CS 483/ECE 408: Applied Parallel Programming
the_cuda_corps
Larry Poon (Chicago), Gerald Kozel (Chicago), & Waleed Ramahi (UIUC)

Milestone 1

[kernel calls that collectively consume more than 90% of the program time]

Time(%)	Time	Calls	Avg	Min	Max	Name
34.07%	118.49ms	9	13.166ms	13.149ms	13.180ms	fermiPlusCgemmLDS 128_batched
27.01%	93.927ms	1	93.927ms	93.927ms	93.927ms	cudnn::detail::implicit _convolve_sgemm
12.69%	44.128ms	9	4.9031ms	2.6906ms	6.2766ms	fft2d_c2r_32x32
8.2%	28.514ms	1	28.514ms	28.514ms	28.514ms	sgemm_sm35_ldg_tn _128x8x256x16x32
6.42%	22.331ms	14	1.5951ms	1.5360us	21.504ms	[CUDA memcpy HtoD]
4.07%	14.156ms	2	7.0781ms	252.38us	13.904ms	cudnn::detail::activati on_fw_4d_kernel

[CUDA API calls that collectively consume more than 90% of the program time.]

Time(%)	Time	Calls	Avg	Min	Max	Name
43.61%	1.93912s	18	107.73ms	15.637us	969.21ms	cudaStreamCreateWi thFlags
27.11%	1.20548s	10	120.55ms	1.2190us	342.20ms	cudaFree
20.62%	917.01ms	27	33.963ms	236.99us	908.89ms	cudaMemGetInfo

API calls and kernel calls are both parts of the CUDA programming interface. Kernels are functions defined by the __global__ specifier which we are used to using in our homeworks. API calls are executed on the host, but have access to the global memory of the device. Kernel calls are executed on the device itself. The runtime API consists of cudaMemcpyTosymbol, cudaMalloc, and other functions used to access global variables. Generally the main tradeoff between the two is that kernel launches are more complex to implement but provide more fine-grained control while the runtime API makes device code management easier and cleaner. There's no significant performance difference between API and kernel calls, based on the statistics in the tables above.

[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': 0.8444}
```

[m1.1.1.py program run time]

```
13.26user 11.98system 0:11.66elapsed 216%CPU (0avgtext+0avgdata
2821748maxresident)k
0inputs+2624outputs (0major+38226minor)pagefaults 0swaps
```

[Output of rai running MXNet on the GPU]

```
✱ Running /usr/bin/time python m1.2.py
Loading fashion-mnist data...
done
Loading model...
[23:46:06] src/operator/./../cudnn_algoreg-inl.h:112: Running performance tests to find the best
convolution algorithm, this can take a while... (setting env variable
MXNET_CUDNN_AUTOTUNE_DEFAULT to 0 to disable)
done
New Inference
EvalMetric: {'accuracy': 0.8444}
```

[m1.2.py program run time]

```
2.29user 1.06system 0:02.82elapsed 118%CPU (0avgtext+0avgdata 1138624maxresident)k
0inputs+3136outputs (0major+153677minor)pagefaults 0swaps
```