Profiling notes

Cara Oct 17, 2024
notes by skye

Install tensorflow for tensorboard:

```
pip install -v tensorflow==2.17 tensorboard-plugin-profile
```
(can, and probably should, be done in its own conda environment), then run tensorboard --logdir=/tmp/tensorboard

Install NVIDIA Nsight

nsys profile –options ./code
Outputs .nsys-rep, .qdrep

Magic CPU Profiling flag:

import os
os.environ['XLA_FLAGS']='--xla_cpu_enable_xprof_traceme'

Notes

- Nsight (GPU profiling)
    - DtoH = device to host
        - Slow! Watch out for these transfers
        - e.g. cuMemcpyDtoHAsync
    - Do you have to do anything special to get the Nsight profile?
        - Lots of docs online
        - Option 1: connect directly to cluster running code, launch profiler from there
        - Option 2: nsys profile … → drag files into Nsight window
    - Does it require certain GPU/CPU type?
        - Any nvidia GPU should be good (not CPU)
    - Useful tips for reading profile?
        - #hlo_op=add.65 etc can be useful to see what it's doing (low level)
        - Also includes jit function name above
        - Google nvidia "cu" functions/kernels
    - JAX profiler GPU support in progress (coming soon…)
    - How painful to install Nsight on cluster?
        - Not too bad as Cara remembers it
        - Knowing commands was harder

- CPU profiling with jax.profiler
    - #1 tip: use magic env var!!!
    - create_perfetto_link=True causes hangs 😕
        - Set to False even if you want perfetto trace
        - Then in specified trace dir, find dir with correct date + time
        - Find trace.json in there, then drag into Perfetto website UI
    - jax.named_scope
        - Lets you put custom trace events that show up in trace viewer (sometimes works)
        - Shows up in device events (e.g. in "fusion" op)
        - Doesn't always show up
        - What if you put jit on top of func2 named_scope that's not showing up?
            - Nothing
            - Could maybe be a compiler feature request to not drop metadata/op names as much
        - What if you put named_scope outside of jit'd code?
            - Nothing
            - Maybe working as intended? Only works inside jit'd code (can be multiple nested jits)
- memory profiling
    - jax.jit(...).lower(...).compile().memory_analysis()
    - Returns a bunch of stats in a dictionary
    - Works on GPU and TPU, not so much on CPU (returns 0s)
- Can we use a C++ profiler for jax cpu?
    - Yes but the output doesn't seem useful (doesn't seem to include any jit'd functions)
- Kaylee's perfetto profiling
    - Profiling equinox autoencoder
    - **Don't forget .block_until_ready() !!!!**
        - Kaylee had to plumb out a jax array
        - Or can pass pytree into jax.block_until_ready(...)
        - Necessary because jax has async dispatch, need to make sure you're not turning off the profiler before execution finishes
            - async dispatch = jax functions immediately return
            - arrays are actually a promise/future that the value will eventually exist
            - usually jax automatically blocks, e.g. to print or do something else synchronous with the array
            - block_until_ready manually forces jax to block until the value is available (i.e. computation finished)
    - saw blank spot under function in tensorboard, used pyspy
        - overall very confusing. mostly needs knowledge of jax internals to understand.

        - looks like it's profiling the compilation? but being called multiple times… why getting cache misses?
        - function being compiled multiple times was defined inside another function that's being called repeatedly → new inner function being defined every time!
        - jax caches on function location in memory. if you redefine it, it's like a new function
        - jitting the outer function can fix (if possible)
- Understanding the profile output
    - Don't try to understand every block – it contains tons of internals! Try to look for hints that make sense (e.g. "compile")
    - Would be useful to have simpler logging of high-level execution timing
- What is perfetto?
    - Chrome trace viewer that kindly provides hosted website you can upload your own traces to