Name: *Chia-Chi, Chen*
NetID: *chiachi5*
Section: *AL1*

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*").

```
Loading fashion-mnist data...Done
Loading model...Done
Conv-GPU==
Layer Time: 77.0541 ms
Op Time: 5.9116 ms
Conv-GPU==
Layer Time: 64.0177 ms
Op Time: 21.3389 ms

Test Accuracy: 0.886


real  0m9.692s
user  0m9.385s
sys 0m0.264s
```

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	*0.576163ms*	*2.07154ms*	*1.155*	*0.86*
1000	*5.9116ms*	*21.3389ms*	*9.692s*	*0.886*
10000	*57.6449ms*	*216.506ms*	*1m35.722s*	*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).

```
100.0      266531896          2     133265948.0       59588959       206942937  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).

```
68.3     1035951641      8     129493955.1       18405      568180534  cudaMemcpy

17.6      266559704      6      44426617.3        2929      206945842  cudaDeviceSynchronize

12.9      196307942      8      24538492.7       67234      191712660  cudaMalloc
```

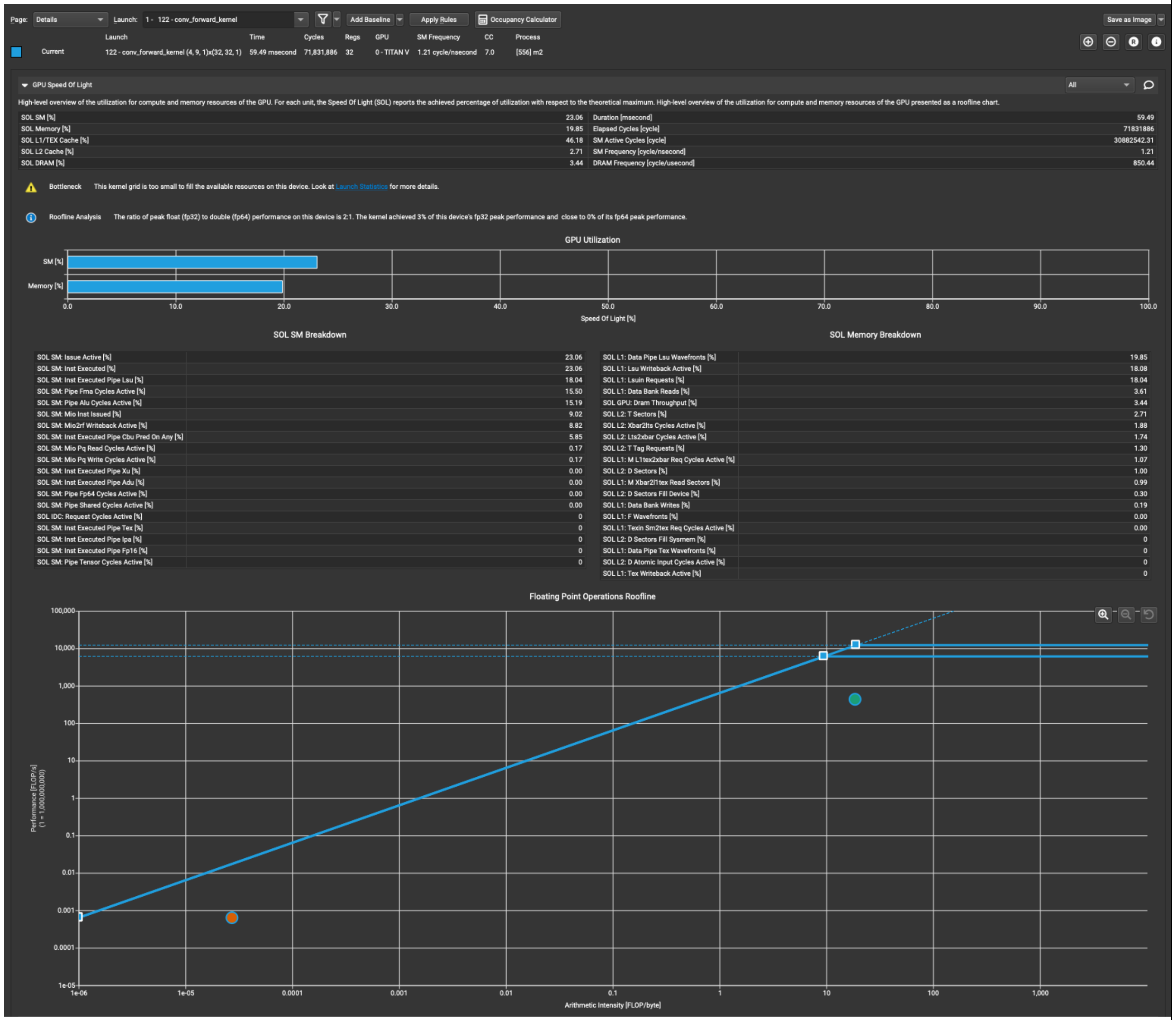
5. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both.

