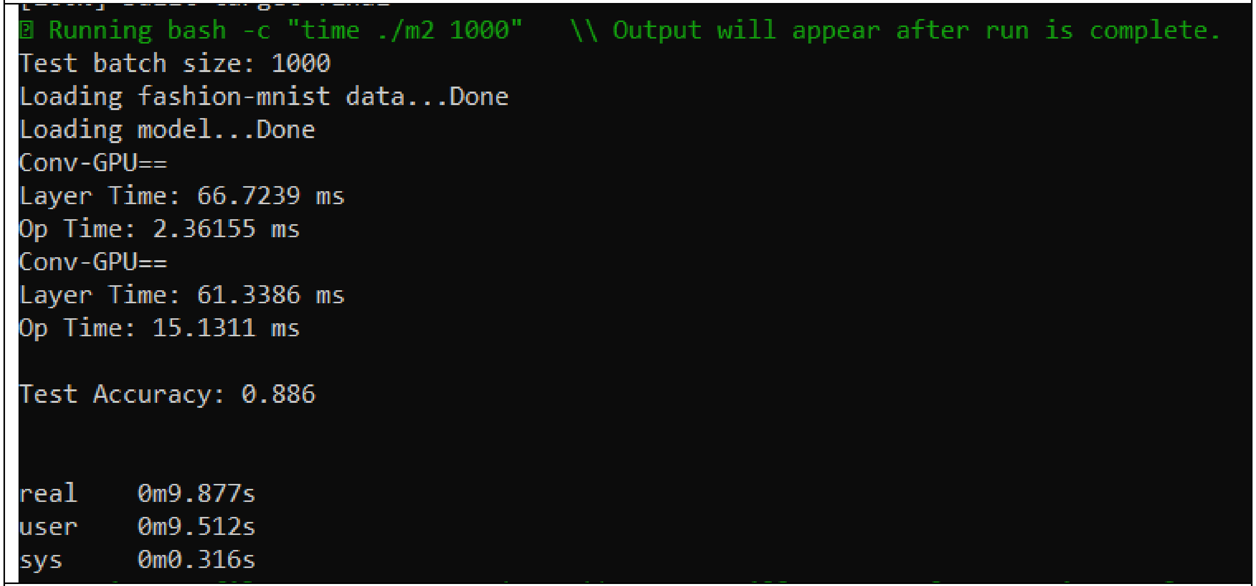
Name: *Lakshya Goyal*
NetID: *lgoyal3*
Section: *AB*

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: 66.7239 ms
Op Time: 2.36155 ms
Conv-GPU==
Layer Time: 61.3386 ms
Op Time: 15.1311 ms

