Training ImageNet in 15 Minutes With ChainerMN: A Scalable Distributed Deep Learning Framework

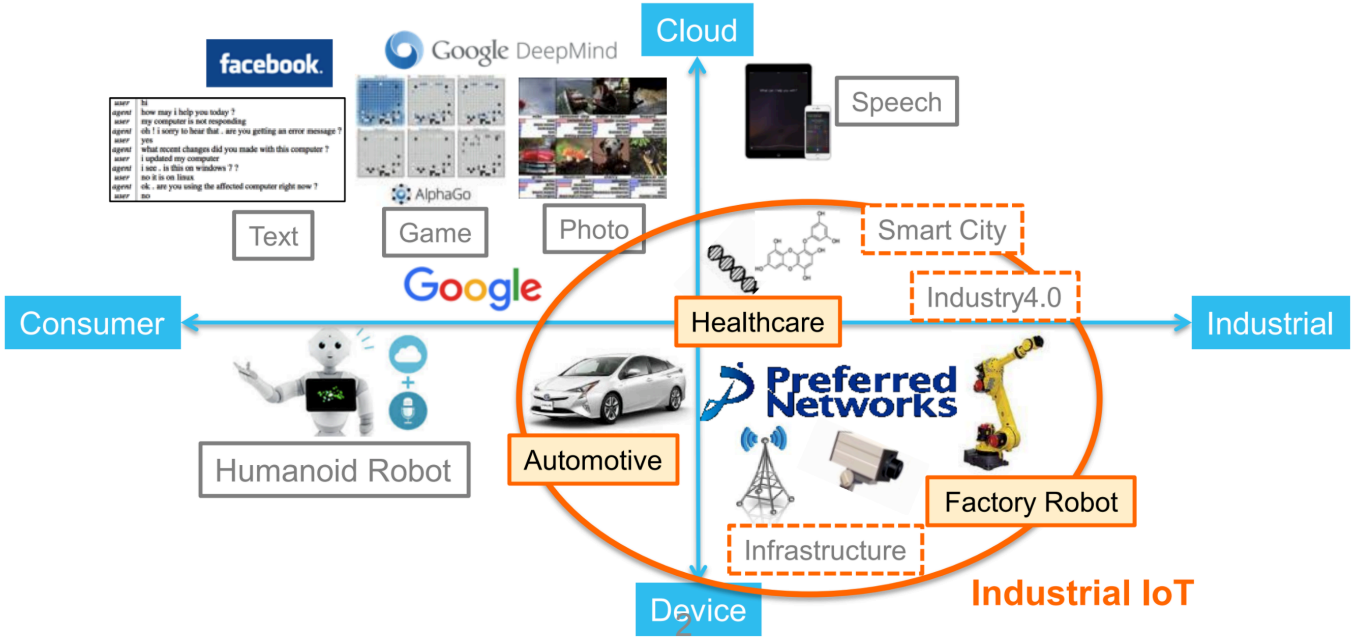
Takuya Akiba, Shuji Suzuki, Keisuke Fukuda, and Kota Uenishi

Preferred Networks, Inc.

Who are we?

Preferred Networks, Inc. (PFN):

A Tokyo-based Deep Learning & IoT company



Research and engineering in PFN

- Strong Engineering partnership

and more!

- Active research
  - Constantly publish papers in top-tier ML conferences
  - Including 3 papers in ICLR'18