Kernel means the user code which will be run parallelized on the GPU device. CUDA API call, on the other hand, is a pre-defined function that generally interacts between the host and the device.

For example, the conv_forward_kernel is the kernel we implemented for our own convolution. It will be executed across several grids and blocks with multiple threads simultaneously. The cudaMemcpy is the CUDA API which we'll call whenever we need to copy data between the host and the device.

6. Show a screenshot of the GPU SOL utilization

1 - 122 - conv_forward_kernel



	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	122 - conv_forward_kernel (4, 9, 1)x(32, 32, 1)	59.49 msecond	71,831,886	32	0 - TITAN V	1.21 cycle/nsecond	7.0	[556] m2

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 [%]	23.06	Duration [msecond]	59.49
SOL Memory [%]	19.85	Elapsed Cycles [cycle]	71831886
SOL L1/TEX Cache [%]	46.18	SM Active Cycles [cycle]	30882542.31
SOL L2 Cache [%]	2.71	SM Frequency [cycle/nsecond]	1.21
SOL DRAM [%]	3.44	DRAM Frequency [cycle/usecond]	850.44

Bottleneck This kernel grid is too small to fill the available resources on this device. Look at Launch Statistics for more details.

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 2:1. The kernel achieved 3% of this device's fp32 peak performance and close to 0% of its fp64 peak performance.

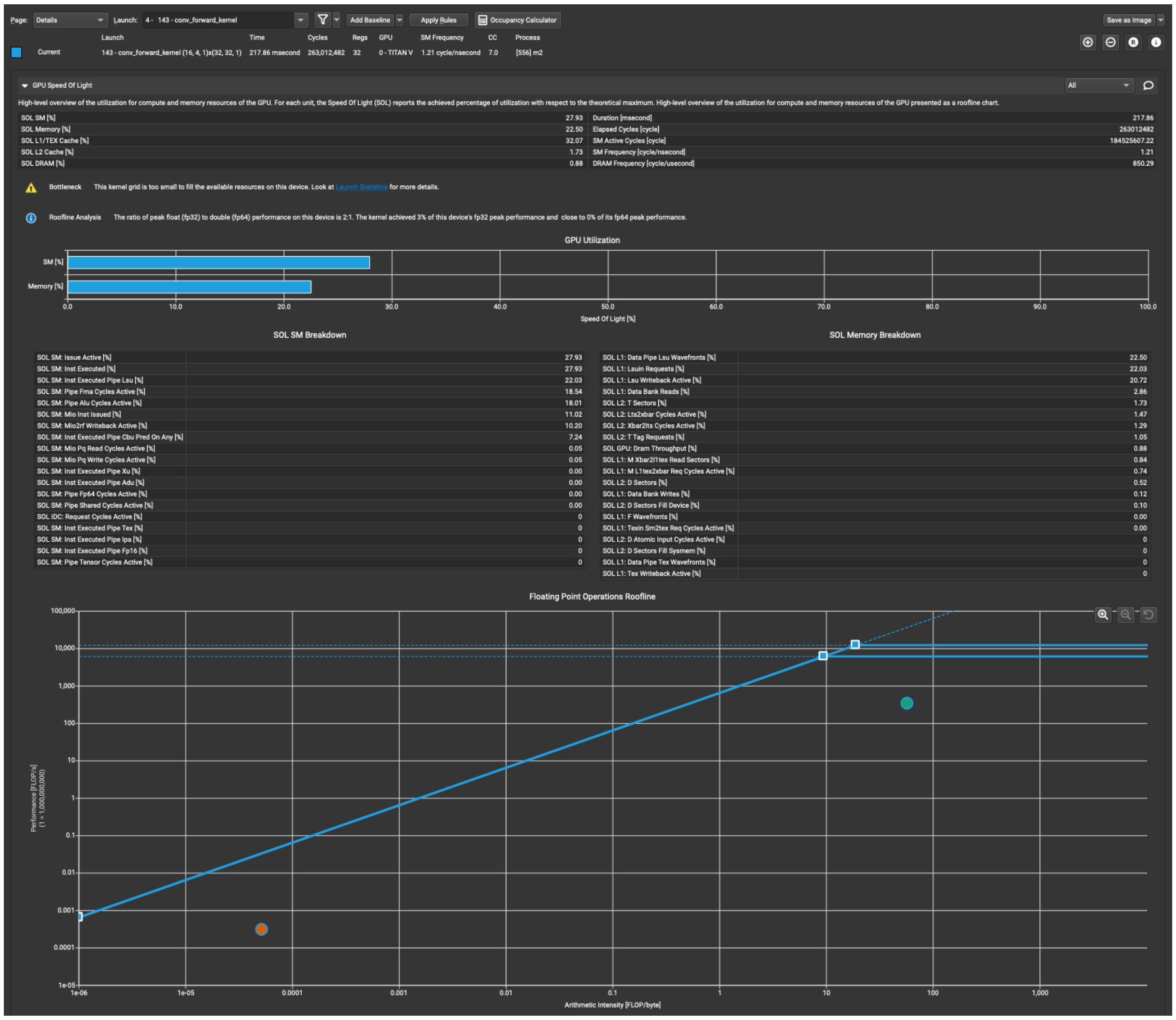
SOL SM Breakdown

SOL SM: Issue Active [%]	23.06
SOL SM: Inst Executed [%]	23.06
SOL SM: Inst Executed Pipe Lsu [%]	18.04
SOL SM: Pipe Fma Cycles Active [%]	15.50
SOL SM: Pipe Alu Cycles Active [%]	15.19
SOL SM: Mio Inst Issued [%]	9.02
SOL SM: Mio2rf Writeback Active [%]	8.82
SOL SM: Inst Executed Pipe Cbu Pred On Any [%]	5.85
SOL SM: Mio Pq Read Cycles Active [%]	0.17
