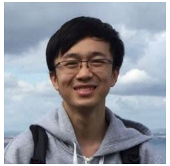
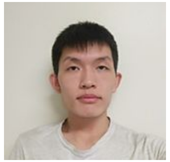
Throughput Prediction of Asynchronous SGD in TensorFlow


Zhuojin Li


Wumo Yan


Marco Paolieri


Leana Golubchik

USC University of Southern California

qed.usc.edu

ICPE, April 23, 2020
icpe2020.spec.org

Training of Deep Neural Networks

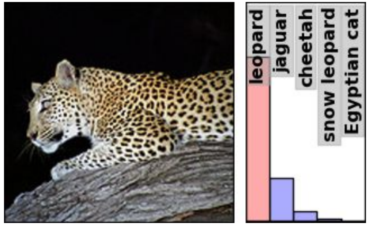
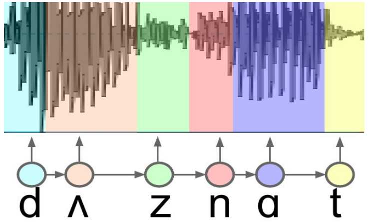
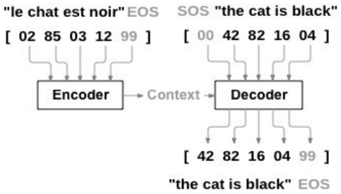


Image Classification
Convolutional NN
[Krizhevsky et al., 2012]



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



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

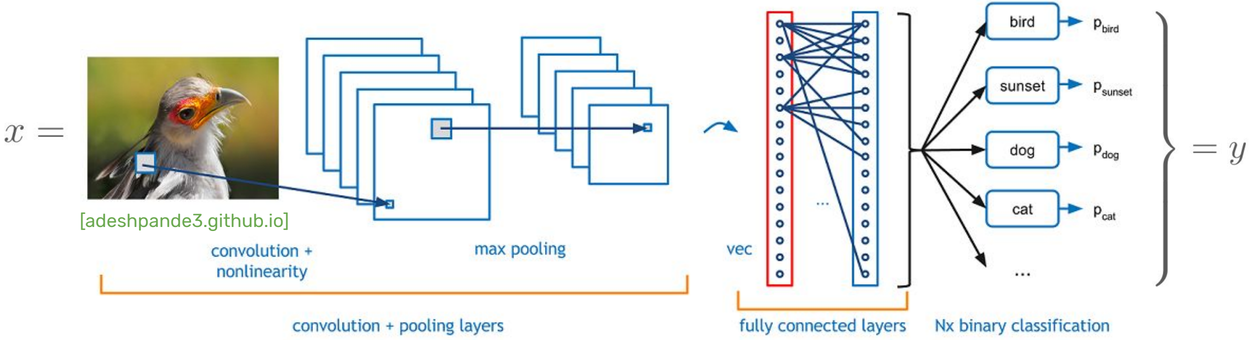
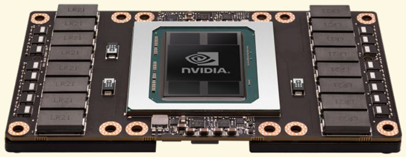


Image Classification

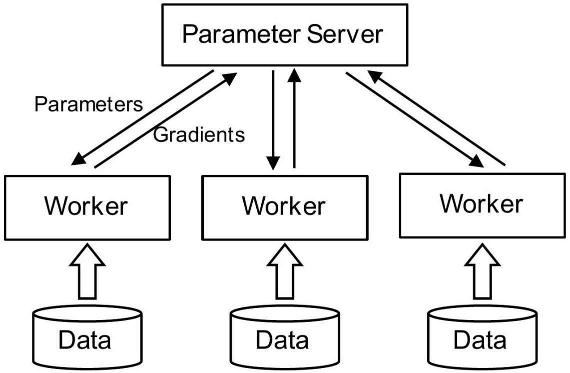
Machine learning models with millions of adjustable parameters (**weights**)

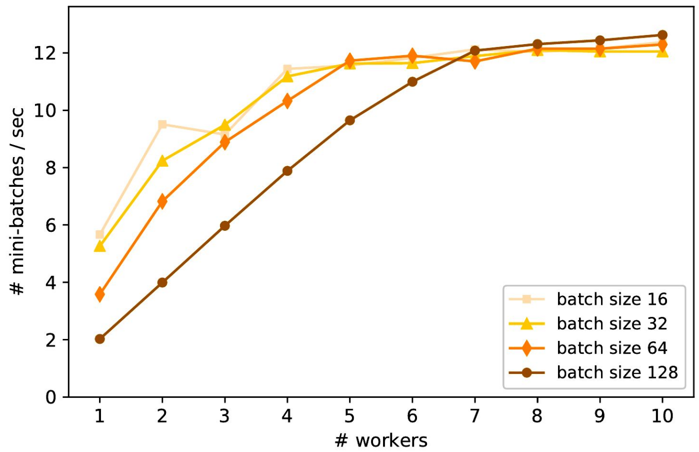
Training with millions of **labeled examples**

