ECE 408 Project Report

Team: Shell
School: UIUC
Zhenbang Wang, zw11, 5d97b22088a5ec28f9cb9501
Qiankun Peng, qinkunp2, 5d97b1fa88a5ec28f9cb94ba
Xiaoyi Shen, xiaoyis2, 5d97b20b88a5ec28f9cb94da

Milestone 3

Correctness and timing with 3 different dataset sizes

✱ Running /usr/bin/time python m3.1.py 100

Loading fashion-mnist data... done

Loading model... done

New Inference

Op Time: 0.007650

Op Time: 0.034392

Correctness: 0.76 Model: ece408

5.09user 3.11system 0:04.44elapsed 184%CPU

✱ Running /usr/bin/time python m3.1.py 1000

Loading fashion-mnist data... done

Loading model... done

New Inference

Op Time: 0.075432

Op Time: 0.317837

Correctness: 0.767 Model: ece408

5.16user 3.23system 0:04.53elapsed 185%CPU

✱ Running /usr/bin/time python m3.1.py 10000

Loading fashion-mnist data... done

Loading model... done

New Inference

Op Time: 0.683182

Op Time: 3.055441

Correctness: 0.7653 Model: ece408

7.83user 4.16system 0:08.15elapsed 147%CPU

Report: demonstrate `nvprof` profiling the execution

✱ Running nvprof python m3.1.py 10000

Loading fashion-mnist data... done

==720== NVPROF is profiling process 720, command: python m3.1.py 10000

Loading model... done

New Inference

Op Time: 0.673526

Op Time: 3.055023

Correctness: 0.7653 Model: ece408

==720== Profiling application: python m3.1.py 10000

==720== Profiling result:

Type Time(%) Time Calls Avg Min Max Name

GPU activities:

98.13% 3.72847s 2 1.86424s 673.48ms 3.05499s mxnet::op::forward_kernel

0.90% 34.233ms 20 1.7116ms 1.0240us 32.016ms [CUDA memcpy HtoD]

0.44% 16.793ms 2 8.3965ms 3.0610ms 13.732ms void hadow::cuda::MapPlanLargeKernel

0.21% 7.8283ms 1 7.8283ms 7.8283ms 7.8283ms volta_sgemm_128x128_tn

0.19% 7.3310ms 2 3.6655ms 25.248us 7.3057ms void op_generic_tensor_kernel

0.12% 4.4316ms 1 4.4316ms 4.4316ms 4.4316ms void cudnn::detail::pooling_fw_4d_kernel

.....

API calls:

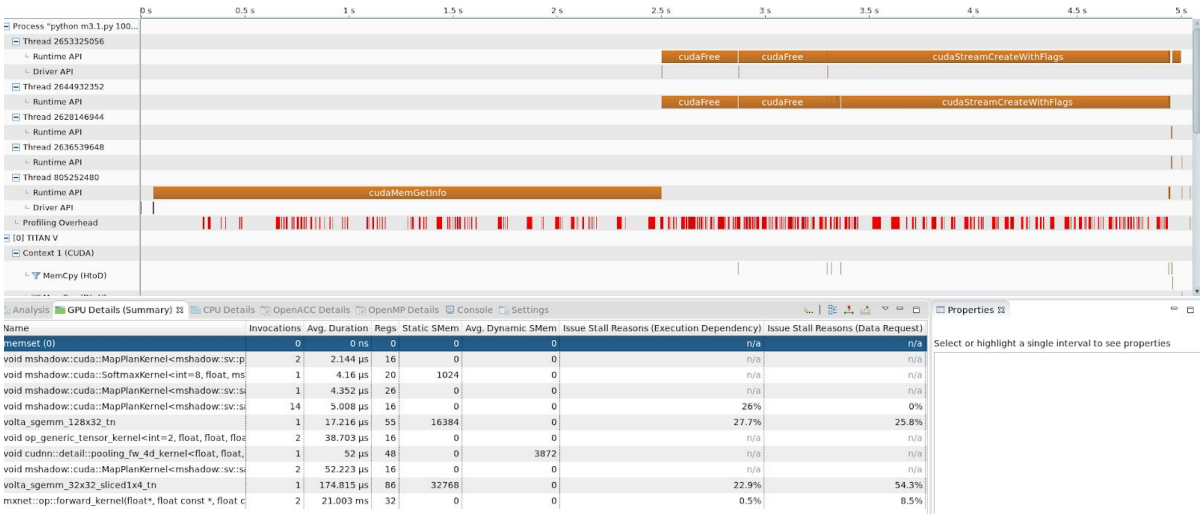
33.82% 3.74530s 6 624.22ms 7.0110us 3.05500s cudaDeviceSynchronize