SOL SM: Mio Pq Write Cycles Active [%]	0.17
SOL SM: Inst Executed Pipe Xu [%]	0.00
SOL SM: Inst Executed Pipe Adu [%]	0.00
SOL SM: Pipe Fp64 Cycles Active [%]	0.00
SOL SM: Pipe Shared Cycles Active [%]	0.00
SOL IDC: Request Cycles Active [%]	0
SOL SM: Inst Executed Pipe Tex [%]	0
SOL SM: Inst Executed Pipe Ipa [%]	0
SOL SM: Inst Executed Pipe Fp16 [%]	0
SOL SM: Pipe Tensor Cycles Active [%]	0

SOL Memory Breakdown

SOL L1: Data Pipe Lsu Wavefronts [%]	19.85
SOL L1: Lsu Writeback Active [%]	18.08
SOL L1: Lsuin Requests [%]	18.04
SOL L1: Data Bank Reads [%]	3.61
SOL GPU: Dram Throughput [%]	3.44
SOL L2: T Sectors [%]	2.71
SOL L2: Xbar2lts Cycles Active [%]	1.88
SOL L2: Lts2xbar Cycles Active [%]	1.74
SOL L2: T Tag Requests [%]	1.30
SOL L1: M L1tex2xbar Req Cycles Active [%]	1.07
SOL L2: D Sectors [%]	1.00
SOL L1: M Xbar2l1tex Read Sectors [%]	0.99
SOL L2: D Sectors Fill Device [%]	0.30
SOL L1: Data Bank Writes [%]	0.19
SOL L1: F Wavefronts [%]	0.00
SOL L1: Texin Sm2tex Req Cycles Active [%]	0.00
SOL L2: D Sectors Fill Sysmem [%]	0
SOL L1: Data Pipe Tex Wavefronts [%]	0
SOL L2: D Atomic Input Cycles Active [%]	0
SOL L1: Tex Writeback Active [%]	0

4 -   143 - conv_forward_kernel



	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
🟦 Current	143 - conv_forward_kernel (16, 4, 1)x(32, 32, 1)	217.86 msecond	263,012,482	32	0 - TITAN V	1.21 cycle/nsecond	7.0	[556] 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 [%]	27.93	Duration [msecond]	217.86
SOL Memory [%]	22.50	Elapsed Cycles [cycle]	263012482
SOL L1/TEX Cache [%]	32.07	SM Active Cycles [cycle]	184525607.22
SOL L2 Cache [%]	1.73	SM Frequency [cycle/nsecond]	1.21
SOL DRAM [%]	0.88	DRAM Frequency [cycle/usecond]	850.29