Scaling up with **GPUs**



Asynchronous SGD with Parameter Server





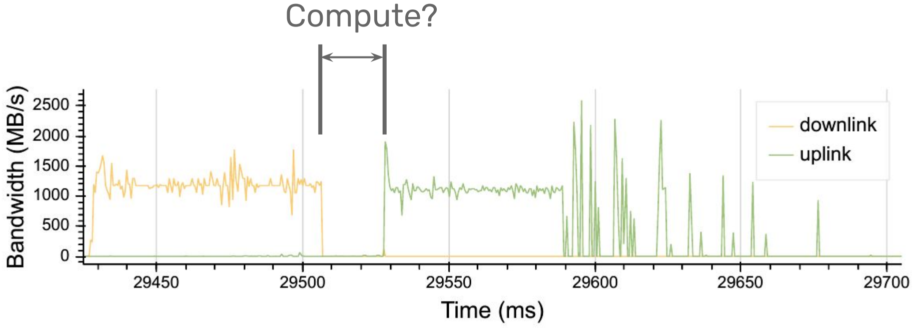
Worker Nodes:

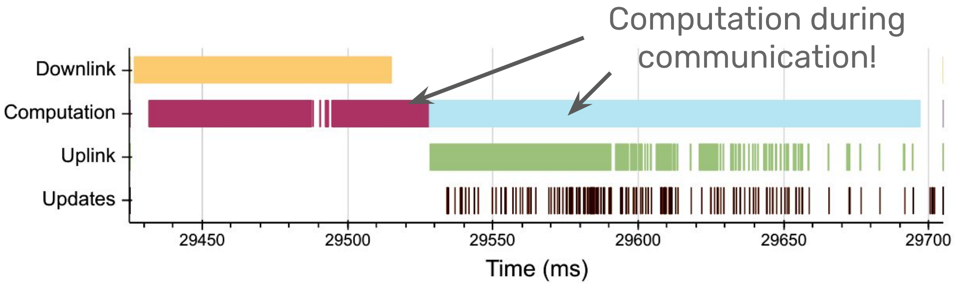
- Receive weights (*downlink*)
- Process batch of examples (*compute*)
- Send update (*uplink*)

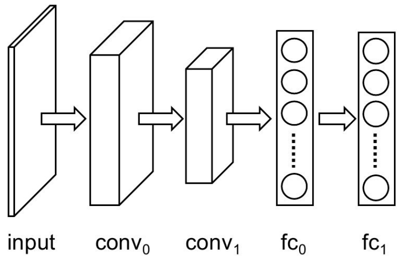
Parameter Server: apply updates to weights (*update*)

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

Overlap of Computation and Communication

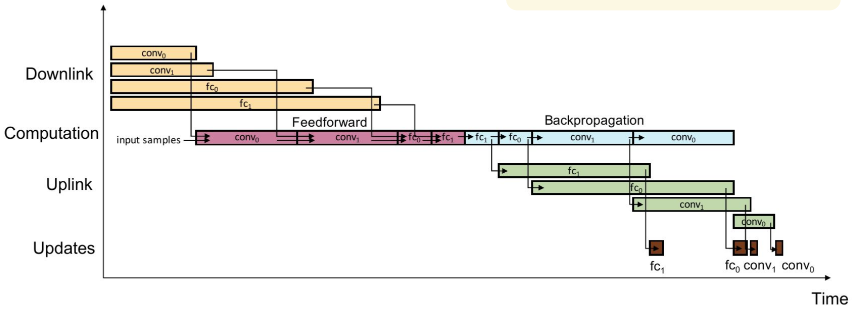






Weights are split into multiple **tensors** (arrays of weights)

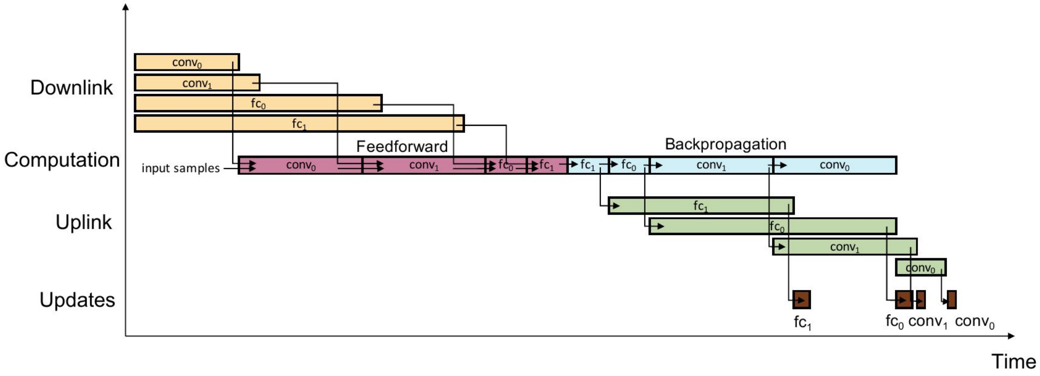
Dependencies between communication and computation operations

