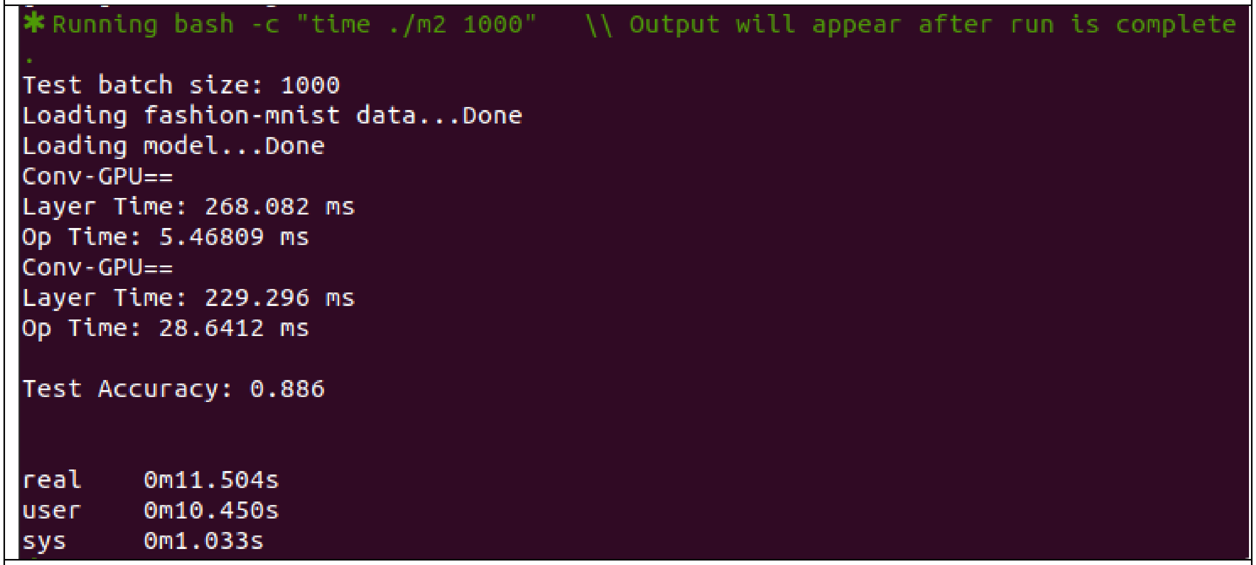
Name: Mingyuan Zhao
NetID: mz55
Section: OD1

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*").



```
✱Running bash -c "time ./m2 1000"   \\ Output will appear after run is complete
.
Test batch size: 1000
Loading fashion-mnist data...Done
Loading model...Done
Conv-GPU==
Layer Time: 268.082 ms
Op Time: 5.46809 ms
Conv-GPU==
Layer Time: 229.296 ms
Op Time: 28.6412 ms

Test Accuracy: 0.886


real    0m11.504s
user    0m10.450s
sys     0m1.033s
```

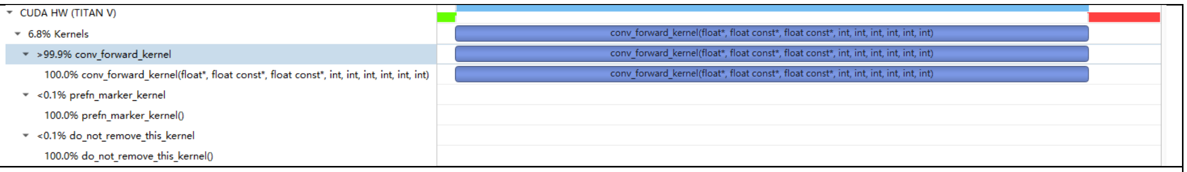
2. For the basic GPU implementation, list Op Times, whole program execution time, and accuracy for batch size of 100, 1k, and 10k images.

Batch Size	Op Time 1	Op Time 2	Total Execution Time	Accuracy
100	0.17625ms	1.23577ms	0m2.376s	0.86
1000	5.46809ms	28.6412ms	0m11.504s	0.886
10000	16.1087ms	63.3525ms	1m37.329s	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).

This is the only kernel and being called twice:

- conv_forward_kernel



```
▾ CUDA HW (TITAN V)
  ▾ 6.8% Kernels
    ▾ >99.9% conv_forward_kernel
        100.0% conv_forward_kernel(float*, float const*, float const*, int, int, int, int, int, int)
    ▾ <0.1% prefn_marker_kernel
        100.0% prefn_marker_kernel()
    ▾ <0.1% do_not_remove_this_kernel
        100.0% do_not_remove_this_kernel()
```

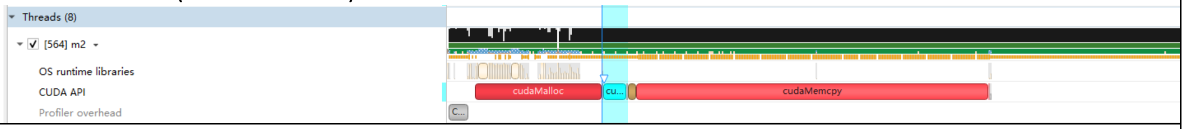
conv_forward_kernel(float*, float const*, float const*, int, int, int, int, int, int)
conv_forward_kernel(float*, float const*, float const*, int, int, int, int, int, int)
conv_forward_kernel(float*, float const*, float const*, int, int, int, int, int, int)

-

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).

