GPU implementation of convolutional neural networks
ECE 408 project

Hu, Qingtao qhu13
Pan, Yunzhe ypan19
Zhu, Qixin qzhu3

Introduction

Neural networks have been extremely popular in recent years.Particularly, convolutional neural network (CNN) is a popular and powerful model that does well in imaging processing. However, CNN poses great challenge in efficient CPU implementation due to the large sizes of image and great amount of parallelized computations of convolutions. This challenge enables the use of GPU to gain a considerable edge over CPU.
// in the future, we achieved xxxx.

Milestone 1

We used MxNet as the neural network framework. The neural networks are run in CUDA on GPU on EWS cloud. The environment and the framework are configured and tested to better understand the performance of the code.

1.1

A piece of hello world CPU code that does nothing is run to test the environment.

Run	Real	User	Sys	Accuracy
1	21.514s	0.352s	0.070s	0.8673
2	14.226s	0.381s	0.079s	0.8673

1.2
The same is done on GPU.

Run	Real	User	Sys	Accuracy
1	54.069s	0.365s	0.074s	0.8673
2	40.931s	0.367s	0.072s	0.8673

1.3
The top four most time-consuming procedures are listed in the table. They altogether occupy 99.37% of the total time. The full NVPROF profile is shown in the appendix.

Item	Time percentage	Time consumed
`cudnn::detail::implicit_convolve_sgemm`	37.09%	2.2517s
`sgemm_sm35_ldg_tn_128x8x256x16x32`	28.82%	50.459ms
`cudnn::detail::activation_fw_4d_kernel`	14.24%	39.214ms
`cudnn::detail::pooling_fw_4d_kernel`	10.66%	19.381ms

Item	Time percentage	Time consumed
`cudaStreamCreateWithFlags`	46.02%	2.25170s
`cudaFree`	29.94%	1.46489s
`cudaMemGetInfo`	20.77%	1.01647s
`cudaStreamSynchronize`	2.64%	129.12ms

Since there was no forward convolution code being run, it is within expectation that the most time consuming part was the memory IO and GPU logistics rather than the real computation.

Milestone 2

A straight-forward for loop CPU CNN is implemented and tested for performance benchmark.

High

Run	Real	User	Sys	Op time	Accuracy
1	51.524s	0.326s	0.086s	11.700103s	0.8562
2	52.846s	0.351s	0.069s	11.776913s	0.8562

Low

Run	Real	User	Sys	Op time	Accuracy
1	55.168s	0.356s	0.075s	11.671123s	0.629
2	81.691s	0.371s	0.078s	22.084566s	0.629

Collaboration

We set up a meeting time and went over the pseudo-codes in the assignment documentation, discussed about the meaning of each line collaboratively. We then wrote codes by ourselves and compared the results with the one in the documentation individually. This way, we all benefited by sharing ideas and getting hands-on experience at the same time. Overall, we distributed labor evenly.

Milestone 3

We implemented the baseline version of forward convolution without optimization: all global memory accesses, lack of use of tiling, nested for loop in kernel. The results are as follow:

High

Run	Real	User	Sys	Op time	Accuracy
1	51.127s	0.501s	0.138s	1.239578s	0.8562

Kernel time

Item	Time percentage	Time consumed

mxnet::op::forward_kernel	93.31%	1.20757s
sgemm_sm35_ldg_tn_128x8x256x16x32	3.00%	38.774ms
cudnn::detail::activation_fw_4d_kernel	1.50%	19.385ms

API time

Item	Time percentage	Time consumed
cudaStreamSynchronize	36.39%	1.98774s
cudaStreamCreateWithFlags	23.70%	1.29477s
cudaFree	22.11%	1.20760s
cudaDeviceSynchronize	15.97%	872.51ms
cudaMemGetInfo	1.43%	8.2483ms

Low

Run	Real	User	Sys	Op time	Accuracy
1	47.419s	0.550s	0.156s	1.239578s	0.629

Kernel time

Item	Time percentage	Time consumed
mxnet::op::forward_kernel	93.33%	1.20728s
sgemm_sm35_ldg_tn_128x8x256x16x32	2.99%	38.723ms
cudnn::detail::activation_fw_4d_kernel	1.50%	19.381ms

Item	Time percentage	Time consumed
`cudaStreamSynchronize`	32.19%	2.49278s
`cudaStreamCreateWithFlags`	25.06%	1.94070s
`cudaFree`	15.59%	1.20762s
`cudaDeviceSynchronize`	15.59%	1.20732s
`cudaMemGetInfo`	11.22%	869.18ms

Not surprisingly, the forward kernel time is the most time-consuming part of the implementation. Because some level of parallelization is exploited, it is much faster than the given baseline implementation.