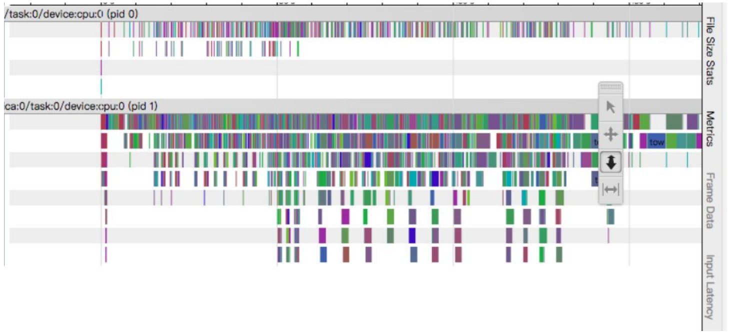


[Lin et al.] **A Model-Based Approach to Streamlining Distributed Training for Asynchronous SGD.** MASCOTS'18

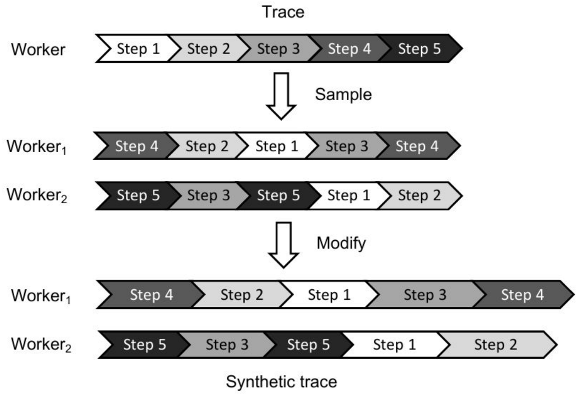
[Zheng et al.] **Cynthia: Cost-Efficient Cloud Resource Provisioning for Predictable Distributed DNN Training.** ICPP'19

Simulation Approach to Throughput Prediction



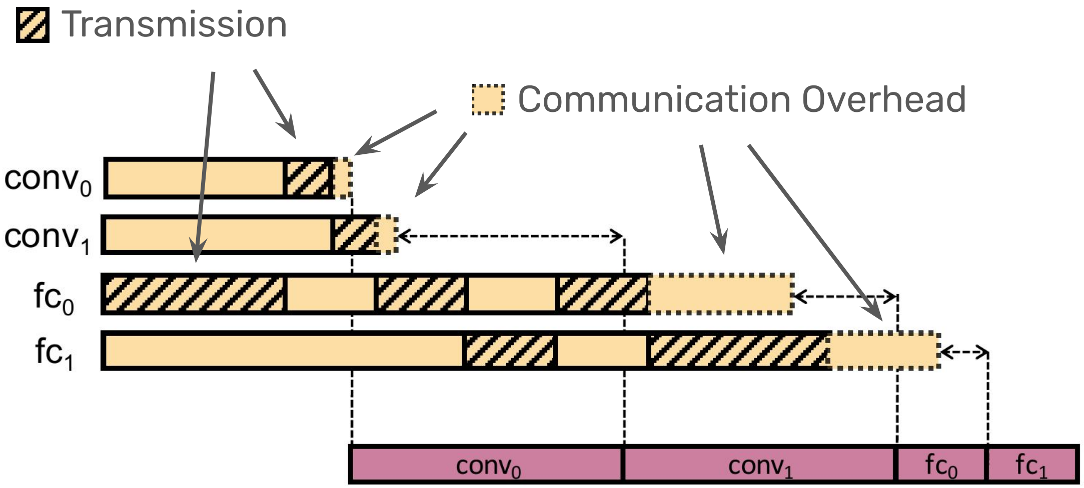


Real traces: hundreds of operations



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

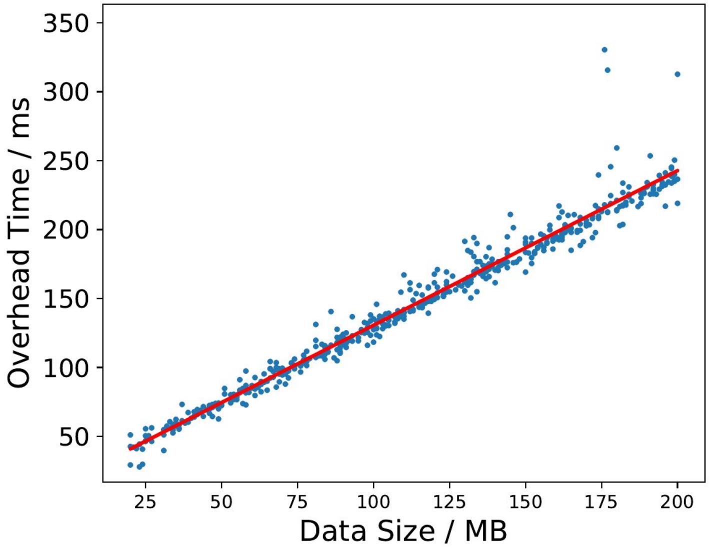
Profiling Challenges in TensorFlow



Problems of recorded durations in profiling traces

- Communication overhead included at the end
- Tensor transmission can be stopped and resumed