There are 4 different CUDA API calls, the ones that consume 90% of the API time are:

- cudaMemcpy (called 8 times)
- cudaMalloc (called 8 times)



```
▾ Threads (8)
  ▾ ☑ [564] m2 ▾
      OS runtime libraries
      CUDA API
      Profiler overhead
```

-

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

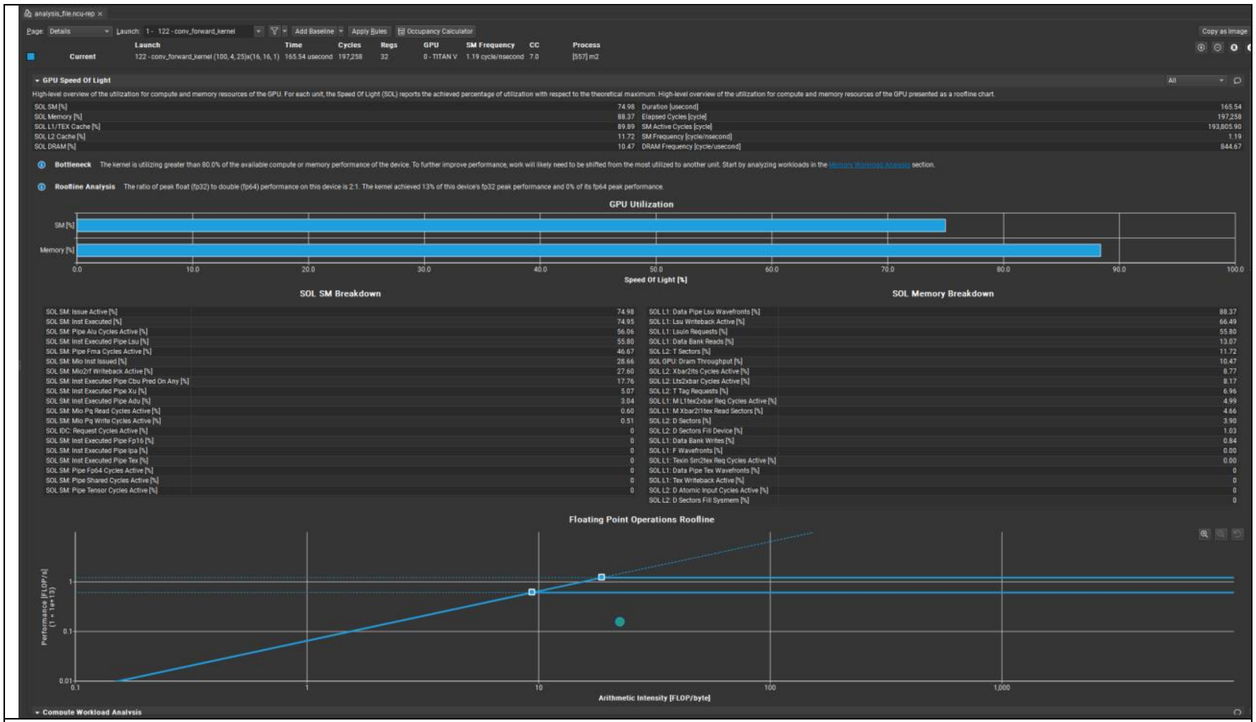
A kernel is a C++ function. When called, “are executed N times in parallel by N different CUDA threads, as opposed to only once like regular C++ functions.”

An API is a set of definitions, protocols and tools for building software. In CUDA, “it provides C and C++ functions that execute on the host to allocate and deallocate device memory, transfer data between host memory and device memory, manage systems with multiple devices, etc.”

The CUDA API call’s part is the section contains the time of CPU using, while the profiling result is the time that the GPUs taking. The total time of the CUDA API call is from the moment it is launched to the moment it is completed, so will overlap with executing kernels.

