Summary & Recommendations

Technologies

Specific:

- PyTorch or Tensorflow as DL/AI platform
- Kubernetes as distributed platform and queuing system

General:

- GPUs as high performance compute accelerators
- sharded sequential objects on object stores

Simple Toolkit

Some simple tools used in this tutorial:

- `WebDataset` for rapid I/O
- `tarproc` for preprocessing
- `tensorcom` for distributed preprocessing
- `AIStore` for object caching / storage
- `kubetpl` as templating engine

Minimalist, simple, high performance tools based on open standards; no lock-in; work for desktop to data center.

Principles apply to other tools/environemnts as well.

Benchmarking and Profiling

- keep your most expensive components busy (usually: GPUs)
- use `nvidia-smi` and `htop`; keep GPU > 90% and CPU < 100%
- instrument your code and always keep track of I/O vs training
- measure with mock data source and mock training loop
- use more sophisticated tools when necessary
- simplify your code: it makes tuning much easier

Availability / Open Source Projects

http://pytorch.org

http://github.com/nvidia/aistore

http://github.com/tmbdev/webdataset

http://github.com/tmbdev/torchmore

http://github.com/tmbdev/tarproc

http://github.com/NVlabs/tensorcom

Questions / Discussion