

Profiling Deep Learning

Denis Boyda

2021 ALCF Simulation, Data, and Learning Workshop, October 5-7

Section overview

Zhen Xie



TensorFlow profiler

Murali Emali



Nvidia Nsight profiler
Intel Vtune profiler

Denis Boyda



MPITrace profiler
Horovod timeline

Outline

- What is profiling
- Motivation to profile
- Some frequent problems
- Common Profiler APIs

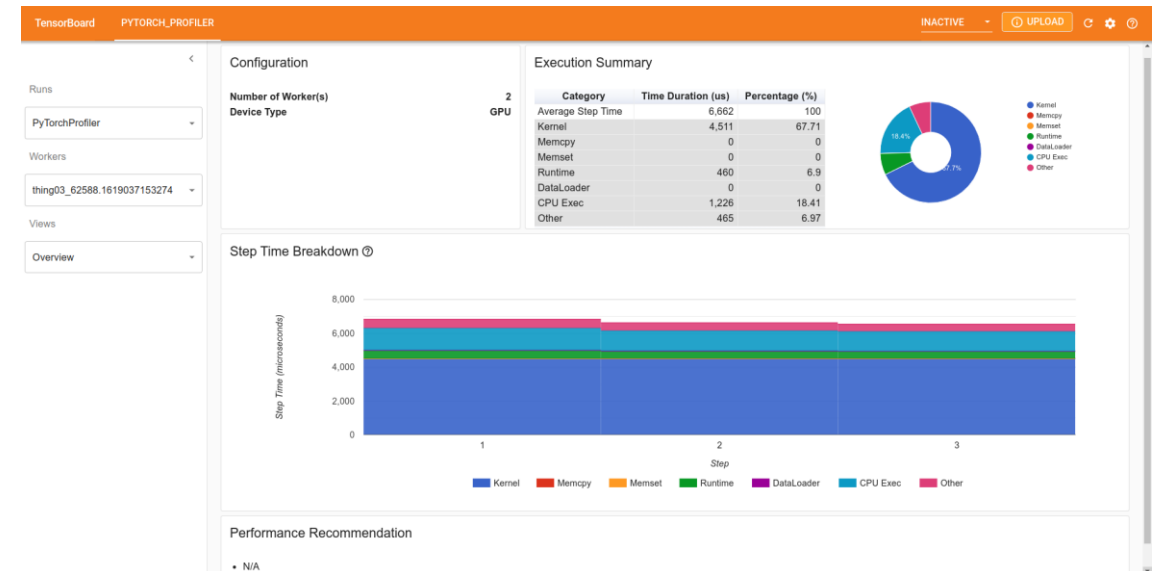
What is profiling?

- Measuring time/memory cost
- Identifying bottleneck
- Tracking call stack
- Measuring communications

```
181 function calls (169 primitive calls) in 3.434 seconds

Ordered by: cumulative time

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
10/1    0.000    0.000    3.434    3.434    module.py:866(_call_impl)
1       0.078    0.078    3.434    3.434    v2.py:21(forward)
1       0.000    0.000    2.817    2.817    <_array_function__ internals>:2(argwhere)
4/1     0.000    0.000    2.817    2.817    {built-in method numpy.core._multiarray_umath.implement_array_function}
1       0.005    0.005    2.817    2.817    numeric.py:537(argwhere)
2       0.000    0.000    2.812    1.406    fromnumeric.py:52(_wrapfunc)
1       0.000    0.000    1.476    1.476    <_array_function__ internals>:2(nonzero)
1       0.000    0.000    1.476    1.476    fromnumeric.py:1816(nonzero)
1       1.476    1.476    1.476    1.476    {method 'nonzero' of 'numpy.ndarray' objects}
1       0.000    0.000    1.335    1.335    <_array_function__ internals>:2(transpose)
1       0.000    0.000    1.335    1.335    fromnumeric.py:601(transpose)
1       0.000    0.000    1.335    1.335    fromnumeric.py:39(_wrapit)
1       0.000    0.000    1.335    1.335    _asarray.py:14(asarray)
1       1.335    1.335    1.335    1.335    {built-in method numpy.array}
1       0.348    0.348    0.348    0.348    {method 'cuda' of 'torch._C._TensorBase' objects}
1       0.188    0.188    0.188    0.188    {method 'cpu' of 'torch._C._TensorBase' objects}
1       0.003    0.003    0.003    0.003    {method 'item' of 'torch._C._TensorBase' objects}
```



Motivation to profile

What is the purpose of profiling?

- finding performance issues
- optimizations and tuning
- fine-tuning

Do you need to deeply profile/optimize your script?

- training vs evaluation
- Are you developing your model or you already know the solution?
- Is your accuracy increasing with scaling?
- Is speed up worth your affords?
 - Is GPU utilization high?
 - Do you expect high improvements?
 - Are you planning to scale?



*“The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; **premature optimization is the root of all evil (or at least most of it) in programming.**”*

@Donald Knuth

Some frequent problems

- slow IO
 - Use designated PT/TF data loaders
 - Multithreading
- (extra) copy CPU ↔ GPU
 - Move all operations to device
 - Rewrite your code with using only PT/TF tensors
 - Use Numba / CuPy
 - Overlap copy and computation
 - Use asynchronous copy
- small batch size
 - Increase batch size until you totally load GPU
- high precision
 - Can you decrease precision to float/half?
 - Autotuning mixed precision in PT

Common profiler APIs

- Context manager

```
with profiler.profile() as prof:  
    model(inputs)
```

```
with torch.profiler.profile(  
    activities=[torch.profiler.ProfilerActivity.CPU,  
               torch.profiler.ProfilerActivity.CUDA],  
    schedule=torch.profiler.schedule(wait=0, warmup=1, active=3),  
    on_trace_ready=torch.profiler.tensorboard_trace_handler(dir_name)  
    ) as prof:
```

```
for _ in range(8):  
    model(inputs)
```

- Python module

```
python -m line_profiler train_GAN.py.lprof
```

```
python -m torch.utils.bottleneck example2/v3.py
```

- Decorator

```
@profile  
def train_loop(batch_size, n_training_iterations, models, opts, global_size):  
  
    logger = logging.getLogger()  
  
    rank = hvd.rank()  
    for i in range(n_training_iterations):
```

Let's play with examples!