Test Accuracy: 0.886


real    0m9.877s
user    0m9.512s
sys     0m0.316s
```

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	*0.249 ms*	*9.03 ms*	*1.181 s*	*86.00 %*
1000	*2.36 ms*	*15.13 ms*	*9.877 s*	*88.60 %*
10000	*23.27 ms*	*151.08 ms*	*1 min 38.494 s*	*87.14 %*

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 % → 169.8 ms

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).

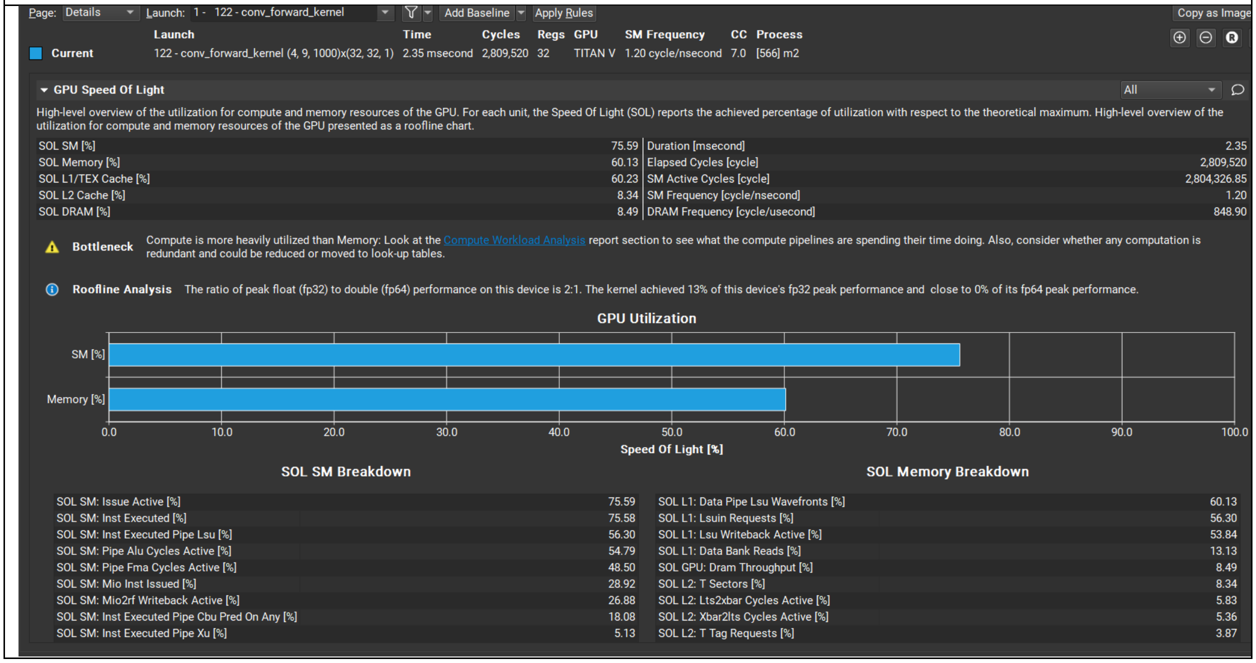
cudaMemcpy → 73.7 %
cudaMalloc → 13.0 %
cudaDeviceSynchronize → 11.4 %

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

The CUDA API calls are used by the CPU configure the kernel by either giving or getting data from the GPU. These execute on the CPU in order to do something to the GPU. Kernels run entirely on the GPU and so aren't affected by any other CPU code.

Conv_forward_kernel is a kernel that doesn't have any execution on the CPU, just the GPU. Whereas the cudaMemcpy() API function is called from the CPU to copy memory to the GPU.

6. Show a screenshot of the GPU SOL utilization



	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	122 - conv_forward_kernel (4, 9, 1000)x(32, 32, 1)	2.35 msecond	2,809,520	32	TITAN V	1.20 cycle/nsecond	7.0	[566] 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 [%]	75.59	Duration [msecond]	2.35
SOL Memory [%]	60.13	Elapsed Cycles [cycle]	2,809,520
SOL L1/TEX Cache [%]	60.23	SM Active Cycles [cycle]	2,804,326.85
SOL L2 Cache [%]	8.34	SM Frequency [cycle/nsecond]	1.20
SOL DRAM [%]	8.49	DRAM Frequency [cycle/usecond]	848.90

Bottleneck Compute is more heavily utilized than Memory: Look at the Compute Workload Analysis report section to see what the compute pipelines are spending their time doing. Also, consider whether any computation is redundant and could be reduced or moved to look-up tables.

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 close to 0% of its fp64 peak performance.

SOL SM Breakdown

SOL SM: Issue Active [%]	75.59
SOL SM: Inst Executed [%]	75.58
SOL SM: Inst Executed Pipe Lsu [%]	56.30
SOL SM: Pipe Alu Cycles Active [%]	54.79
SOL SM: Pipe Fma Cycles Active [%]	48.50
SOL SM: Mio Inst Issued [%]	28.92
SOL SM: Mio2rf Writeback Active [%]	26.88
SOL SM: Inst Executed Pipe Cbu Pred On Any [%]	18.08
SOL SM: Inst Executed Pipe Xu [%]	5.13

SOL Memory Breakdown

SOL L1: Data Pipe Lsu Wavefronts [%]	60.13
SOL L1: Lsuin Requests [%]	56.30
SOL L1: Lsu Writeback Active [%]	53.84
SOL L1: Data Bank Reads [%]	13.13
SOL GPU: Dram Throughput [%]	8.49
SOL L2: T Sectors [%]	8.34
SOL L2: Lts2xbar Cycles Active [%]	5.83
SOL L2: Xbar2lts Cycles Active [%]	5.36
SOL L2: T Tag Requests [%]	3.87