Collaboration

The project is done in a highly coherent and integrated way. There was no pronounced division of work. We studied the algorithm together. We wrote and debugged the code together. We contributed equally to this part.

Final Submission

To be finished.

Conclusion

To be finished.

Future work

To be finished

Reference

To be finished

Improvements in the course

To be finished

Appendix

NVPROF profile output

```
==308== Profiling application: python /src/m1.2.py
==308== Profiling result:
Time(%)      Time     Calls       Avg       Min       Max  Name
 37.09%  50.459ms         1  50.459ms  50.459ms  50.459ms  void
cudnn::detail::implicit_convolve_sgemm<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,
cudnn::detail::implicit_convolve_sgemm<float, int=1024, int=5, int=5, int=3, int=3, int=3, int=1,
bool=1, bool=0, bool=1>*, float const *, kernel_conv_params, int, float, float, int, float const
*, float const *, int, int)
 28.82%  39.214ms         1  39.214ms  39.214ms  39.214ms  sgemm_sm35_ldg_tn_128x8x256x16x32
14.24%  19.381ms         2  9.6906ms  460.86us  18.920ms  void
cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4,
cudnn::detail::tanh_func<float>>(cudnnTensorStruct, float const *,
cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4,
cudnn::detail::tanh_func<float>>, cudnnTensorStruct*, float, cudnnTensorStruct*, int,
cudnnTensorStruct*)
 10.66%  14.498ms         1  14.498ms  14.498ms  14.498ms  void
cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float,
cudnnNanPropagation_t=0>, int=0>(cudnnTensorStruct, float const *,
cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float,
cudnnNanPropagation_t=0>, int=0>, cudnnTensorStruct*, cudnnPoolingStruct, float,
cudnnPoolingStruct, int, cudnn::reduced_divisor, float)
  4.50%  6.1212ms        13  470.86us  1.5040us  4.2044ms  [CUDA memcpy HtoD]
  2.68%  3.6496ms         1  3.6496ms  3.6496ms  3.6496ms  sgemm_sm35_ldg_tn_64x16x128x8x32
0.82%  1.1208ms         1  1.1208ms  1.1208ms  1.1208ms  void mshadow::cuda::SoftmaxKernel<int=8,
float, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>>(mshadow::gpu, int=2,
unsigned int)
  0.55%  755.09us        12  62.924us  2.1120us  380.83us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::ScalarExp<float>, float>>(mshadow::gpu, unsigned int,
mshadow::Shape<int=2>, int=2)
0.32%  436.47us         2  218.24us  16.736us  419.74us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::plusto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::Broadcast1DExp<mshadow::Tensor<mshadow::gpu, int=1, float>,
float, int=2, int=1>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=2)
  0.29%  392.44us         1  392.44us  392.44us  392.44us  sgemm_sm35_ldg_tn_32x16x64x8x16
0.02%  23.647us         1  23.647us  23.647us  23.647us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::ReduceWithAxisExp<mshadow::red::maximum,
mshadow::Tensor<mshadow::gpu, int=3, float>, float, int=3, bool=1, int=2>, float>>(mshadow::gpu,
unsigned int, mshadow::Shape<int=2>, int=2)
  0.01%  9.5040us         1  9.5040us  9.5040us  9.5040us  [CUDA memcpy DtoH]
==308== API calls:
Time(%)      Time     Calls       Avg       Min       Max  Name
 46.02%  2.25170s        18  125.09ms  22.983us  1.12539s  cudaStreamCreateWithFlags
 29.94%  1.46489s        10  146.49ms     844ns  421.32ms  cudaFree
 20.77%  1.01647s        24  42.353ms  265.37us  1.00902s  cudaMemGetInfo
  2.64%  129.12ms        25  5.1646ms  5.6030us  83.772ms  cudaStreamSynchronize
  0.24%  11.891ms         8  1.4864ms  14.727us  5.6310ms  cudaMemcpy2DAsync
  0.19%  9.1674ms        42  218.27us  9.4430us  1.7049ms  cudaMalloc
  0.09%  4.4705ms         4  1.1176ms  28.466us  4.3187ms  cudaStreamCreate
  0.03%  1.5692ms         4  392.29us  338.48us  437.54us  cuDeviceTotalMem
  0.02%  950.37us       352  2.6990us     247ns  75.594us  cuDeviceGetAttribute
  0.02%  918.04us       114  8.0530us     942ns  327.22us  cudaEventCreateWithFlags
  0.01%  651.03us        23  28.305us  10.582us  106.39us  cudaLaunch
  0.01%  421.95us         6  70.325us  20.972us  135.35us  cudaMemcpy
  0.00%  131.68us         4  32.921us  22.761us  49.633us  cuDeviceGetName
  0.00%  100.67us         2  50.335us  25.446us  75.224us  cudaStreamCreateWithPriority
  0.00%  94.720us        32  2.9600us  1.0310us  9.2280us  cudaSetDevice
  0.00%  91.836us       110     834ns     553ns  2.9280us  cudaDeviceGetAttribute
  0.00%  78.254us       147     532ns     274ns  1.2230us  cudaSetupArgument
  0.00%  32.574us        23  1.4160us     529ns  3.6950us  cudaConfigureCall
  0.00%  18.628us        10  1.8620us     995ns  2.6750us  cudaGetDevice
  0.00%  10.535us         1  10.535us  10.535us  10.535us  cudaBindTexture
  0.00%  9.3610us        16     585ns     356ns     777ns  cudaPeekAtLastError
  0.00%  7.3230us         1  7.3230us  7.3230us  7.3230us  cudaStreamGetPriority
  0.00%  5.5710us         2  2.7850us  2.0610us  3.5100us  cudaStreamWaitEvent
  0.00%  5.5620us         6     927ns     420ns  1.7820us  cuDeviceGetCount
  0.00%  4.9790us         6     829ns     525ns  1.2770us  cuDeviceGet
  0.00%  4.9150us         2  2.4570us  1.5770us  3.3380us  cudaEventRecord
  0.00%  4.0900us         2  2.0450us  1.5300us  2.5600us  cudaDeviceGetStreamPriorityRange
  0.00%  3.9460us         6     657ns     430ns     886ns  cudaGetLastError
  0.00%  2.9840us         3     994ns     869ns  1.1170us  cuInit
  0.00%  2.3760us         1  2.3760us  2.3760us  2.3760us  cudaUnbindTexture
  0.00%  2.1760us         3     725ns     666ns     840ns  cuDriverGetVersion
  0.00%  1.4040us         1  1.4040us  1.4040us  1.4040us  cudaGetDeviceCount
```