6. Show a screenshot of the GPU SOL utilization

Kernel1:



	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	122 - conv_forward_kernel (100, 4, 25)x(16, 16, 1)	165.54 usecond	197,258	32	0 - TITAN V	1.19 cycle/nsecond	7.0	[557] m2

GPU Speed Of Light

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 [%]	74.98	Duration [usecond]	165.54
SOL Memory [%]	88.37	Elapsed Cycles [cycle]	197,258
SOL L1/TEX Cache [%]	89.89	SM Active Cycles [cycle]	193,805.90
SOL L2 Cache [%]	11.72	SM Frequency [cycle/nsecond]	1.19
SOL DRAM [%]	10.47	DRAM Frequency [cycle/usecond]	844.67

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 13% of this device's fp32 peak performance and 0% of its fp64 peak performance.

GPU Utilization

SOL SM Breakdown

SOL SM: Issue Active [%]	74.98
SOL SM: Inst Executed [%]	74.95
SOL SM: Pipe Alu Cycles Active [%]	56.06
SOL SM: Inst Executed Pipe Lsu [%]	55.80
SOL SM: Pipe Fma Cycles Active [%]	46.67
SOL SM: Mio Inst Issued [%]	28.66
SOL SM: Mio2rf Writeback Active [%]	27.60
SOL SM: Inst Executed Pipe Cbu Pred On Any [%]	17.76
SOL SM: Inst Executed Pipe Xu [%]	5.07
SOL SM: Inst Executed Pipe Adu [%]	3.04
SOL SM: Mio Pq Read Cycles Active [%]	0.60
SOL SM: Mio Pq Write Cycles Active [%]	0.51
SOL IDC: Request Cycles Active [%]	0
SOL SM: Inst Executed Pipe Fp16 [%]	0
SOL SM: Inst Executed Pipe Ipa [%]	0
SOL SM: Inst Executed Pipe Tex [%]	0
SOL SM: Pipe Fp64 Cycles Active [%]	0
SOL SM: Pipe Shared Cycles Active [%]	0
SOL SM: Pipe Tensor Cycles Active [%]	0

SOL Memory Breakdown

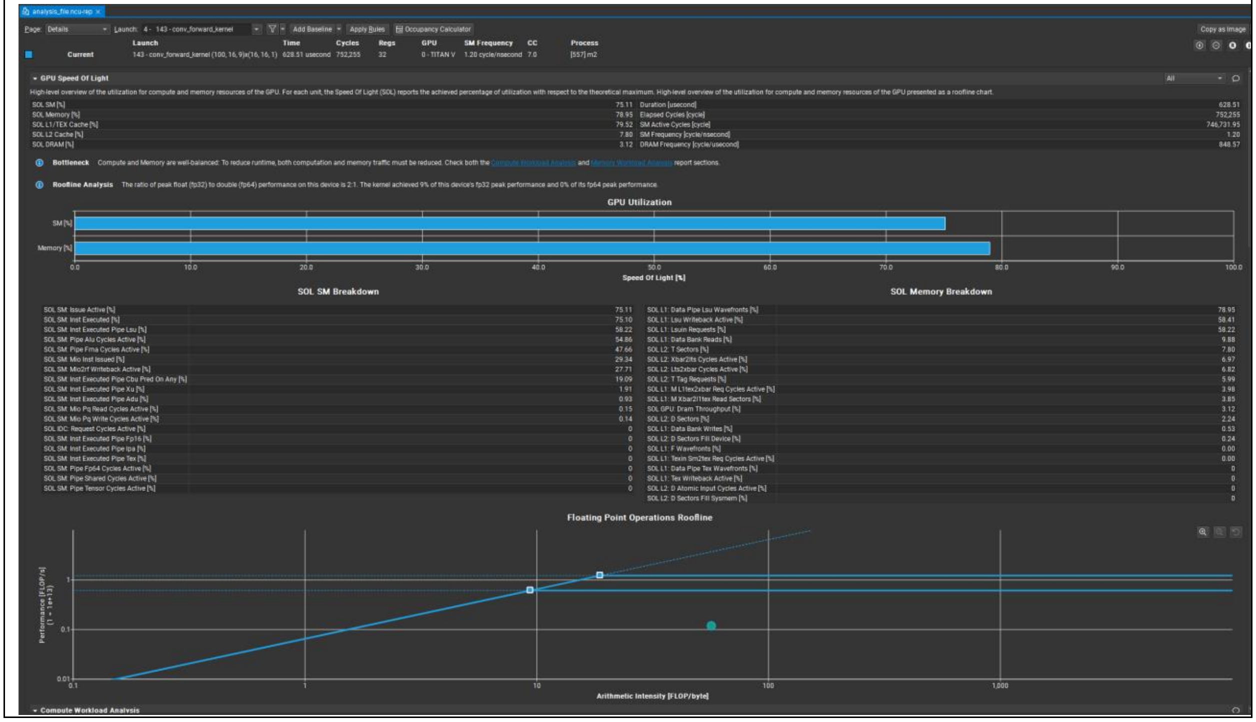
SOL L1: Data Pipe Lsu Wavefronts [%]	88.37
SOL L1: Lsu Writeback Active [%]	66.49
SOL L1: Lsuin Requests [%]	55.80
SOL L1: Data Bank Reads [%]	13.07
SOL L2: T Sectors [%]	11.72
SOL GPU: Dram Throughput [%]	10.47
SOL L2: Xbar2lts Cycles Active [%]	8.77
SOL L2: Lts2xbar Cycles Active [%]	8.17
SOL L2: T Tag Requests [%]	6.96
SOL L1: M L1tex2xbar Req Cycles Active [%]	4.99
SOL L1: M Xbar2l1tex Read Sectors [%]	4.66
SOL L2: D Sectors [%]	3.90
SOL L2: D Sectors Fill Device [%]	1.03
SOL L1: Data Bank Writes [%]	0.84
SOL L1: F Wavefronts [%]	0.00
SOL L1: Texin Sm2tex Req Cycles Active [%]	0.00
SOL L1: Data Pipe Tex Wavefronts [%]	0
SOL L1: Tex Writeback Active [%]	0
SOL L2: D Atomic Input Cycles Active [%]	0
SOL L2: D Sectors Fill Sysmem [%]	0