Preferred Networks

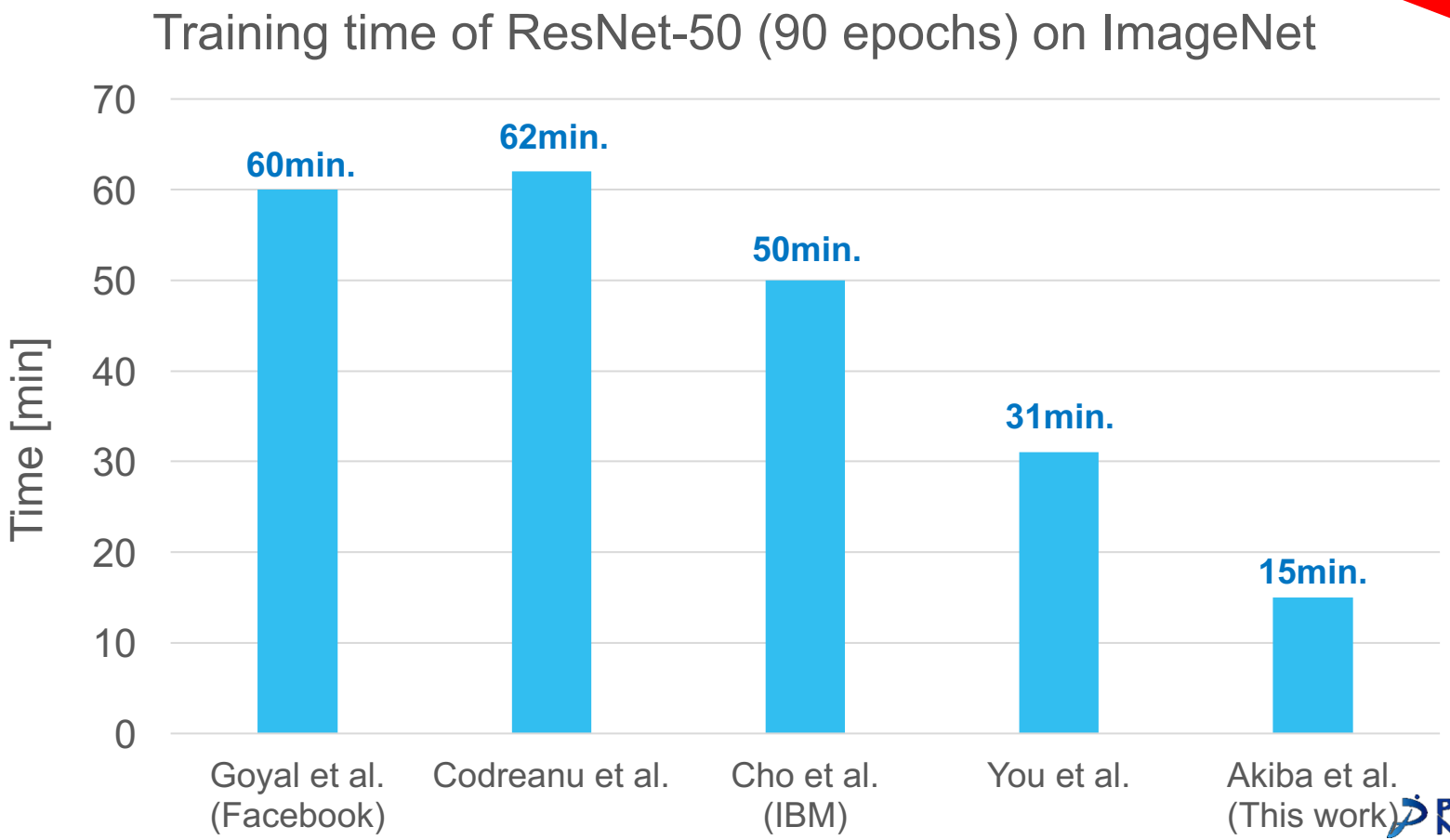
“Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions”
arXiv:1710.06280