∗ The build folder has been uploaded to http://s3.amazonaws.com/files.rai-project.com/userdata/build-434ad40e-e368-463a-b303-a9b813afe7a6.tar.gz. The data will be present for only a short duration of time.

∗ Server has ended your request.

```
real 1m41.534s
user 0m0.441s
sys 0m0.260s
```

3.1 high profile

```
==310== NVPROF is profiling process 310, command: python m3.1.py ece408-high 10000
Loading model... done
Op Time: 1.207671
Correctness: 0.8562 Model: ece408-high
==310== Profiling application: python m3.1.py ece408-high 10000
==310== Profiling result:
Time(%)      Time     Calls       Avg       Min       Max  Name
93.31%  1.20757s         1  1.20757s  1.20757s  1.20757s  void
mxnet::op::forward_kernel<mshadow::gpu, float>(float*, mxnet::op::forward_kernel<mshadow::gpu,
float> const *, mxnet::op::forward_kernel<mshadow::gpu, float> const , int, int, int, int, int,
int)
  3.00%  38.774ms         1  38.774ms  38.774ms  38.774ms  sgemm_sm35_ldg_tn_128x8x256x16x32
  1.50%  19.385ms         2  9.6924ms  458.17us  18.927ms  void
cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4,
cudnn::detail::tanh_func<float>>(cudnnTensorStruct, float const *,
cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4,
cudnn::detail::tanh_func<float>>, cudnnTensorStruct*, float, cudnnTensorStruct*, int,
cudnnTensorStruct*)
  1.12%  14.457ms         1  14.457ms  14.457ms  14.457ms  void
cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float,
cudnnNanPropagation_t=0>, int=0>(cudnnTensorStruct, float const *,
cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float,
cudnnNanPropagation_t=0>, int=0>, cudnnTensorStruct*, cudnnPoolingStruct, float,
cudnnPoolingStruct, int, cudnn::reduced_divisor, float)
0.58%  7.5608ms        13  581.60us  1.5670us  5.3588ms  [CUDA memcpy HtoD]
  0.28%  3.6150ms         1  3.6150ms  3.6150ms  3.6150ms  sgemm_sm35_ldg_tn_64x16x128x8x32
  0.09%  1.1139ms         1  1.1139ms  1.1139ms  1.1139ms  void
mshadow::cuda::SoftmaxKernel<int=8, float, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu,
int=2, float>, float>, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>,
float>>(mshadow::gpu, int=2, unsigned int)
0.06%  748.21us        12  62.350us  2.1120us  377.50us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::ScalarExp<float>, float>>(mshadow::gpu, unsigned int,
mshadow::Shape<int=2>, int=2)
  0.03%  433.18us         2  216.59us  16.671us  416.51us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::plusto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::Broadcast1DExp<mshadow::Tensor<mshadow::gpu, int=1, float>,
float, int=2, int=1>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=2)
  0.03%  389.50us         1  389.50us  389.50us  389.50us  sgemm_sm35_ldg_tn_32x16x64x8x16