28.30% 3.13395s 22 142.45ms 14.961us 1.63567s cudaStreamCreateWithFlags

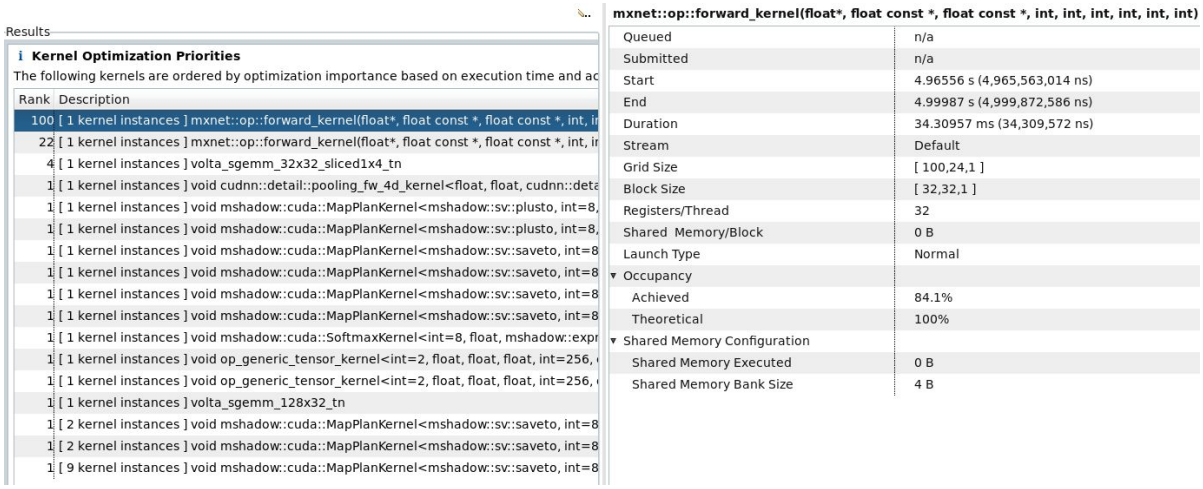
21.26% 2.35497s 22 107.04ms 69.205us 2.35030s cudaMemGetInfo

14.31% 1.58523s 18 88.068ms 1.2000us 422.74ms cudaFree

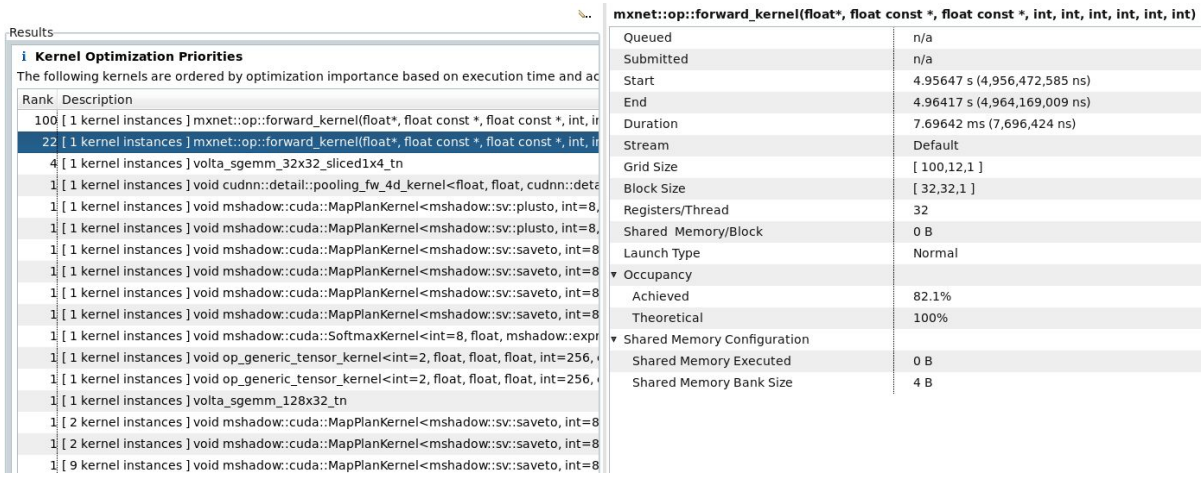
.....