Estimation of Communication Overhead



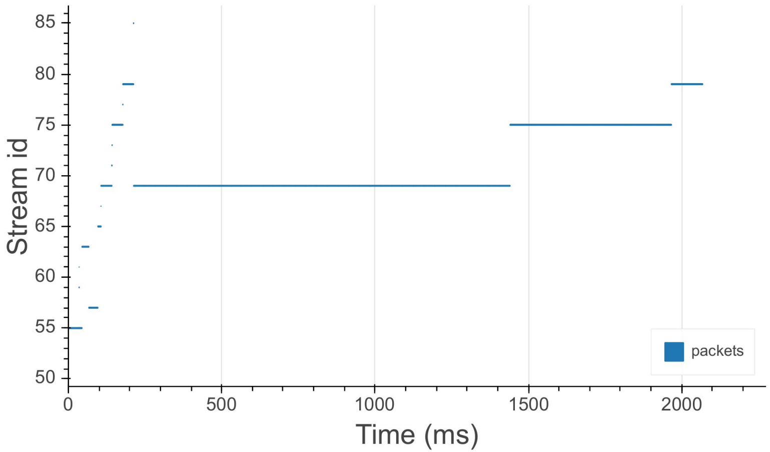
Linear Model

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

Parameters α, β estimated once for each platform (private cluster, cloud CPU cluster, cloud GPU cluster).

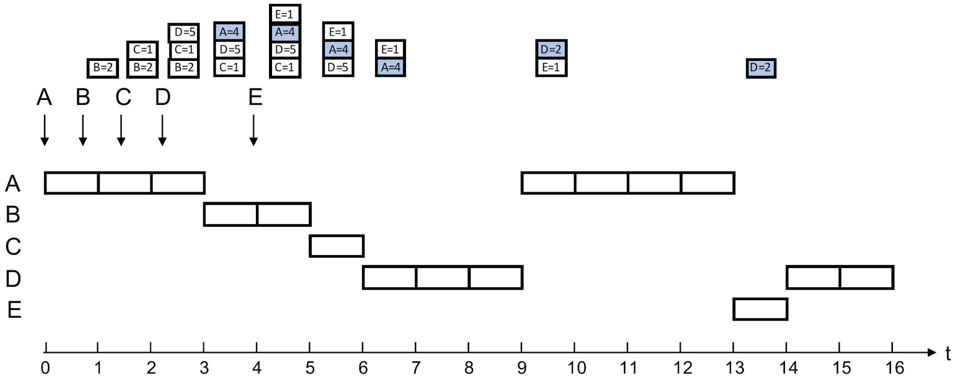
Overhead due to tensor deserialization and copies between memory buffers.

Multiplexing Model of Downlink and Uplink



Each stream is transmitted up to the size of the control window.

Next, pending streams are transmitted until completion.

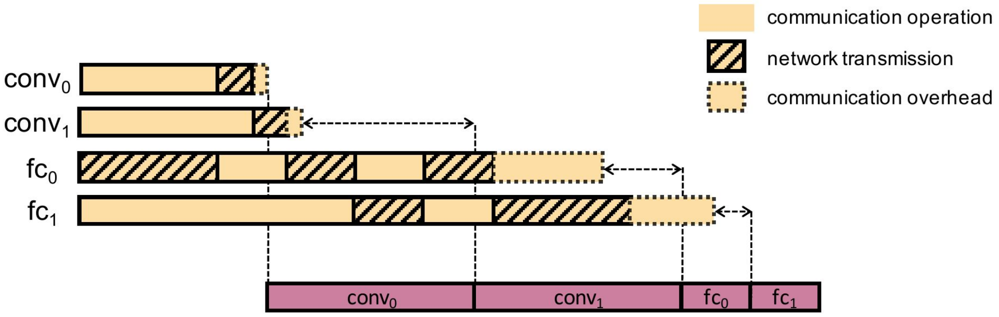


DNN Model	End-time Prediction Error	Private Cluster	AWS Cloud
AlexNet	Mean	1.82%	2.89%
	95th Percentile	3.35%	9.71%
GoogLeNet	Mean	1.69%	3.43%
	95th Percentile	3.74%	9.14%
ResNet-50	Mean	1.26%	4.36%
	95th Percentile	2.32%	9.70%
Inception-V3	Mean	1.02%	9.23%
	95th Percentile	3.92%	20.98%

Networking Optimizations

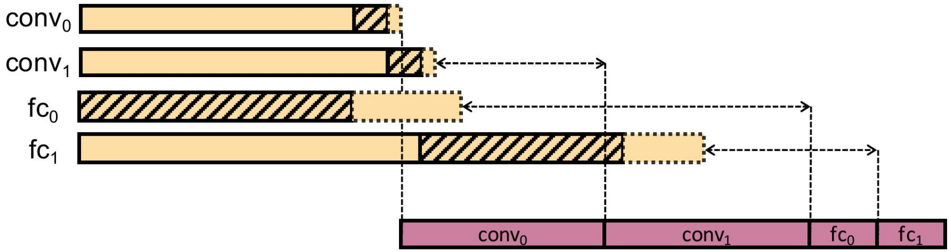
Multiplexing of multiple streams can increase the duration of a training step (if required tensors are delayed)

