Names: Mike Pauls, Jayati Singh, Austin Lee
NetIds: mepauls2, jayati, sal3
Team name: team_name
School: On-campus

CHECKPOINT 2

Kernels (>90% of program time):*Here we assume GPU activity = memory transfer + kernel operations (gpu hardware usage)*

32.06% 35.969ms 20 1.7984ms 1.1200us 33.609ms **[CUDA memcpy HtoD]**

17.88% 20.062ms 1 20.062ms 20.062ms 20.062ms **volta_scudnn_128x64_relu_interior_nn_v1**

17.16% 19.252ms 4 4.8129ms 4.8122ms 4.8133ms **volta_gcgemm_64x32_nt**

8.53% 9.5671ms 4 2.3918ms 2.0052ms 3.1255ms **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, int, float, float, cudnn::reduced_divisor, bool, float*, float*, int2, int, int)**

7.80% 8.7556ms 1 8.7556ms 8.7556ms 8.7556ms **volta_sgemm_128x128_tn**

6.42% 7.2052ms 2 3.6026ms 25.536us 7.1797ms **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)**

5.70% 6.3895ms 4 1.5974ms 1.2742ms 2.0207ms **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)**

3.88% 4.3527ms 1 4.3527ms 4.3527ms 4.3527ms **void cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float, cudnnNanPropagation_t=0>, int=0, bool=0>(cudnnTensorStruct, float const *, cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float, cudnnNanPropagation_t=0>, int=0, bool=0>, cudnnTensorStruct*, cudnnPoolingStruct, float, cudnnPoolingStruct, int, cudnn::reduced_divisor, float)**

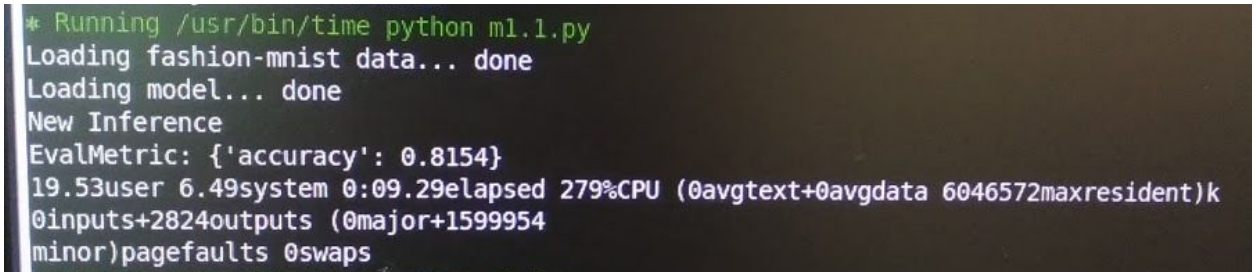
CUDA API Calls (>90% program time):

1. cudaStreamCreateWithFlags
   41.41%(time%) 3.08766s(time) 22(calls) 140.35ms(avg) 14.396us(min) 1.61488s(max)

2. cudaMemGetInfo
   33.15%(time%) 2.47141s(time) 24(calls) 102.98ms(avg) 55.402us(min) 2.46633s(max)
3. cudaFree
   21.17%(time%) 1.57836s(time) 19(calls) 83.072ms(avg) 1.2440us(min) 421.47ms(max)

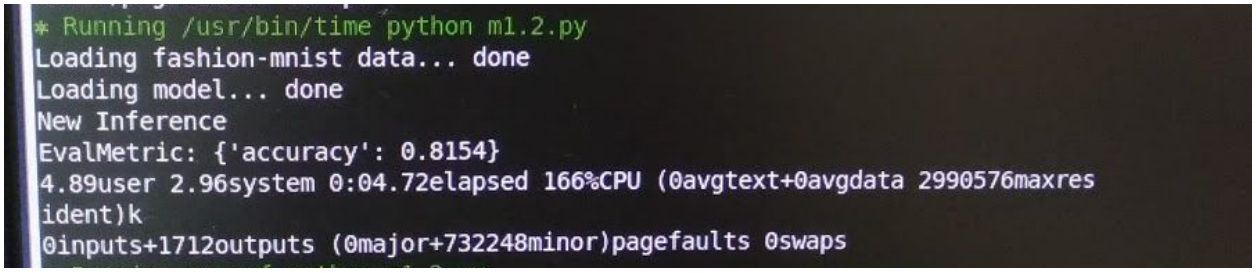
Kernel launch vs. API call:
CUDA API calls are instructions (cudaMemcpy, cudaGetDevice, etc.) that are executed by the host (CPU) to initiate memory transfer, execution or to communicate with the GPU and are executed once.
GPU kernels are C functions that when called, are executed N times in parallel by N different CUDA threads, as opposed to only once like regular C functions. (N is defined by the grid).

Rai running MXNet on the CPU:



Program run time:
User: 19.53 seconds
System: 6.49 seconds
Elapsed: 0:09.29

Rai running MXNet on the GPU:



Program run time:
User: 4.89 seconds
System: 2.96 seconds

Elapsed: 0:04.72

Whole program execution time:
New Inference:10000
User: 88.36 seconds
System: 10.38 seconds
Elapsed: 1:16.79 seconds
Op Time: 11.134082 seconds
Op Time: 61.390580 seconds
Correctness: 0.7653 Model: ece408

New Inference:1000
User: 18.35 seconds
System: 2.70 seconds
Elapsed: 0:11.22
Op Time: 1.317549
Op Time: 6.760934
Correctness: 0.767 Model: ece408

New Inference:100
User: 8.61
System: 2.60
Elapsed: 0:03.16
Op Time: 0.119225
Op Time: 0.676391
Correctness: 0.76 Model: ece408

CHECKPOINT 3

New Inference:10000
User: 5.40
System: 2.81
Elapsed: 0:05.34
Op Time: 0.024416
Op Time: 0.082711
Correctness: 0.7653 Model: ece408

New Inference:1000
User: 4.60
System: 2.71
Elapsed: 0:04.31
Op Time: 0.002462

Op Time: 0.008827
Correctness: 0.767 Model: ece408

New Inference:100
User: 4.93
System: 2.82
Elapsed: 0:04.35
Op Time: 0.000275
Op Time: 0.000925
Correctness: 0.76 Model: ece408

Nvprof analysis: