

ECE 408 Report

Team Name: tiledtitans

Members:

- Jackson DeDobbelaere (dedobbe2)
- Matthew Grossfeld (grossfe2)
- Xinbo Wu (xinbowu2)

All on campus students

MILESTONE 1

List of all kernels that collectively consume more than 90% of the program time:

- [CUDA memcpy HtoD]
- void cudnn::detail::implicit_convolve_sgemm<float, float, int=1024, int=5, int=5, int=3, int=3, int=3, int=1, bool=1, bool=0, bool=1>(int, int, int, float const *, int, float*, cudnn::detail::implicit_convolve_sgemm<float, float, int=1024, int=5, int=5, int=3, int=3, int=3, int=1, bool=1, bool=0, bool=1>*, kernel_conv_params, int, float, float, int, float, float, int, int)
- Volta_cgemm_64x32_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)
- Volta_sgemm_128x128_tn
- void fft2d_c2r_32x32<float, bool=0, bool=0, unsigned int=1, 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)
- 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)

List of all CUDA API calls that collectively consume more than 90% of the program time:

- cudaStreamCreateWithFlags
- cudaMemGetInfo
- cudaFree

Explanation of the difference between kernels and API calls:

API calls are done with the CPU and execute from the time the call is made to the time it is returned. Kernels are ran on the GPU and are measured by the total time that the kernel is executing and running instructions. cudaLaunchKernel is an API function called from the CPU

that can launch kernels from the GPU. The profiling done on the API accounts for all of the launch overhead while the kernel timing only contains a small portion of it.

Output of rai running MXNet on the CPU:

```
1. bash
om requests<2.19.0,>=2.18.4->mxnet==1.3.1) (2.6)
Requirement already satisfied: urllib3<1.23,>=1.21.1 in /root/.local/lib/python2.7/site-packa
ges (from requests<2.19.0,>=2.18.4->mxnet==1.3.1) (1.22)
Requirement already satisfied: certifi>=2017.4.17 in /root/.local/lib/python2.7/site-packages
 (from requests<2.19.0,>=2.18.4->mxnet==1.3.1) (2018.11.29)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /root/.local/lib/python2.7/site-packa
ges (from requests<2.19.0,>=2.18.4->mxnet==1.3.1) (3.0.4)
Installing collected packages: mxnet
  Running setup.py develop for mxnet
Successfully installed mxnet
* Running /usr/bin/time python m1.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8236}
9.14user 3.50system 0:05.28elapsed 239%CPU (0avgtext+0avgdata 2472816maxresident)k
0inputs+2824outputs (0
major+666007minor)pagefaults 0swaps
* The build folder has been uploaded to http://s3.amazonaws.com/files.rai-project.com/userdat
a/build-5c7c6526c63bea0ecd2adc99.tar.gz. The data will be present for only a short duration o
f time.
* Server has ended your request.
wirelessprv-10-194-40-73:CS 483 jacksonedobbelaere$
```

Program run time of MXNet on the CPU:

- User: 9.14
- System: 3.50
- Elapsed: 0:05.28

Output of rai running MXNet on the GPU:

```
1. bash
om requests<2.19.0,>=2.18.4->mxnet==1.3.1) (2.6)
Requirement already satisfied: urllib3<1.23,>=1.21.1 in /root/.local/lib/python2.7/site-packa
ges (from requests<2.19.0,>=2.18.4->mxnet==1.3.1) (1.22)
Requirement already satisfied: certifi>=2017.4.17 in /root/.local/lib/python2.7/site-packages
 (from requests<2.19.0,>=2.18.4->mxnet==1.3.1) (2018.11.29)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /root/.local/lib/python2.7/site-packa
ges (from requests<2.19.0,>=2.18.4->mxnet==1.3.1) (3.0.4)
Installing collected packages: mxnet
  Running setup.py develop for mxnet
Successfully installed mxnet
* Running /usr/bin/time python m1.2.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8236}
4.23user 3.38system 0:04.17elapsed 182%CPU (0avgtext+
0avgdata 2846888maxresident)k
0inputs+4552outputs (0major+662021minor)pagefaults 0swaps
* The build folder has been uploaded to http://s3.amazonaws.com/files.rai-project.com/userdat
a/build-5c7c65e4c63bea0eebf3e091.tar.gz. The data will be present for only a short duration o
f time.
* Server has ended your request.
wirelessprv-10-194-40-73:CS 483 jacksondedobbelaere$
```

Program run time of MXNet on the GPU:

- User: 4.23
- System: 3.38
- Elapsed: 0:04.17

MILESTONE 2

100 Images:

- **Whole Program Execution Time:**
 - User: 2.77
 - System: 2.65
 - Elapsed: 0:01.03
- **Layer 1 Op Time:** 0.034247
- **Layer 2 Op Time:** 0.074316

1,000 Images:

- **Whole Program Execution Time:**
 - User: 4.24
 - System: 3.15
 - Elapsed: 0:01.95
- **Layer 1 Op Time:** 0.238204
- **Layer 2 Op Time:** 0.743339

10,000 Images:

- **Whole Program Execution Time:**
 - User: 14.97
 - System: 4.47
 - Elapsed: 0:11.37
- **Layer 1 Op Time:** 2.427247
- **Layer 2 Op Time:** 7.383249

MILESTONE 3

100 Images:

- **Correctness:** 0.84
- **Whole Program Execution Time:**
 - User: 4.36
 - System: 3.27
 - Elapsed: 0:04.34
- **Layer 1 Op Time:** 0.000102
- **Layer 2 Op Time:** 0.000236