Flow control can be disabled in gRPC and transmissions ordered

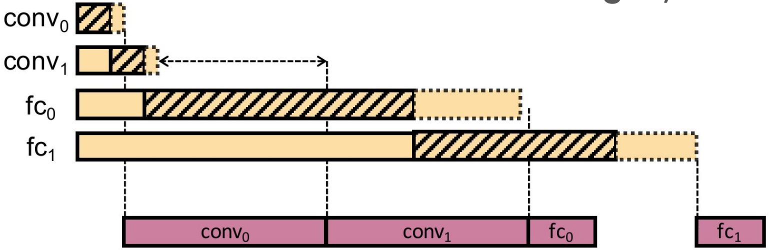


Default

[Hashemi et al.] **TicTac: Accelerating distributed deep learning with communication scheduling.** SysML'19



Flow-control Disabled



Flow-control Disabled, TIC ordering

Simulation with Multiple Workers

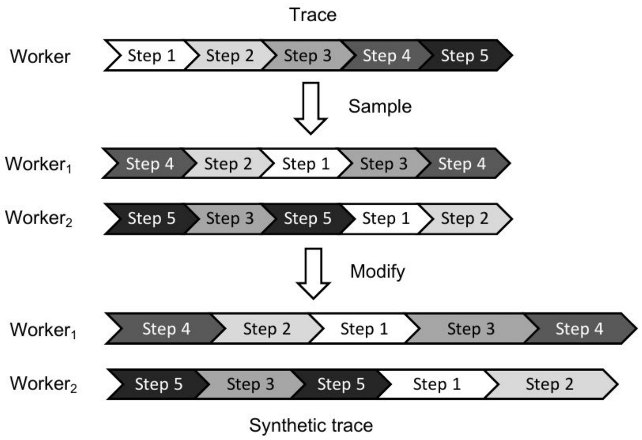
Given a system configuration, including:

- Network bandwidth B
- Number of worker nodes W
- Number of parameter servers M
- Parameters α, β 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 α, β of the communication overhead model

Job Profiling

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

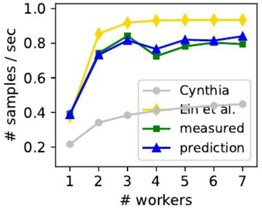
Prediction

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

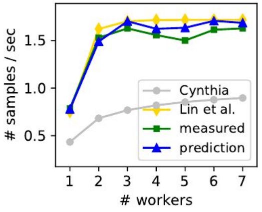
Validation

Run clusters with 2,...,W workers, skip 50 steps, compute throughput on next 50

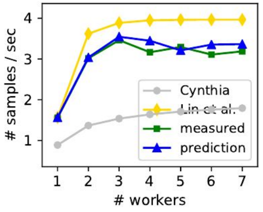
Private CPU Cluster



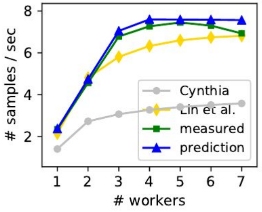
(a) AlexNet, batch size = 2



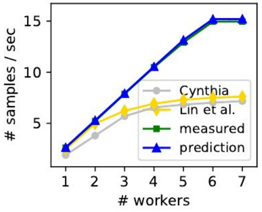
(b) AlexNet, batch size = 4



(c) AlexNet, batch size = 8

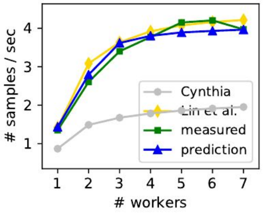


(d) AlexNet, batch size = 16

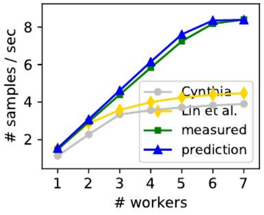


(e) AlexNet, batch size = 32

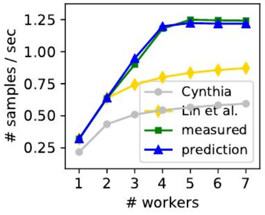
Batch Sizes



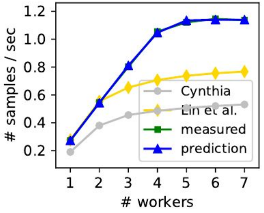
(a) GoogLeNet, batch size = 1



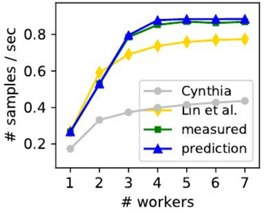
(b) GoogLeNet, batch size = 2



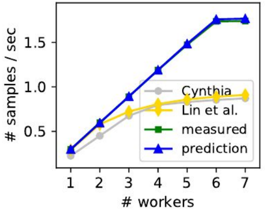
(c) Inception-v3, batch size = 1



(d) ResNet-50, batch size = 1



(e) VGG-11, batch size = 4

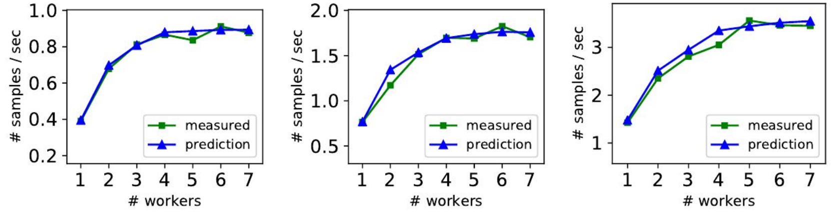


(f) VGG-11, batch size = 8

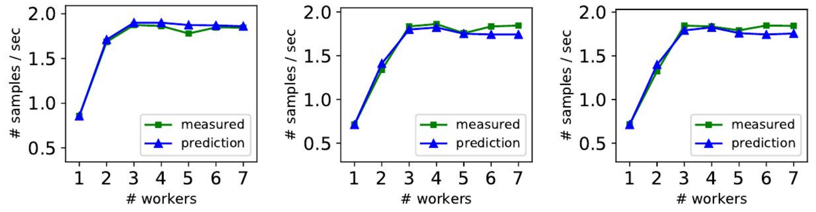
DNN Models

Private CPU Cluster: Networking Optimizations

AlexNet, batch size = 4

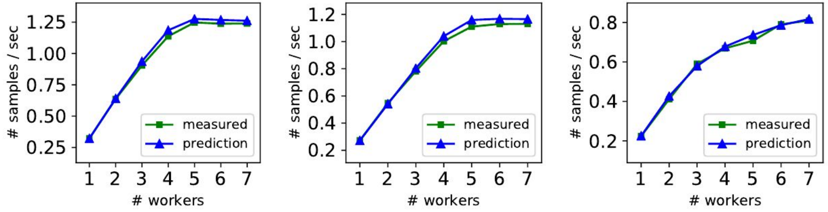


(a) AlexNet, batch size = 2 (b) AlexNet, batch size = 4 (c) AlexNet, batch size = 8



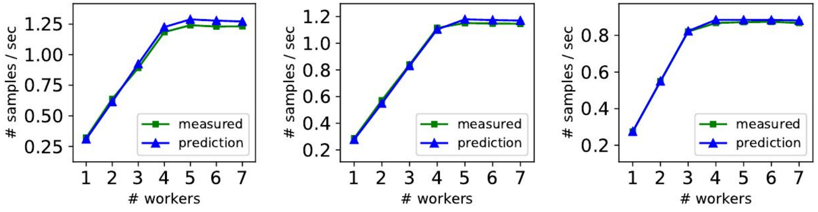
(a) TIC Order (b) TIC Reverse Order (c) Random Order

Flow-control disabled, various orderings



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

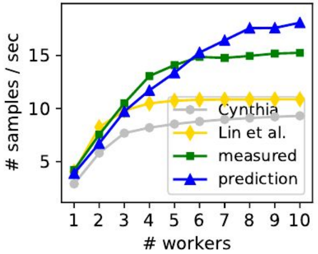
Flow-control disabled



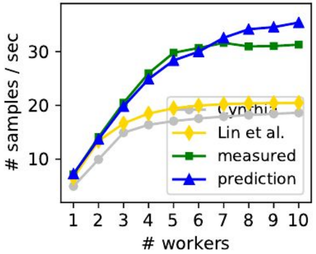
(a) Inception-v3, batch size = 1 (b) ResNet-50, batch size = 1 (c) VGG-11, batch size = 4

Flow-control disabled, TIC ordering

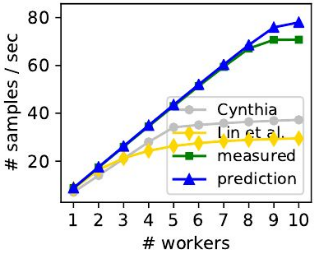
Cloud Cluster: CPU-only



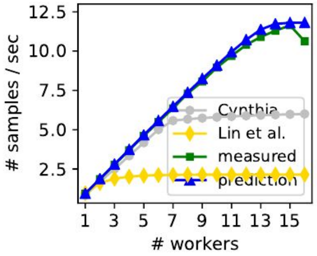
(a) AlexNet, batch size = 4



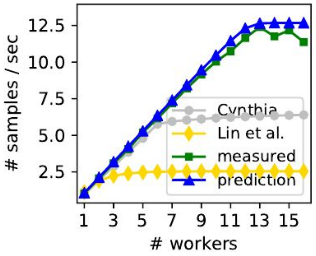
(b) AlexNet, batch size = 8



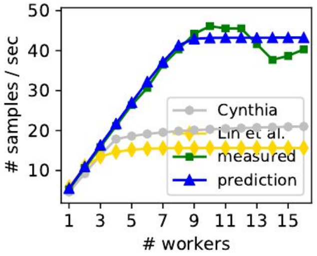
(c) AlexNet, batch size = 16



(d) ResNet-50, batch size = 1

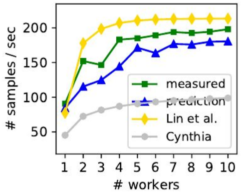


(e) Inception-v3, batch size = 1

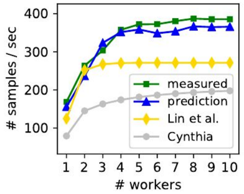


(f) GoogLeNet, batch size = 1

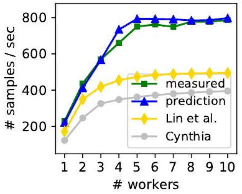
Cloud Cluster: GPU-enabled



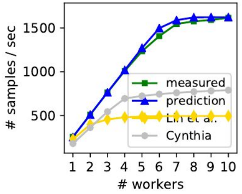
(a) Inception-v3, batch size = 16



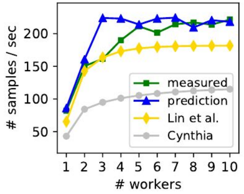
(b) Inception-v3, batch size = 32



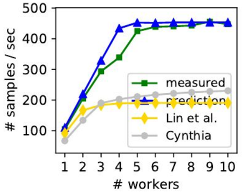
(c) Inception-v3, batch size = 64



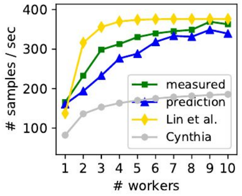
(d) Inception-v3, batch size = 128



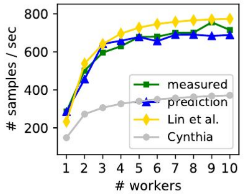
(e) Inception-v4, batch size = 32



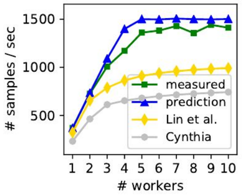
(f) Inception-v4, batch size = 64



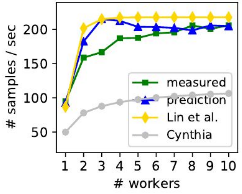
(g) ResNet-50, batch size = 32



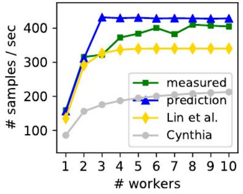
(h) ResNet-50, batch size = 64



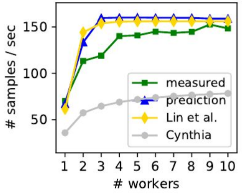
(i) ResNet-50, batch size = 128



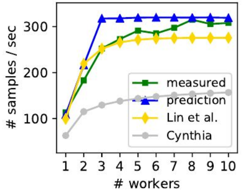
(j) ResNet-101, batch size = 32



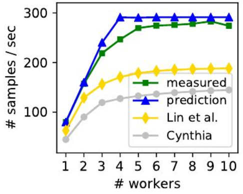
(k) ResNet-101, batch size = 64



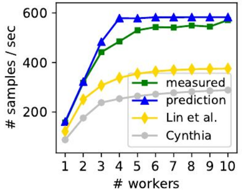
(l) ResNet-152, batch size = 32



(m) ResNet-152, batch size = 64

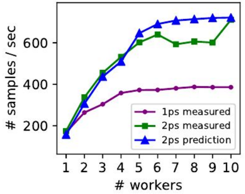


(n) VGG-11, batch size = 128