Floating Point Operations Roofline

Performance [FLOP/s] (1 = 1e+13)
1
0.1
0.01
0.1
1
10
100
1,000
Arithmetic Intensity [FLOP/byte]

Compute Workload Analysis

Kernel2:



	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	143 - conv_forward_kernel (100, 16, 9)x(16, 16, 1)	628.51 usecond	752,255	32	0 - TITAN V	1.20 cycle/nsecond	7.0	[557] m2

GPU Speed Of Light

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	Duration [usecond]	628.51
SOL Memory [%]	78.95	Elapsed Cycles [cycle]	752,255
SOL L1/TEX Cache [%]	79.52	SM Active Cycles [cycle]	746,731.95
SOL L2 Cache [%]	7.80	SM Frequency [cycle/nsecond]	1.20
SOL DRAM [%]	3.12	DRAM Frequency [cycle/usecond]	848.57

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

SOL SM Breakdown

SOL SM: Issue Active [%]	75.11
SOL SM: Inst Executed [%]	75.10
SOL SM: Inst Executed Pipe Lsu [%]	58.22
SOL SM: Pipe Alu Cycles Active [%]	54.86
SOL SM: Pipe Fma Cycles Active [%]	47.66
SOL SM: Mio Inst Issued [%]	29.34
SOL SM: Mio2rf Writeback Active [%]	27.71
SOL SM: Inst Executed Pipe Cbu Pred On Any [%]	19.09
SOL SM: Inst Executed Pipe Xu [%]	1.91
SOL SM: Inst Executed Pipe Adu [%]	0.93
SOL SM: Mio Pq Read Cycles Active [%]	0.15
SOL SM: Mio Pq Write Cycles Active [%]	0.14
SOL IDC: Request Cycles Active [%]	0
SOL SM: Inst Executed Pipe Fp16 [%]	0
SOL SM: Inst Executed Pipe Ipa [%]	0
SOL SM: Inst Executed Pipe Tex [%]	0
SOL SM: Pipe Fp64 Cycles Active [%]	0
SOL SM: Pipe Shared Cycles Active [%]	0
SOL SM: Pipe Tensor Cycles Active [%]	0

SOL Memory Breakdown

SOL L1: Data Pipe Lsu Wavefronts [%]	78.95
SOL L1: Lsu Writeback Active [%]	58.41
SOL L1: Lsuin Requests [%]	58.22
SOL L1: Data Bank Reads [%]	9.88
SOL L2: T Sectors [%]	7.80
SOL L2: Xbar2lts Cycles Active [%]	6.97
SOL L2: Lts2xbar Cycles Active [%]	6.82
SOL L2: T Tag Requests [%]	5.99
SOL L1: M L1tex2xbar Req Cycles Active [%]	3.98
SOL L1: M Xbar2l1tex Read Sectors [%]	3.85
SOL GPU: Dram Throughput [%]	3.12
SOL L2: D Sectors [%]	2.24
SOL L1: Data Bank Writes [%]	0.53
SOL L2: D Sectors Fill Device [%]	0.24
SOL L1: F Wavefronts [%]	0.00
SOL L1: Texin Sm2tex Req Cycles Active [%]	0.00
SOL L1: Data Pipe Tex Wavefronts [%]	0
SOL L1: Tex Writeback Active [%]	0
SOL L2: D Atomic Input Cycles Active [%]	0
SOL L2: D Sectors Fill Sysmem [%]	0

Floating Point Operations Roofline

Performance [FLOP/s] (1 = 1e+13)
1
0.1
0.01
0.1
1
10
100
1,000
Arithmetic Intensity [FLOP/byte]

Compute Workload Analysis