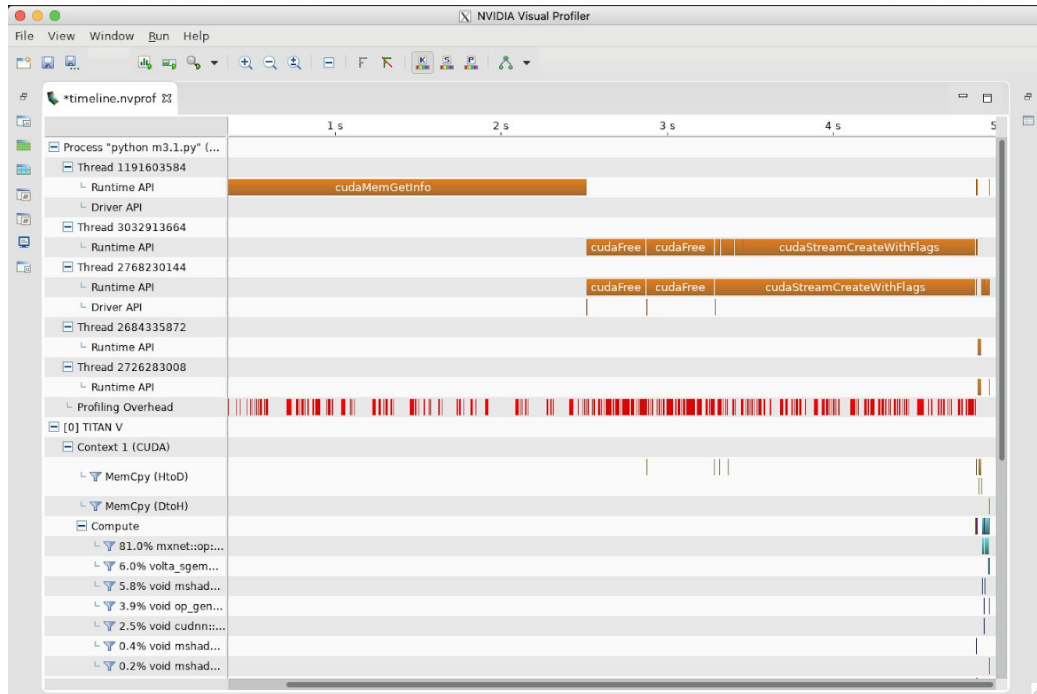
1,000 Images:

- **Correctness:** 0.852
- **Whole Program Execution Time:**
 - User: 4.42
 - System: 3.39
 - Elapsed: 0:04.22
- **Layer 1 Op Time:** 0.000908
- **Layer 2 Op Time:** 0.002467

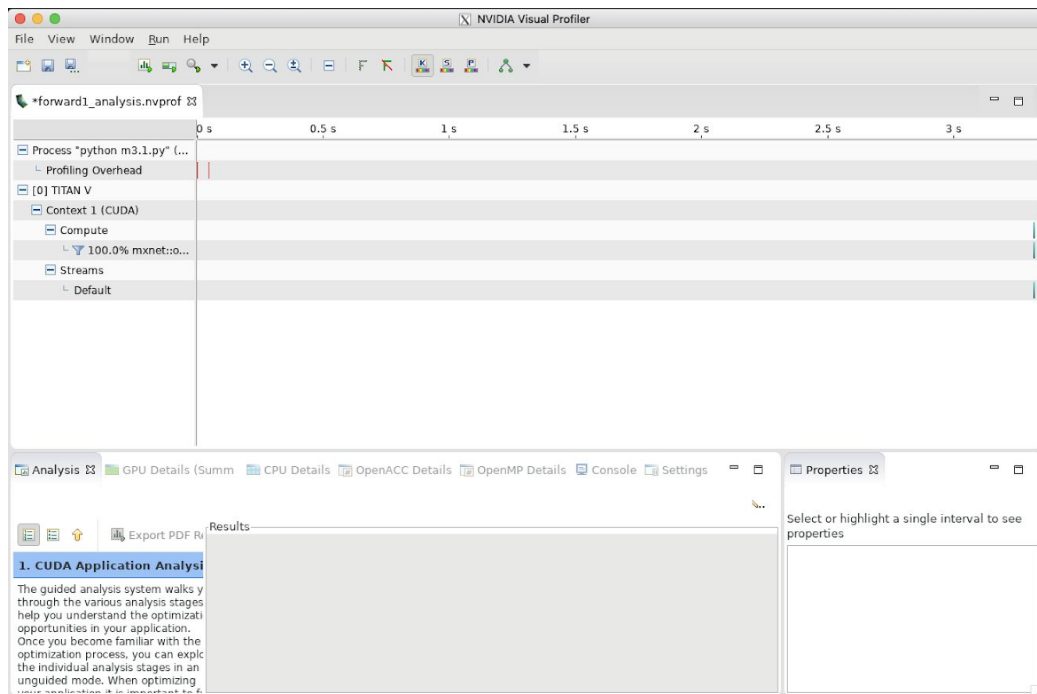
10,000 Images:

- **Correctness:** 0.8397
- **Whole Program Execution Time:**
 - User: 4.33
 - System: 3.31
 - Elapsed: 0:04.34
- **Layer 1 Op Time:** 0.009238
- **Layer 2 Op Time:** 0.024392

NVVP output of timeline.nvprof:



NVVP output of forward1_analysis.nvprof:



NVVP output of forward2_analysis.nvprof:

