milestone.
Cuda API calls are the api cuda provided, which allow host to allocate or copy data to GPU device, like cudaMemcpy, cudaMalloc.

6. Show a screenshot of the GPU SOL utilization




Page: Details
Launch: 4 - 146 - conv_forward_kernel
Add Baseline
Apply Rules
Copy as Image
Launch
Time
Cycles
Regs
GPU
SM Frequency
CC
Process
Current
146 - conv_forward_kernel (10000, 16, 9)x(16, 16, 1)
62.97 msecond
76,106,901
32
TITAN V
1.21 cycle/nsecond
7.0
[557] m2
GPU Speed Of Light
All
High-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SOL) reports the achieved percentage of utilization with respect to the theoretical maximum. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.
SOL SM [%] 74.31
SOL Memory [%] 78.57
SOL L1/TEX Cache [%] 78.56
SOL L2 Cache [%] 9.59
SOL DRAM [%] 28.61
Duration [msecond] 62.97
Elapsed Cycles [cycle] 76,106,901
SM Active Cycles [cycle] 76,026,322.76
SM Frequency [cycle/nsecond] 1.21
DRAM Frequency [cycle/usecond] 851.41
Bottleneck Compute and Memory are well-balanced: To reduce runtime, both computation and memory traffic must be reduced. Check both the Compute Workload Analysis and Memory Workload Analysis report sections.
Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 2:1. The kernel achieved 9% of this device's fp32 peak performance and 0% of its fp64 peak performance.
GPU Utilization
SM [%]
Memory [%]
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
Speed Of Light [%]
SOL SM Breakdown
SOL SM: Issue Active [%] 74.31
SOL SM: Inst Executed [%] 74.30
SOL SM: Inst Executed Pipe Lsu [%] 57.59
SOL SM: Pipe Alu Cycles Active [%] 54.28
SOL SM: Pipe Fma Cycles Active [%] 47.16
SOL SM: Mio Inst Issued [%] 29.03
SOL SM: Mio2rf Writeback Active [%] 27.87
SOL SM: Inst Executed Pipe Cbu Pred On Any [%] 18.88
SOL SM: Inst Executed Pipe Xu [%] 1.89
SOL SM: Inst Executed Pipe Adu [%] 0.92
SOL SM: Mio Pq Read Cycles Active [%] 0.15
SOL SM: Mio Pq Write Cycles Active [%] 0.13
SOL IDC: Request Cycles Active [%] 0
SOL SM: Inst Executed Pipe Fp16 [%] 0
SOL SM: Inst Executed Pipe Ipa [%] 0
SOL Memory Breakdown
SOL L1: Data Pipe Lsu Wavefronts [%] 78.57
SOL L1: Lsu Writeback Active [%] 58.75
SOL L1: Lsuin Requests [%] 57.59
SOL GPU: Dram Throughput [%] 28.61
SOL L1: Data Bank Reads [%] 9.77
SOL L2: D Sectors Fill Device [%] 9.59
SOL L2: T Sectors [%] 7.75
SOL L2: Xbar2lts Cycles Active [%] 6.77
SOL L2: Lts2xbar Cycles Active [%] 6.73
SOL L2: T Tag Requests [%] 5.84
SOL L2: D Sectors [%] 4.59
SOL L1: M L1tex2xbar Req Cycles Active [%] 3.87
SOL L1: M Xbar2l1tex Read Sectors [%] 3.84
SOL L1: Data Bank Writes [%] 0.53
SOL L1: F Wavefronts [%] 0.00