Throughput Prediction of Asynchronous SGD in TensorFlow

Zhuojin Li  Wumo Yan  Marco Paolieri  Leana Golubchik

ICPE, April 23, 2020  icpe2020.spec.org
Training of Deep Neural Networks

Image Classification
Convolutional NN
[Oleksyshyn et al., 2012]

Speech Recognition
Recurrent NN + HMM
[Hinton et al., 2012]

Machine Translation
RNN Encoder-Decoder
[Sutskever et al., 2014]

Image Classification
Machine learning models with millions of adjustable parameters (weights)

Training with millions of labeled examples

Scaling up with GPUs
Asynchronous SGD with Parameter Server

Worker Nodes:
- Receive weights (downlink)
- Process batch of examples (compute)
- Send update (uplink)

Parameter Server: apply updates to weights (update)

Training throughput (examples/s) of Inception-v3 on AWS p3.2xlarge instances (NVIDIA V100 GPU)
Overlap of Computation and Communication

Weights are split into multiple tensors (arrays of weights)

Dependencies between communication and computation operations

[Lin et al.] A Model-Based Approach to Streamlining Distributed Training for Asynchronous SGD. MASCOTS’18

Simulation Approach to Throughput Prediction

Replay single-worker traces with multiple workers, accounting for reduced bandwidth.

Real traces: hundreds of operations
Profiling Challenges in TensorFlow

Problems of recorded durations in profiling traces
- Communication overhead included at the end
- Tensor transmission can be stopped and resumed
Estimation of Communication Overhead

**Linear Model**

\[
\text{transmission overhead} = \alpha \times \text{size} + \beta
\]

Parameters \(\alpha, \beta\) estimated once for each platform (private cluster, cloud CPU cluster, cloud GPU cluster).

Overhead due to tensor deserialization and copies between memory buffers.
Each stream is transmitted up to the size of the control window.

Next, pending streams are transmitted until completion.
Networking Optimizations

Multiplexing of multiple streams can increase the duration of a training step (if required tensors are delayed)

Flow control can be disabled in gRPC and transmissions ordered

[Hashemi et al.] TicTac: Accelerating distributed deep learning with communication scheduling. SysML’19
Simulation with Multiple Workers

Given a system configuration, including:

- Network bandwidth $B$
- Number of worker nodes $W$
- Number of parameter servers $M$
- Parameters $\alpha, \beta$ of communication overhead model

We simulate a sequence of SGD steps with $W$ workers by sampling steps from the profiling trace.

Each worker replays the sampled step (a graph of communication and computation operations) but ...

- Tensor transmissions are scheduled using our multiplexing model
- When multiple workers are in the downlink or uplink phase, bandwidth is shared equally
- Parsing overhead added after the reception of a tensor
Experimental Setup

Validation Platforms
- **Private cluster** of nodes with 4-core CPU, 16 GB RAM, 1 Gbps Ethernet
- **AWS c4.8xlarge** instances: 36-core CPU, 60 GB RAM, 10 Gbps Ethernet
- **AWS p3.2xlarge** instances: 8-core CPU, NVIDIA V100 GPU, 10 Gbps Ethernet

Platform Profiling
Estimate the parameters $\alpha, \beta$ of the communication overhead model

Job Profiling
For each job, run 100 steps with a single worker node to obtain profiling trace

Prediction
Run trace simulator with 2,...,$W$ workers for 1000 steps to evaluate the mean throughput along the trace.

Validation
Run clusters with 2,...,$W$ workers, skip 50 steps, compute throughput on next 50
Private CPU Cluster

Batch Sizes

- (a) AlexNet, batch size = 2
- (b) AlexNet, batch size = 4
- (c) AlexNet, batch size = 8
- (d) AlexNet, batch size = 16
- (e) AlexNet, batch size = 32

DNN Models

- (a) GoogLeNet, batch size = 1
- (b) GoogLeNet, batch size = 2
- (c) Inception-v3, batch size = 1
- (d) ResNet-50, batch size = 1
- (e) VGG-11, batch size = 4
- (f) VGG-11, batch size = 8
Private CPU Cluster: Networking Optimizations

Flow-control disabled, various orderings

(a) AlexNet, batch size = 2
(b) AlexNet, batch size = 4
(c) AlexNet, batch size = 8

Flow-control disabled

(d) Inception-v3, batch size = 1
(e) ResNet-50, batch size = 4
(f) VGG-11, batch size = 4

Flow-control disabled, TIC ordering

(a) Inception-v3, batch size = 1
(b) ResNet-50, batch size = 1
(c) VGG-11, batch size = 4

AlexNet, batch size = 4
Cloud Cluster: CPU-only

(a) AlexNet, batch size = 4
(b) AlexNet, batch size = 8
(c) AlexNet, batch size = 16
(d) ResNet-50, batch size = 1
(e) Inception-v3, batch size = 1
(f) GoogLeNet, batch size = 1
Cloud Cluster: GPU-enabled

(a) Inception-v3, batch size = 16
(b) Inception-v3, batch size = 32
(c) Inception-v3, batch size = 64
(d) Inception-v3, batch size = 128
(e) Inception-v4, batch size = 32
(f) Inception-v4, batch size = 64
(g) ResNet-50, batch size = 32
(h) ResNet-50, batch size = 64
(i) ResNet-50, batch size = 128
(j) ResNet-101, batch size = 32
(k) ResNet-101, batch size = 64
(l) ResNet-152, batch size = 32
(m) ResNet-152, batch size = 64
(n) VGG-11, batch size = 128
(o) VGG-11, batch size = 256
Limited improvement from two parameters servers in VGG-11 (h) due to uneven split of DNN weights.
Cost and Time Savings

Prediction is faster and less expensive (simulation of the computation, on CPU nodes instead of p3.2xlarge)
Conclusions

- Approach to the prediction of training throughput of asynchronous SGD in TensorFlow
  - Tracing information from minimal single-worker profiling
  - Discrete-event simulation to generate synthetic traces with multiple worker nodes
- Faster and less expensive than direct measurements with multiple workers
- Good accuracy across DNN models, batch sizes, and platforms, networking optimizations
- Future work: more fine-grained analytical models