Jiyu Hu
jiyuhu2
rai id: 5d97b1cc88a5ec28f9cb9464

Leihao Chen
leihaoc2
rai id: 5d97b1b088a5ec28f9cb9430

Anthony Nguyen
alnguyn2
rai id: 5d97b1f288a5ec28f9cb94ab

Milestone 3:

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

mxnet::op::forward_kernel(float*, float const *, float const *, int, int, int, int, int, int)

[CUDA memcpy HtoD]

void mshadow::cuda::MapPlanLargeKernel<mshadow::sv::saveto, int=8, int=1024, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=4, float>, float>, mshadow::expr::Plan<mshadow::expr::BinaryMapExp<mshadow::op::mul, mshadow::expr::ScalarExp<float>, mshadow::Tensor<mshadow::gpu, int=4, float>, float, int=1>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=4, 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 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)

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

cudaStreamCreateWithFlags

cudaMemGetInfo

cudaFree

3) difference between kernels and API calls:

Kernels are C functions defined by the user to execute N times in parallel by N CUDA threads. Therefore, a kernel launch is to execute the user defined C function.

API functions are the functions provided by CUDA to execute some operations on CUDA GPU. API function calls are when users adopt the provided CUDA functions.

4) output of rai running MXNet on the GPU:

*Running nvprof python m3.1py

New Inference

Op Time: 0.042851

Op Time: 0.359589

Correctness: 0.7653 Model: ece408

5) output of rai running MXNet on CPU (2.1):

*Running nvprof python m2.1py

New Inference

Op Time: 11.235300

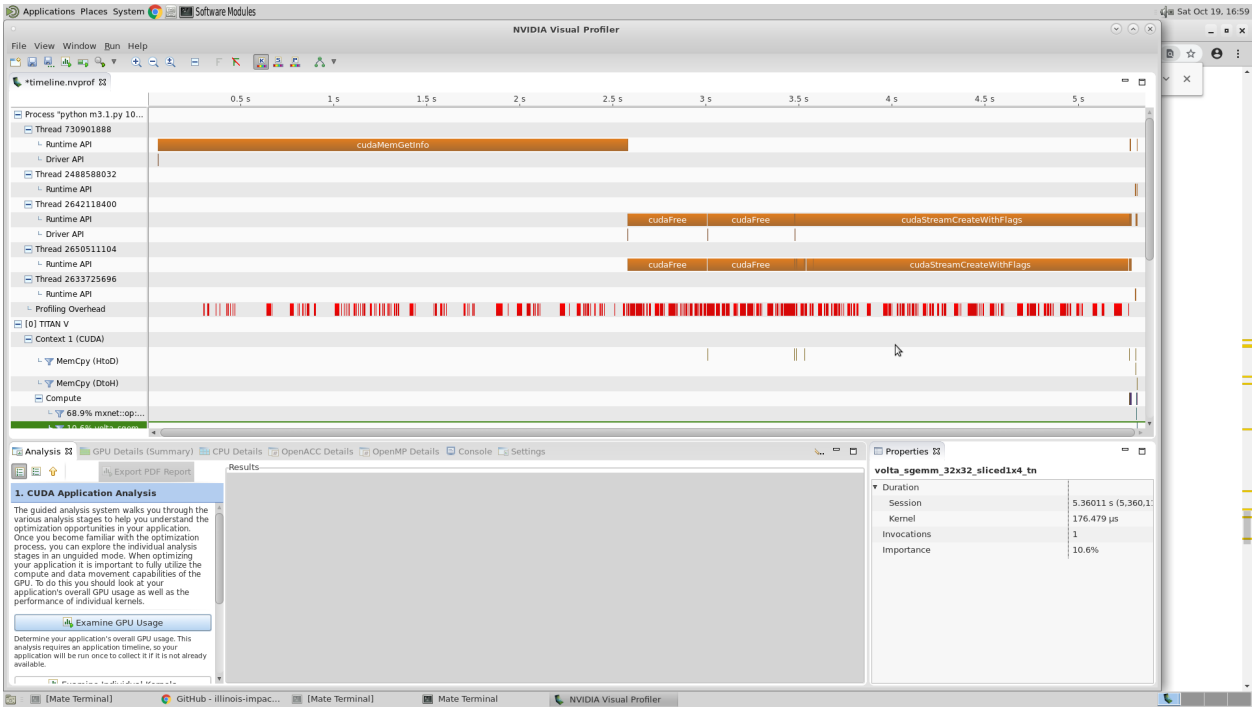
Op Time: 60.05617

Correctness: 0.7653 Model: ece408

6) Whole program execution time

5.31user 3.38system 0:04.85elapsed 179%CPU (0avgtext+0avgdata 2989092maxresident)k

7) NVVP results



Milestone 2:

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

```
[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)
```

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

```
cudaStreamCreateWithFlags

cudaMemGetInfo

cudaFree
```

3) difference between kernels and API calls:
   Kernels are C functions defined by the user to execute N times in parallel by N CUDA threads. Therefore, a kernel launch is to execute the user defined C function.
   API functions are the functions provided by CUDA to execute some operations on CUDA GPU. API function calls are when users adopt the provided CUDA functions.

4) output of rai running MXNet on the CPU:

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

   program runtime:

```
17.09user 4.83system 0:09.02elapsed 243%CPU (0avgtext+0avgdata 6044912maxresident)k
```

5) output of rai running MXNet on the GPU:

```
* Running /usr/bin/time python m1.2.py
Loading fashion-mnist data... done
Loading model... done
New Inference
EvalMetric: {'accuracy': 0.8154}
```

   program runtime:

```
5.05user 3.23system 0:04.76elapsed 174%CPU (0avgtext+0avgdata 2963832maxresident)k
```

CPU Implementation:

6) Whole program execution time:

```
105.64user 9.82system 1:34.79elapsed 121%CPU (0avgtext+0avgdata 6044512maxresident)k
```

7) Op times:

```
Op Time: 13.013842
Op Time: 77.417661
```