Bottleneck This kernel grid is too small to fill the available resources on this device. Look at Launch Statistics for more details.

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 2:1. The kernel achieved 3% of this device's fp32 peak performance and close to 0% of its fp64 peak performance.

SOL SM Breakdown

SOL SM: Issue Active [%]	27.93
SOL SM: Inst Executed [%]	27.93
SOL SM: Inst Executed Pipe Lsu [%]	22.03
SOL SM: Pipe Fma Cycles Active [%]	18.54
SOL SM: Pipe Alu Cycles Active [%]	18.01
SOL SM: Mio Inst Issued [%]	11.02
SOL SM: Mio2rf Writeback Active [%]	10.20
SOL SM: Inst Executed Pipe Cbu Pred On Any [%]	7.24
SOL SM: Mio Pq Read Cycles Active [%]	0.05
SOL SM: Mio Pq Write Cycles Active [%]	0.05
SOL SM: Inst Executed Pipe Xu [%]	0.00
SOL SM: Inst Executed Pipe Adu [%]	0.00
SOL SM: Pipe Fp64 Cycles Active [%]	0.00
SOL SM: Pipe Shared Cycles Active [%]	0.00
SOL IDC: Request Cycles Active [%]	0
SOL SM: Inst Executed Pipe Tex [%]	0
SOL SM: Inst Executed Pipe Ipa [%]	0
SOL SM: Inst Executed Pipe Fp16 [%]	0
SOL SM: Pipe Tensor Cycles Active [%]	0

SOL Memory Breakdown

SOL L1: Data Pipe Lsu Wavefronts [%]	22.50
SOL L1: Lsuin Requests [%]	22.03
SOL L1: Lsu Writeback Active [%]	20.72
SOL L1: Data Bank Reads [%]	2.86
SOL L2: T Sectors [%]	1.73
SOL L2: Lts2xbar Cycles Active [%]	1.47
SOL L2: Xbar2lts Cycles Active [%]	1.29
SOL L2: T Tag Requests [%]	1.05
SOL GPU: Dram Throughput [%]	0.88
SOL L1: M Xbar2l1tex Read Sectors [%]	0.84
SOL L1: M L1tex2xbar Req Cycles Active [%]	0.74
SOL L2: D Sectors [%]	0.52
SOL L1: Data Bank Writes [%]	0.12
SOL L2: D Sectors Fill Device [%]	0.10
SOL L1: F Wavefronts [%]	0.00
SOL L1: Texin Sm2tex Req Cycles Active [%]	0.00
SOL L2: D Atomic Input Cycles Active [%]	0
SOL L2: D Sectors Fill Sysmem [%]	0
SOL L1: Data Pipe Tex Wavefronts [%]	0
SOL L1: Tex Writeback Active [%]	0