Milestone 2 Report

Aditi Panwar (apanwa3), Neva Manali (manalil2), Vladimir Montchik (vam4)

Team Name: apvmnm School affliction: On-campus

October 2019

1 nvprof Profiling

Trimming the output a bit from the nvprof run, we provide:

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

Output:

```
...
* Running nvprof python m1.2.py
Loading fashion-mnist data... done
==264== NVPROF is profiling process 264, command: python m1.2.py
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8154}
==264== Profiling application: python m1.2.py
==264== Profiling result:
Type            Time(%)  Time      Calls       Avg       Min       Max  Name
GPU activities: 32.06%   35.522ms     20  1.7761ms  1.1200us  33.192ms  [CUDA memcpy HtoD]
                18.03%   19.978ms      1  19.978ms  19.978ms  19.978ms  volta_scudnn_128x64...
                17.26%   19.125ms      4  4.7812ms  4.7794ms  4.7843ms  volta_gcgemm_64x32_...
                 8.64%   9.5744ms      4  2.3936ms  1.9974ms  3.1196ms  void fft2d_c2r_32x3...
                 7.19%   7.9640ms      1  7.9640ms  7.9640ms  7.9640ms  volta_sgemm_128x128...
                 6.56%   7.2673ms      2  3.6336ms  25.184us  7.2421ms  void op_generic_ten...
                 5.78%   6.4042ms      4  1.6010ms  1.2587ms  2.0346ms  void fft2d_r2c_32x3...
                 3.93%   4.3538ms      1  4.3538ms  4.3538ms  4.3538ms  void cudnn::detail:...
...
API calls:      42.02%   3.11594s     22  141.63ms  13.772us  1.60424s  cudaStreamCreateWithF...
                33.18%   2.46098s     24  102.54ms  58.814us  2.44646s  cudaMemGetInfo
                21.21%   1.57319s     19  82.800ms  1.2310us  421.68ms  cudaFree
...
```

Without the trimming (but no template or parameter arguments), these are the following kernels that consume more than 90% of the time:

- `[CUDA memcpy HtoD]`
- `volta_scudnn_128x64_relu_interior_nn_v1`
- `volta_gcgemm_64x32_nt`
- `void fft2d_c2r_32x32<...>(...)`
- `volta_sgemm_128x128_tn`
- `void op_generic_tensor_kernel<...>(...)`
- `void fft2d_r2c_32x32<...>(...)`
- `void cudnn::detail::pooling_fw_4d_kernel<...>(...)`

Without the trimming (but no template or parameter arguments), these are the following CUDA API calls that consume more than 90% of the time:

- `cudaStreamCreateWithFlags`
- `cudaMemGetInfo`
- `cudaFree`

Now, to answer the question:

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

A kernel is just a function, with the __**global**__ keyword, that has been specified to run on the GPUs, while a CUDA API call is part of the library of code already written that allows programmers to write kernels or to call existing functionality, like cudaFree.

2 Running MXNet on the CPU

Trimming the output a bit, we provide:

- *Report: Show output of rai running MXNet on the CPU*
- *Report: List program run time*

Output:

```
* Running /usr/bin/time python m1.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8154}
17.03user 4.89system 0:08.94elapsed 245%CPU (0avgtext+0avgdata 6045960maxresident)k
0inputs+2824outputs (0major+1604073
minor)pagefaults 0swaps
```

Elapsed execution time: 8.94 seconds

3 Running MXNet on the GPU

Trimming the output a bit, we provide:

- *Report: Show output of rai running MXNet on the GPU*
- *Report: List program run time*

Output:

```
* Running /usr/bin/time python m1.2.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8154}
5.06user 3.26system 0:04.72elapsed 175%CPU (0avgtext+0avgdata 2999612maxresident)k
0inputs+4536outputs (0major+737148minor)pagefaults 0swaps
```

Elapsed execution time: 4.72 seconds

4 CPU Implementation

Trimming the output a bit, we provide:

- *Report: List whole program execution time*
- *Report: List Op Times*

Output:

```
* Running /usr/bin/time python m2.1.py
Loading fashion-mnist data... done
Loading model... done
New Inference
Op Time: 10.826889
Op Time: 59.171352
Correctness: 0.7653 Model: ece408
82.92user 8.53system 1:13.62elapsed 124%CPU (0avgtext+0avgdata 6045476maxresident)k
0inputs+0outputs (0major+2308136minor)pagefaults 0swaps
```

Total elapsed execution time: 1 minute and 13.62 seconds

Op times: 10.826889 seconds for the first operation, and 59.171352 seconds for the second operation.