Name: *Qian Liyang*
NetID: *3190110719*
Section: *ZJUI*

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: 62.5606 ms
Op Time: 1.69811 ms
Conv-GPU==
Layer Time: 50.8151 ms
Op Time: 6.47096 ms

Test Accuracy: 0.886
```

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	*0.182508 ms*	*0.6552 ms*	*0m1.143s*	*0.86*
1000	*1.69811ms*	*6.47096ms*	*0m9.645s*	*0.886*
10000	*16.7434 ms*	*64.4371 ms*	*1m37.553s*	*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).

```
Generating CUDA Memory Operation Statistics...
CUDA Kernel Statistics (nanoseconds)

Time(%)      Total Time   Instances         Average         Minimum         Maximum  Name
-------  --------------  ----------  --------------  --------------  --------------  ----------------------------------
-----------------------------------------
  100.0        81608506           2      40804253.0        16774710        64833796  conv_forward_kernel

    0.0            2880           2          1440.0            1376            1504  prefn_marker_kernel

    0.0            2560           2          1280.0            1280            1280  do_not_remove_this_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).

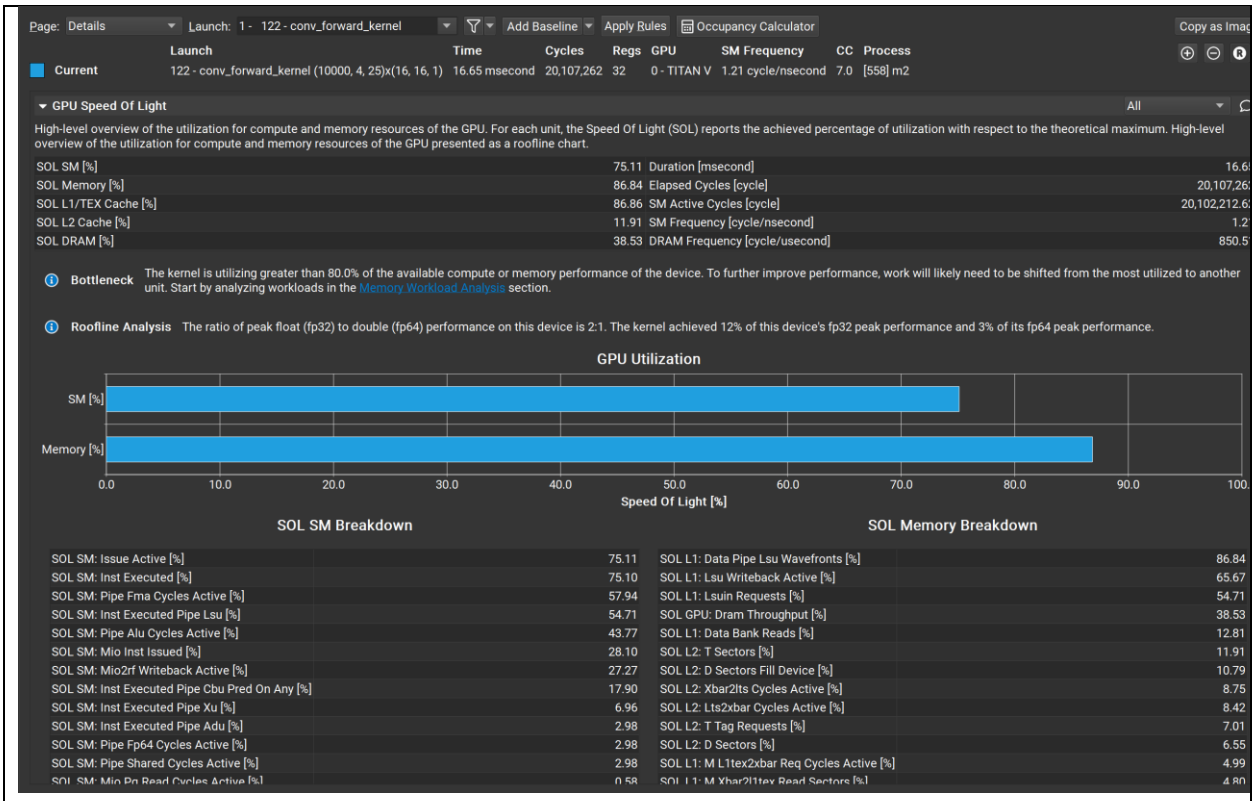
```
Generating CUDA API Statistics...
CUDA API Statistics (nanoseconds)

Time(%)      Total Time       Calls         Average         Minimum         Maximum  Name
-------  --------------  ----------  --------------  --------------  --------------  ----------------------------------
-------------------------------------------
   68.1      1183718378           8     147964797.3           43349       561717210  cudaMemcpy

   26.0       451257818           8      56407227.2          201793       441706478  cudaMalloc

    4.7        81646576           8      10205822.0             980        64837910  cudaDeviceSynchronize

    1.0        17701844           6       2950307.3           18529        17551338  cudaLaunchKernel

    0.2         2925108           8        365638.5          158512          757110  cudaFree
```

More than 90% is cudaMecpy and cudaMalloc

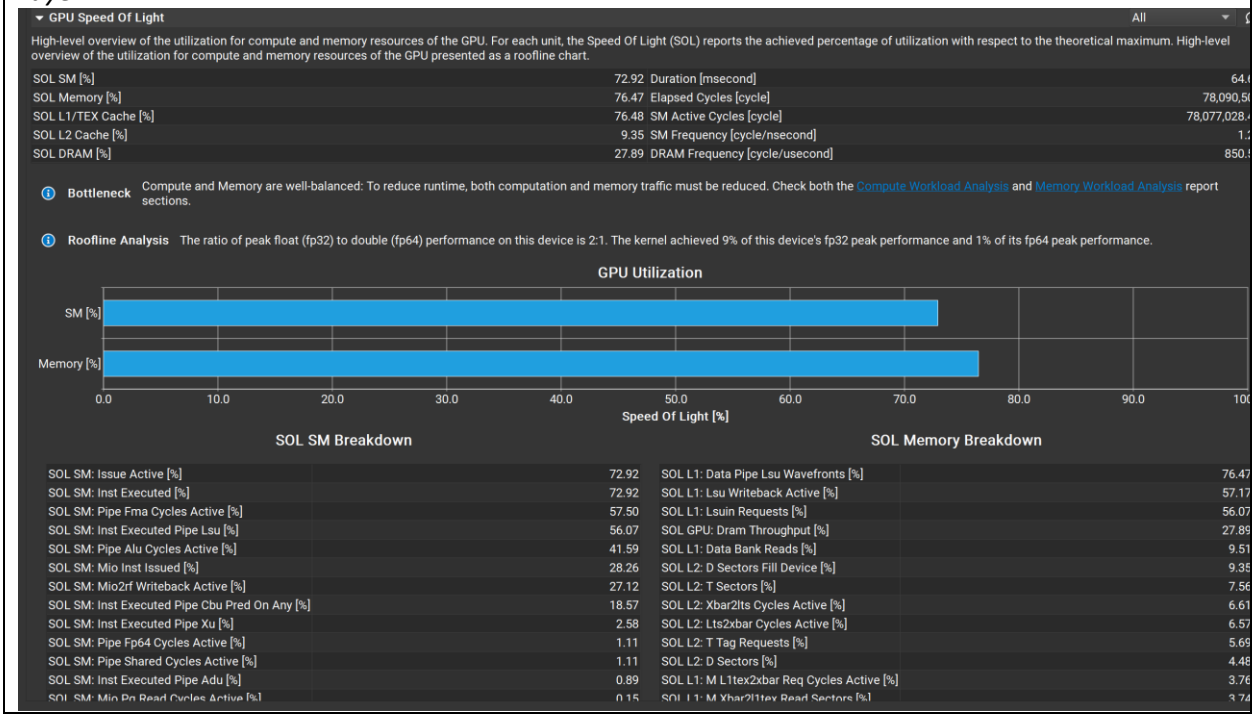
5. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both.
    1. CUDA API calls for data transferring is larger than the kernel data transferring. For example, in above figures, the average for cudaMemcpy is 1.5e9 ns but for kernel, the total running time is just 4.1e8, which is only 27% of cudaMemcpy. That's because the gpu have larger bandwidth than the main memory for host on most machines.
    2. Cuda API is called by the CPU and kernerl is called by GPU. The CUDA API interface in nsys shows the functions running on cpu, which are related to CUDA library, for example, the cudaMemcpy . The CUDA Kernel interface shows the functions running on GPU, for example, the conv_forward_kernel

6. Show a screenshot of the GPU SOL utilization

Note: the screen shot is for 10,000 data size
Layer 1:


Page: Details
Launch: 1 - 122 - conv_forward_kernel
Add Baseline
Apply Rules
Occupancy Calculator
Copy as Imag
Launch Time Cycles Regs GPU SM Frequency CC Process
Current 122 - conv_forward_kernel (10000, 4, 25)x(16, 16, 1) 16.65 msecond 20,107,262 32 0 - TITAN V 1.21 cycle/nsecond 7.0 [558] 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 [%] 75.11
SOL Memory [%] 86.84
SOL L1/TEX Cache [%] 86.86
SOL L2 Cache [%] 11.91
SOL DRAM [%] 38.53
Duration [msecond] 16.6
Elapsed Cycles [cycle] 20,107,26
SM Active Cycles [cycle] 20,102,212.6
SM Frequency [cycle/nsecond] 1.2
DRAM Frequency [cycle/usecond] 850.5
Bottleneck The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the most utilized to another unit. Start by analyzing workloads in the Memory Workload Analysis section.
Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 2:1. The kernel achieved 12% of this device's fp32 peak performance and 3% 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.
Speed Of Light [%]
SOL SM Breakdown
SOL SM: Issue Active [%] 75.11
SOL SM: Inst Executed [%] 75.10
SOL SM: Pipe Fma Cycles Active [%] 57.94
SOL SM: Inst Executed Pipe Lsu [%] 54.71
SOL SM: Pipe Alu Cycles Active [%] 43.77
SOL SM: Mio Inst Issued [%] 28.10
SOL SM: Mio2rf Writeback Active [%] 27.27
SOL SM: Inst Executed Pipe Cbu Pred On Any [%] 17.90
SOL SM: Inst Executed Pipe Xu [%] 6.96
SOL SM: Inst Executed Pipe Adu [%] 2.98
SOL SM: Pipe Fp64 Cycles Active [%] 2.98
SOL SM: Pipe Shared Cycles Active [%] 2.98
SOL SM: Mio Pp Read Cycles Active [%] 0.58
SOL Memory Breakdown
SOL L1: Data Pipe Lsu Wavefronts [%] 86.84
SOL L1: Lsu Writeback Active [%] 65.67
SOL L1: Lsuin Requests [%] 54.71
SOL GPU: Dram Throughput [%] 38.53
SOL L1: Data Bank Reads [%] 12.81
SOL L2: T Sectors [%] 11.91
SOL L2: D Sectors Fill Device [%] 10.79
SOL L2: Xbar2lts Cycles Active [%] 8.75
SOL L2: Lts2xbar Cycles Active [%] 8.42
SOL L2: T Tag Requests [%] 7.01
SOL L2: D Sectors [%] 6.55
SOL L1: M L1tex2xbar Req Cycles Active [%] 4.99
SOL L1: M Xbar2l1tex Read Sectors [%] 4.80

Layer 2:


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 [%] 72.92
SOL Memory [%] 76.47
SOL L1/TEX Cache [%] 76.48
SOL L2 Cache [%] 9.35
SOL DRAM [%] 27.89
Duration [msecond] 64.
Elapsed Cycles [cycle] 78,090,5
SM Active Cycles [cycle] 78,077,028.
SM Frequency [cycle/nsecond] 1.2
DRAM Frequency [cycle/usecond] 850.5
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 1% 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
Speed Of Light [%]
SOL SM Breakdown
SOL SM: Issue Active [%] 72.92
SOL SM: Inst Executed [%] 72.92
SOL SM: Pipe Fma Cycles Active [%] 57.50
SOL SM: Inst Executed Pipe Lsu [%] 56.07
SOL SM: Pipe Alu Cycles Active [%] 41.59
SOL SM: Mio Inst Issued [%] 28.26
SOL SM: Mio2rf Writeback Active [%] 27.12
SOL SM: Inst Executed Pipe Cbu Pred On Any [%] 18.57
SOL SM: Inst Executed Pipe Xu [%] 2.58
SOL SM: Pipe Fp64 Cycles Active [%] 1.11
SOL SM: Pipe Shared Cycles Active [%] 1.11
SOL SM: Inst Executed Pipe Adu [%] 0.89
SOL SM: Mio Pp Read Cycles Active [%] 0.15
SOL Memory Breakdown
SOL L1: Data Pipe Lsu Wavefronts [%] 76.47
SOL L1: Lsu Writeback Active [%] 57.17
SOL L1: Lsuin Requests [%] 56.07
SOL GPU: Dram Throughput [%] 27.89
SOL L1: Data Bank Reads [%] 9.51
SOL L2: D Sectors Fill Device [%] 9.35
SOL L2: T Sectors [%] 7.56
SOL L2: Xbar2lts Cycles Active [%] 6.61
SOL L2: Lts2xbar Cycles Active [%] 6.57
SOL L2: T Tag Requests [%] 5.69
SOL L2: D Sectors [%] 4.48
SOL L1: M L1tex2xbar Req Cycles Active [%] 3.76
SOL L1: M Xbar2l1tex Read Sectors [%] 3.74