0.00%  23.487us         1  23.487us  23.487us  23.487us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::ReduceWithAxisExp<mshadow::red::maximum,
mshadow::Tensor<mshadow::gpu, int=3, float>, float, int=3, bool=1, int=2>, float>>(mshadow::gpu,
unsigned int, mshadow::Shape<int=2>, int=2)
  0.00%  9.9840us         1  9.9840us  9.9840us  9.9840us  [CUDA memcpy DtoH]
==310== API calls:
Time(%)      Time     Calls       Avg       Min       Max  Name
 36.39%  1.98774s        18  110.43ms  17.470us  993.51ms  cudaStreamCreateWithFlags
 23.70%  1.29477s        10  129.48ms     786ns  382.11ms  cudaFree
 22.11%  1.20760s         1  1.20760s  1.20760s  1.20760s  cudaDeviceSynchronize
 15.97%  872.51ms        23  37.935ms  235.20us  865.73ms  cudaMemGetInfo
  1.43%  78.344ms        25  3.1338ms  5.2430us  42.214ms  cudaStreamSynchronize
0.15%  8.2483ms         8  1.0310ms  7.7770us  5.4923ms  cudaMemcpy2DAsync
  0.12%  6.6010ms        41  161.00us  12.485us  1.1344ms  cudaMalloc
  0.04%  2.1427ms         4  535.68us  41.893us  2.0121ms  cudaStreamCreate
  0.03%  1.3769ms         4  344.22us  339.21us  355.91us  cuDeviceTotalMem
  0.02%  875.87us       352  2.4880us     247ns  66.000us  cuDeviceGetAttribute
  0.01%  719.81us       114  6.3140us     618ns  300.88us  cudaEventCreateWithFlags
  0.01%  532.88us        23  23.168us  10.936us  63.957us  cudaLaunch
  0.01%  375.20us         6  62.533us  23.245us  130.61us  cudaMemcpy
  0.00%  108.96us         4  27.241us  16.139us  31.597us  cuDeviceGetName
  0.00%  98.868us        30  3.2950us     673ns  26.995us  cudaSetDevice
  0.00%  70.682us       104     679ns     417ns  2.2810us  cudaDeviceGetAttribute
  0.00%  62.412us       140     445ns     259ns  1.7900us  cudaSetupArgument
  0.00%  38.338us         2  19.169us  18.373us  19.965us  cudaStreamCreateWithPriority
  0.00%  30.662us        23  1.3330us     524ns  4.1380us  cudaConfigureCall
  0.00%  27.274us        10  2.7270us  1.3570us  6.9340us  cudaGetDevice
  0.00%  8.8900us        16     555ns     375ns     997ns  cudaPeekAtLastError
  0.00%  5.0900us         6     848ns     258ns  2.0140us  cuDeviceGetCount
  0.00%  4.4670us         1  4.4670us  4.4670us  4.4670us  cudaStreamGetPriority
  0.00%  4.3890us         6     731ns     421ns  1.2330us  cuDeviceGet
  0.00%  4.1870us         2  2.0930us  1.3870us  2.8000us  cudaStreamWaitEvent
  0.00%  3.8380us         2  1.9190us  1.2320us  2.6060us  cudaEventRecord
  0.00%  3.2460us         3  1.0820us     973ns  1.2210us  cuInit
  0.00%  3.1940us         2  1.5970us  1.3500us  1.8440us  cudaDeviceGetStreamPriorityRange
  0.00%  2.5230us         5     504ns     274ns     680ns  cudaGetLastError
  0.00%  2.3880us         3     796ns     747ns     838ns  cuDriverGetVersion
  0.00%  1.3640us         1  1.3640us  1.3640us  1.3640us  cudaGetDeviceCount
```

3.1 high low

```
==314== NVPROF is profiling process 314, command: python m3.1.py ece408-low 10000
Loading model... done
Op Time: 1.207424
Correctness: 0.629 Model: ece408-low
==314== Profiling application: python m3.1.py ece408-low 10000
==314== Profiling result:
Time(%)      Time     Calls       Avg       Min       Max  Name
 93.33%  1.20728s         1  1.20728s  1.20728s  1.20728s  void
mxnet::op::forward_kernel<mshadow::gpu, float>(float*, mxnet::op::forward_kernel<mshadow::gpu,
float> const *, mxnet::op::forward_kernel<mshadow::gpu, float> const , int, int, int, int, int,
int)
  2.99%  38.723ms         1  38.723ms  38.723ms  38.723ms  sgemm_sm35_ldg_tn_128x8x256x16x32
1.50%  19.381ms         2  9.6903ms  459.07us  18.922ms  void
cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4,
cudnn::detail::tanh_func<float>>(cudnnTensorStruct, float const *,
cudnn::detail::activation_fw_4d_kernel<float, float, int=128, int=1, int=4,
cudnn::detail::tanh_func<float>>, cudnnTensorStruct*, float, cudnnTensorStruct*, int,
cudnnTensorStruct*)
  1.12%  14.452ms         1  14.452ms  14.452ms  14.452ms  void
cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float,
cudnnNanPropagation_t=0>, int=0>(cudnnTensorStruct, float const *,
cudnn::detail::pooling_fw_4d_kernel<float, float, cudnn::detail::maxpooling_func<float,
cudnnNanPropagation_t=0>, int=0>, cudnnTensorStruct*, cudnnPoolingStruct, float,
cudnnPoolingStruct, int, cudnn::reduced_divisor, float)
  0.56%  7.2941ms        13  561.09us  1.6000us  5.2045ms  [CUDA memcpy HtoD]
  0.28%  3.6543ms         1  3.6543ms  3.6543ms  3.6543ms  sgemm_sm35_ldg_tn_64x16x128x8x32
0.09%  1.1103ms         1  1.1103ms  1.1103ms  1.1103ms  void mshadow::cuda::SoftmaxKernel<int=8,
float, mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>>(mshadow::gpu, int=2,
unsigned int)
0.06%  748.34us        12  62.361us  2.1110us  377.91us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::ScalarExp<float>, float>>(mshadow::gpu, unsigned int,
mshadow::Shape<int=2>, int=2)
0.03%  434.94us         2  217.47us  17.503us  417.43us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::plusto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::Broadcast1DExp<mshadow::Tensor<mshadow::gpu, int=1, float>,
float, int=2, int=1>, float>>(mshadow::gpu, unsigned int, mshadow::Shape<int=2>, int=2)
  0.03%  396.89us         1  396.89us  396.89us  396.89us  sgemm_sm35_ldg_tn_32x16x64x8x16
  0.00%  23.904us         1  23.904us  23.904us  23.904us  void
mshadow::cuda::MapPlanKernel<mshadow::sv::saveto, int=8,
mshadow::expr::Plan<mshadow::Tensor<mshadow::gpu, int=2, float>, float>,
mshadow::expr::Plan<mshadow::expr::ReduceWithAxisExp<mshadow::red::maximum,
