Comparative analysis of distributed data parallel training packages

Andreea Zaharia — R244 Mini-project proposal
Motivation

Contribution of the project

- Deep learning models
- Distributed training across multiple workers/multiple GPUs
- Multiple packages implementing similar solutions
- Different optimisations or communication backends
- Lack of performance comparison between the most used frameworks
Open-source solutions
Native all-reduce modules

• Synchronous, decentralised approach
• All-Reduce pattern
• Overlap between communication and computation
• Native to their respective frameworks:
  • PyTorch DistributedDataParallel and DataParallel
  • Tensorflow MirroredStrategy and MultiWorkerMirroredStrategy
Open-source solutions

Horovod

- Supports Tensorflow, PyTorch, and MXNet.
- Ring all-reduce.
Horovod performance

Actual throughput: Tf vs Horovod vs ideal

Figure 6: A comparison of images processed per second with standard distributed TensorFlow and Horovod when running a distributed training job over different numbers of NVIDIA Pascal GPUs for Inception V3 and ResNet-101 TensorFlow models over 25GbE TCP.
Experiment ideas
Experiment design and implementation

• Implement the same models in Tensorflow and PyTorch
  • Training tasks: e.g. ResNet50, MobileNetV2, AlexNet
  • Multiple physical GPUs (prototyping on multiple logical GPUs)

• Run each with its native framework
• Run each with Horovod, instead of the native frameworks
• Vary communication backends (NCCL, GLOO).
Experiment ideas

Experiment metrics

• Comparative analysis of time and performance with respect to resources

• Example metrics:
  • Per-epoch accuracy as the number of GPUs increases
  • Speedup as the number of GPUs increases
  • Distributed train time per step (ms/step) and train time in total
  • Etc.
Objectives
Outcome and contribution

• Run the experiments presented (or equivalent ones)
• Compile the data into a performance evaluation
• Write a compelling comparative analysis of the three solutions
• Hypothesise on the causes of differences observed (if any)
Work plan
Preliminary reading and next steps

• Read and reviewed the PyTorch DDP paper.
• Read through existing evaluations of each framework.
• Explored options for setting up the multi-GPU system.

• Next step:
  • Implement the experiment code on the logical GPUs setup.
  • Configure an adjustable distributed system with feedback loop.
Questions? Suggestions?