forward1_anlalysis.nvprof:



forward2_anlalysis.nvprof:



Milestone 2

Report: Include a list of all kernels that collectively consume more than 90% of the program time.

33.29% 35.488ms 20 1.7744ms 1.0560us 33.053ms [CUDA memcpy HtoD]

16.83% 17.945ms 1 17.945ms 17.945ms 17.945ms volta_scudnn_128x64_relu_interior_nn_v1

16.20% 17.272ms 4 4.3179ms 4.3157ms 4.3225ms volta_gcgemm_64x32_nt

8.88% 9.4669ms 4 2.3667ms 1.9527ms 3.1200ms void fft2d_c2r_32x32<float, bool=0, bool=0, unsigned int=0, bool=0, bool=0>(float*, float2 const *, int, int, int, int, int, int, int, int, float, float, cudnn::reduced_divisor, bool, float*, float*, int2, int, int)

7.34% 7.8283ms 1 7.8283ms 7.8283ms 7.8283ms volta_sgemm_128x128_tn

6.82% 7.2746ms 2 3.6373ms 25.696us 7.2489ms void op_generic_tensor_kernel<int=2, float, float, float, int=256, cudnnGenericOp_t=7, cudnnNanPropagation_t=0, cudnnDimOrder_t=0, int=1>(cudnnTensorStruct, float*, cudnnTensorStruct, float const *, cudnnTensorStruct, float const *, float, float, float, float, dimArray, reducedDivisorArray)

5.96% 6.3584ms 4 1.5896ms 1.2508ms 2.0463ms void fft2d_r2c_32x32<float, bool=0, unsigned int=0, bool=0>(float2*, float const *, int, int, int, int, int, int, int, int, int, cudnn::reduced_divisor, bool, int2, int, int)

Report: Include a list of all CUDA API calls that collectively consume more than 90% of the program time.

44.73% 3.27839s 22 149.02ms 14.917us 1.63840s cudaStreamCreateWithFlags

31.24% 2.28938s 24 95.391ms 57.163us 2.28312s cudaMemGetInfo

21.70% 1.59038s 19 83.704ms 1.2520us 427.52ms cudaFree

Report: Include an explanation of the difference between kernels and API calls

Kernels are the major computational component, taking the responsibilities like data transfer between host and devices and launching GPU computation. Kernels are executed asynchronously.

While API calls interact with the CUDA driver and runtime libraries. API calls can be synchronous or asynchronous.

Report: Show output of rai running MXNet on the CPU

Loading fashion-mnist data... done

Loading model... done

New Inference

EvalMetric: {'accuracy': 0.8154}

Report: List program run time

20.22user 6.92system 0:10.19elapsed 266%CPU

Report: Show output of rai running MXNet on the GPU

Loading fashion-mnist data... done

Loading model... done

New Inference

EvalMetric: {'accuracy': 0.8154}

Report: List program run time

5.14user 3.19system 0:04.82elapsed 172%CPU

M2.1 CPU implementation

Report: List whole program execution time

Dataset size 100:

4.89user 2.50system 0:01.90elapsed 388%CPU

Dataset size 1000:

12.75user 3.12system 0:09.16elapsed 173%CPU

Dataset size 10000:

87.80user 9.97system 1:16.46elapsed 127%CPU

Report: List Op Times

Dataset size 100:

Op Time: 0.115347

Op Time: 0.613382

Dataset size 1000:

Op Time: 1.108179

Op Time: 6.560096

Dataset size 10000:

Op Time: 11.660823

Op Time: 60.506440