mshadow::Tensor<mshadow::gpu, int=3, float>, float, int=3, bool=1, int=2>, float>>(mshadow::gpu,
unsigned int, mshadow::Shape<int=2>, int=2)
  0.00%  9.7270us         1  9.7270us  9.7270us  9.7270us  [CUDA memcpy DtoH]
==314== API calls:
Time(%)      Time     Calls       Avg       Min       Max  Name
32.19%  2.49278s        25  99.711ms  5.0350us  1.20729s  cudaStreamSynchronize
 25.06%  1.94070s        18  107.82ms  18.003us  969.98ms  cudaStreamCreateWithFlags
 15.59%  1.20762s        10  120.76ms     681ns  340.52ms  cudaFree
 15.59%  1.20732s         1  1.20732s  1.20732s  1.20732s  cudaDeviceSynchronize
 11.22%  869.18ms        23  37.790ms  235.92us  862.45ms  cudaMemGetInfo
  0.19%  15.001ms         8  1.8751ms  13.689us  7.4208ms  cudaMemcpy2DAsync
  0.08%  6.3905ms        41  155.87us  10.738us  1.1389ms  cudaMalloc
  0.02%  1.3687ms         4  342.17us  338.67us  349.83us  cuDeviceTotalMem
  0.01%  975.15us       352  2.7700us     245ns  157.86us  cuDeviceGetAttribute
  0.01%  874.70us       114  7.6720us     626ns  303.27us  cudaEventCreateWithFlags
  0.01%  567.82us        23  24.688us  10.936us  80.927us  cudaLaunch
  0.01%  466.02us         6  77.670us  29.550us  124.40us  cudaMemcpy
  0.00%  180.81us         4  45.203us  32.082us  73.465us  cudaStreamCreate
  0.00%  117.42us         4  29.356us  25.724us  31.243us  cuDeviceGetName
  0.00%  77.354us       104     743ns     413ns  2.1170us  cudaDeviceGetAttribute
  0.00%  71.489us        30  2.3820us     824ns  7.5000us  cudaSetDevice
  0.00%  62.593us       140     447ns     254ns  1.4290us  cudaSetupArgument
  0.00%  37.149us         2  18.574us  18.436us  18.713us  cudaStreamCreateWithPriority
  0.00%  30.865us        23  1.3410us     549ns  4.2540us  cudaConfigureCall
  0.00%  27.957us        10  2.7950us  1.5520us  6.3230us  cudaGetDevice
  0.00%  9.2750us        16     579ns     363ns  1.0350us  cudaPeekAtLastError
  0.00%  5.2370us         6     872ns     285ns  1.8220us  cuDeviceGetCount
  0.00%  4.5490us         1  4.5490us  4.5490us  4.5490us  cudaStreamGetPriority
  0.00%  3.9800us         2  1.9900us  1.4780us  2.5020us  cudaStreamWaitEvent
  0.00%  3.8350us         2  1.9170us  1.2330us  2.6020us  cudaEventRecord
  0.00%  3.5520us         6     592ns     365ns     858ns  cuDeviceGet
0.00%  3.5050us         2  1.7520us  1.4650us  2.0400us  cudaDeviceGetStreamPriorityRange
  0.00%  3.0250us         3  1.0080us     883ns  1.2090us  cuInit
  0.00%  2.8460us         5     569ns     322ns     783ns  cudaGetLastError
  0.00%  2.5880us         3     862ns     701ns  1.0850us  cuDriverGetVersion
  0.00%  1.1560us         1  1.1560us  1.1560us  1.1560us  cudaGetDeviceCount
```