Names: Abhay Varmaraja, Jonathan Nativ, Raghava Ravi
NetIDs: abhaymv2, jnativ2, raghava4
RAI ID: 5d97b21b88a5ec28f9cb94f8, 5d97b1f088a5ec28f9cb94a8, 5d97b20088a5ec28f9cb94c6
Team: junior_eligibility
School Affiliation: On campus

90% program time kernels:

[CUDA memcpy HtoD]
volta_scudnn_128x64_relu_interior_nn_v1
volta_gcgemm_64x32_nt
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)
volta_sgemm_128x128_tn
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)
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)
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)

90% time API calls:

cudaStreamCreateWithFlags
cudaMemGetInfo
cudaFree

Difference between kernels and API calls:

CUDA API calls are made by the host to interact with the device and device memory. These are calls such as cudaMemcpy, cudaMalloc, cudaFree, kernel invocation, etc.
CUDA kernels are the device code that is scheduled on the compute queue when the host invokes a kernel, or when the device invokes a kernel. This code is prefaced by __global__ or __device__.

CPU MxNet Output:

```
✱ Running /usr/bin/time python m1.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8154}
```

```
18.13user 5.48system 0:10.17elapsed 232%CPU (0avgtext+0avgdata 6046756maxresident)k
0inputs+2824outputs (0major+1597129minor)p
agefaults 0swaps
```

CPU MxNet Runtime:
user: 18.13
system: 5.48
elapsed: 0:10.17

GPU MxNet Output:

```
✱ Running /usr/bin/time python m1.2.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8154}
10.53user 2.06system 0:06.47elapsed 194%CPU (0avgtext+0avgdata 2986076maxresident)k
0inputs+1712outputs (0major+731329min
or)pagefaults 0swaps
```

GPU MxNet Runtime:
user: 10.53
system: 2.06
elapsed: 0:06.47

CPU Implementation Output:

```
✱ Running /usr/bin/time python m2.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 11.170585
Op Time: 62.900722
Correctness: 0.7653 Model: ece408
89.86user 10.13system 1:18.52elapsed 127%CPU (0avgtext+0avgdata 6045132maxresident)k
0inputs+2824outputs (0major+2310115minor)pagefaults 0sw
aps
```

CPU Implementation Execution Times:
user: 89.86
system: 10.13
elapsed: 1:18.52

CPU Implementation OP Times:

OP 1: 11.170585

OP 2: 62.900722