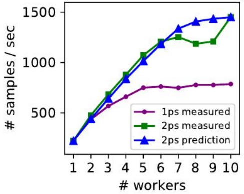


(o) VGG-11, batch size = 256

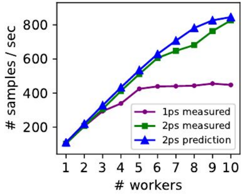
Cloud Cluster: GPU-enabled, two PS



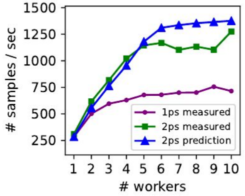
(a) Inception-v3, batch size = 32



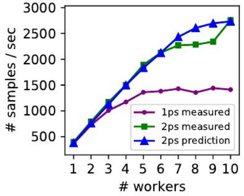
(b) Inception-v3, batch size = 64



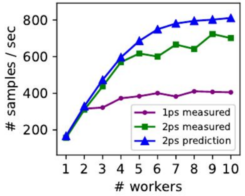
(c) Inception-v4, batch size = 64



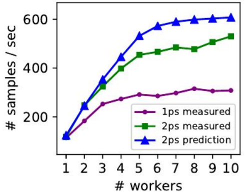
(d) ResNet-50, batch size = 64



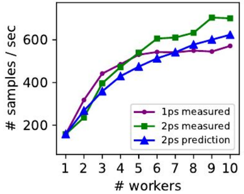
(e) ResNet-50, batch size = 128



(f) ResNet-101, batch size = 64



(g) ResNet-152, batch size = 64



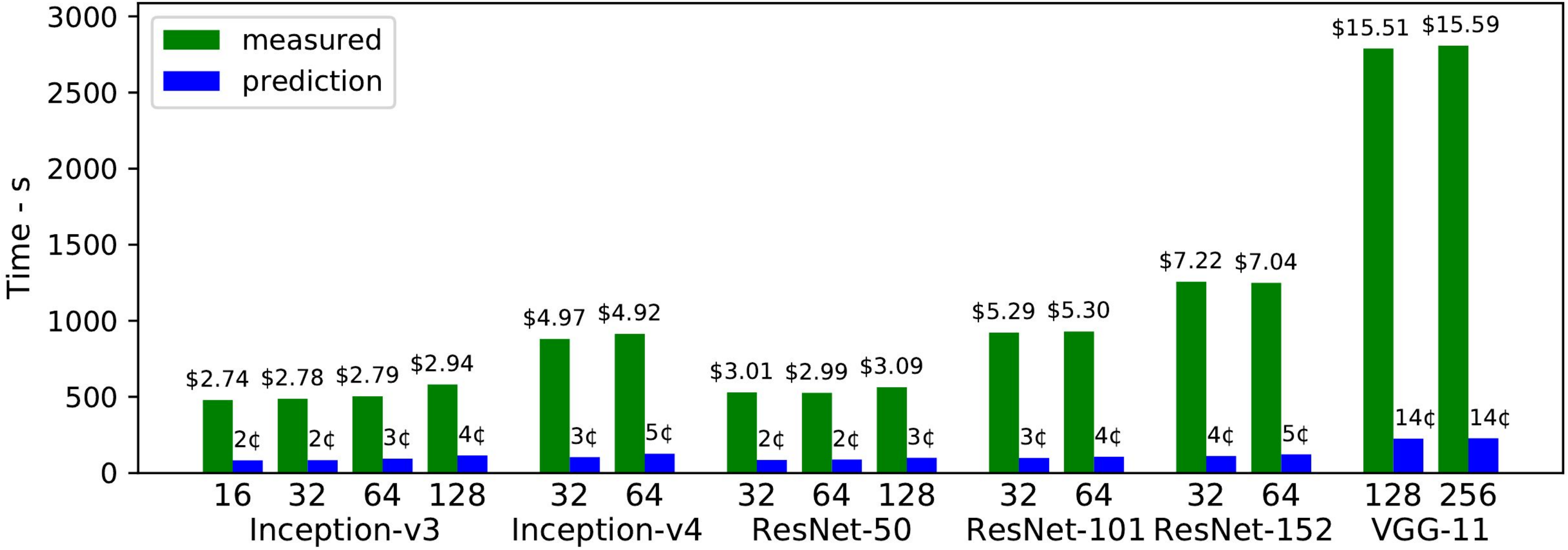
(h) VGG-11, batch size = 256

VGG-11 Weights Partition

PS_1	PS_2
conv0/conv2d: 6.8KB	conv0/biases: 256B
conv1/biases: 512B	conv1/conv2d: 0.3MB
conv2/conv2d: 1.1MB	conv2/biases: 1KB
conv3/biases: 1KB	conv3/conv2d: 2.2MB
conv4/conv2d: 4.5MB	conv4/biases: 2KB
conv5/biases: 2KB	conv5/conv2d: 9MB
conv6/conv2d: 9MB	conv6/biases: 2KB
conv7/biases: 2KB	conv7/conv2d: 9MB
fc0/weights: 392MB	fc0/biases: 16KB
407MB	fc1/weights: 64MB
	fc1/biases: 16KB
	fc2/weights: 15.6MB
	fc2/weights: 16KB
	100MB

Limited improvement from two parameters servers in VGG-11 (h) due to uneven split of DNN weights

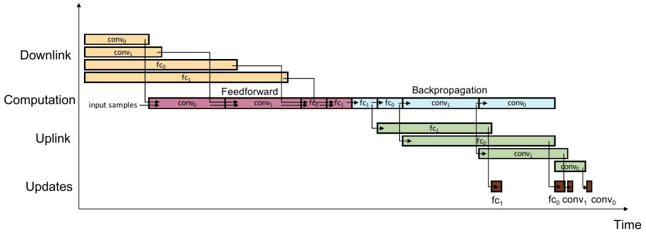
Cost and Time Savings

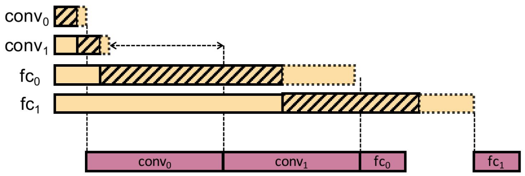


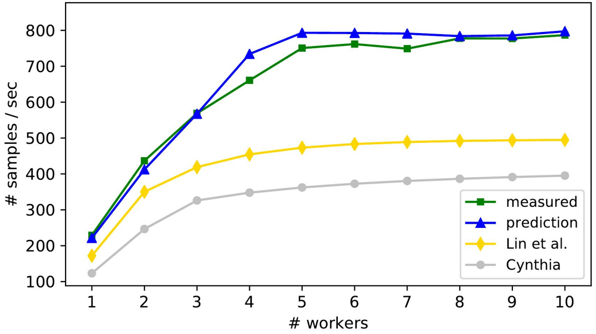
Prediction is faster and less expensive (simulation of the computation, on CPU nodes instead of `p3.2xlarge`)

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







Inception-V3, batch=64, `p3.2xlarge`