Preferred Networks

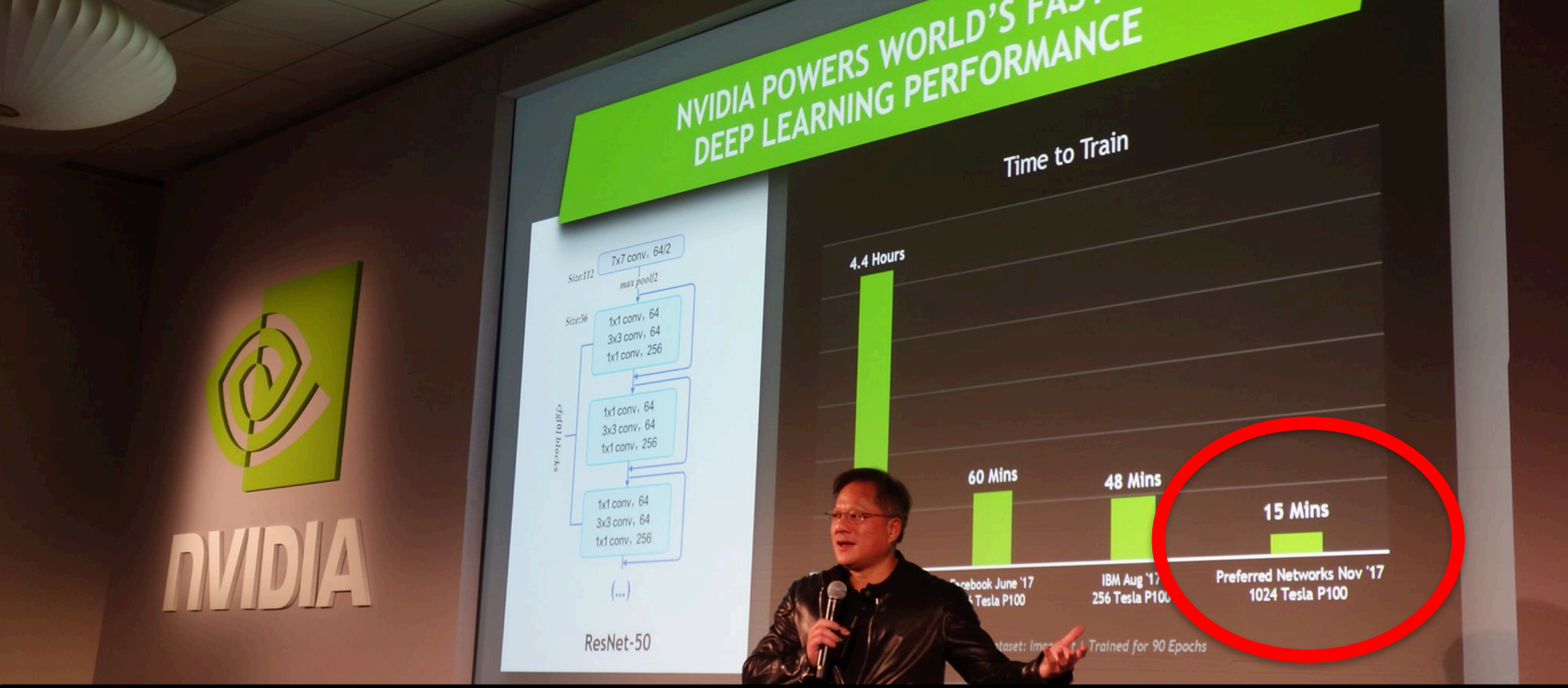
Distributed Deep Learning

Preferred Networks

Details later !



Preferred Networks



Jen-Hsun Huang

NVIDIA CEO, at SC'17

What we want:

Shorter training time

It is always better

No questions? ☺

Preferred Networks

Answer: Not Really.

Even if training time is faster…

- Model accuracy is degraded => 😥
- Programming is hard => 😥

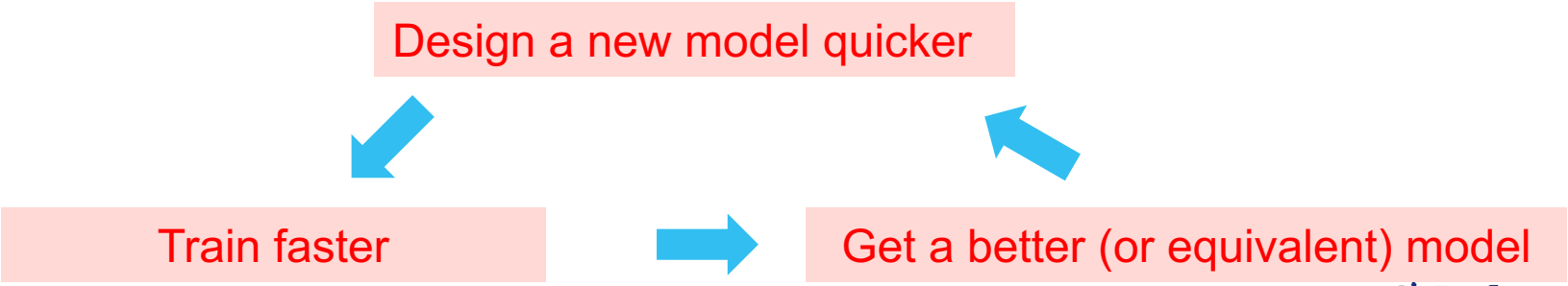
Increasing the training throughput is easy…
But it does not necessarily make R&D faster

Preferred Networks

What we really want:

~~Shorter training time~~

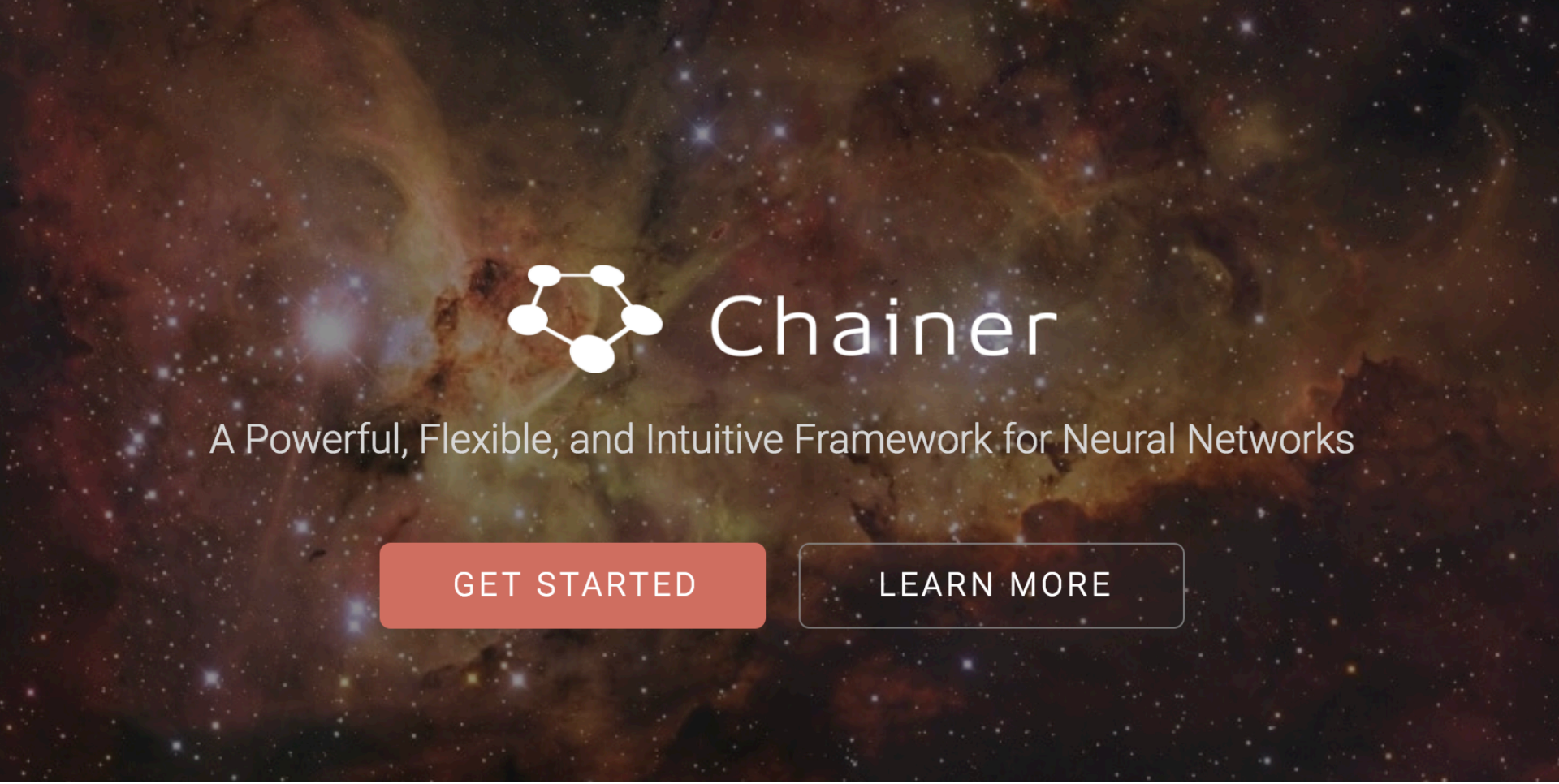
Faster R&D cycle



Preferred Networks

Background of the ImageNet challenge

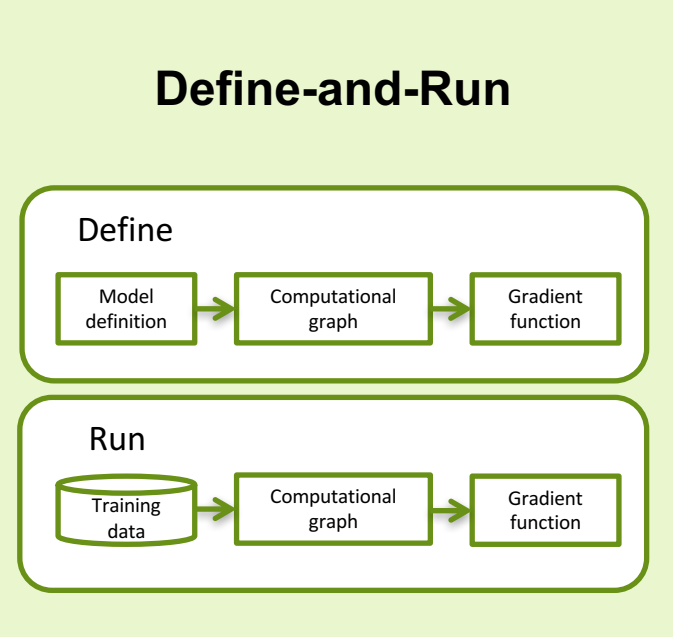
Preferred Networks



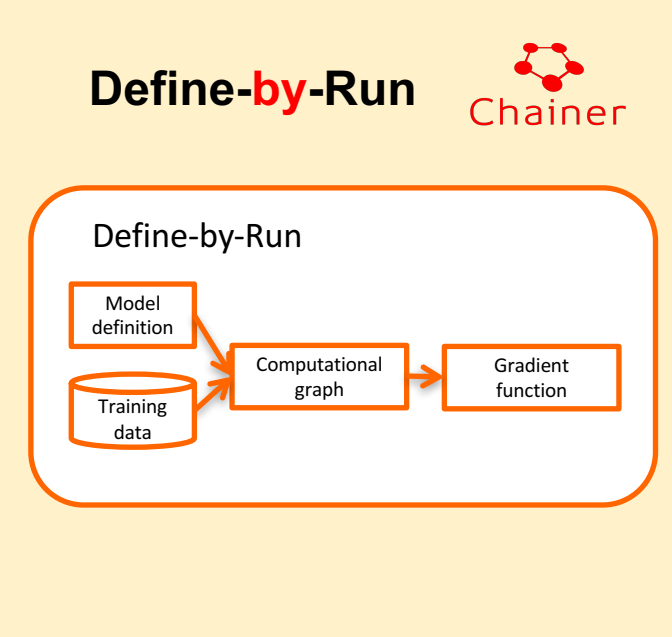
https://chainer.org/

Preferred Networks

Chainer:
A Flexible Deep Learning Framework



Caffe2, TensorFlow etc.



PyTorch, TensorFlow(Eager Execution) etc.

Preferred Networks

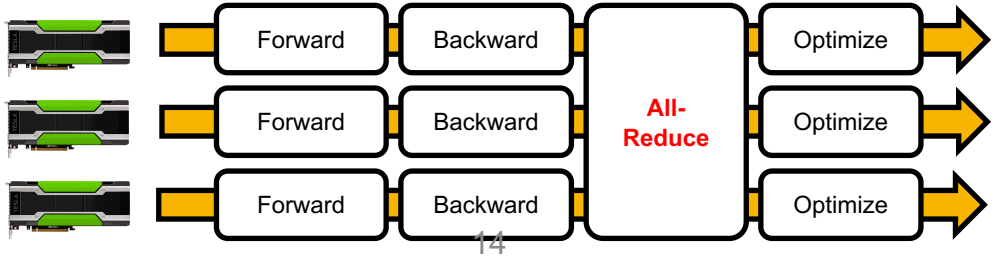
ChainerMN: Distributed Training with Chainer

- Add-on package for Chainer
- Enables multi-node distributed deep learning using NVIDIA NCCL2

Features

- **Scalable**: Near-linear scaling with hundreds of GPUs
- **Flexible**: Even GANs, dynamic NNs, and RL are applicable

Distributed Training with ChainerMN



Preferred Networks

MN-1: an in-house supercomputer

- NVIDIA Tesla P100 × **1024**
  8 GPUs per node, 128 nodes in total
- Inter-connected by InfiniBand FDR
  2 HCAs per node, tree-like topology

The number of employees is about 120, so this is relatively very large for us!

Fun!

(Do you think it's crazy?)

Preferred Networks

OK, let's tackle the ImageNet problem
with our 1024 P100 GPUs!

Our goal: **15 min**.

- Training CNNs on ImageNet is very time consuming
- Original ResNet-50 paper : **29 hours using 8 GPUs**
- Notable achievement by Goyal et al.: **1 hour using 256 GPUs.**

⇒ We can use 1024 GPUs. 1 hour / 1024 * 256 = 15 mins. 🤔
Sounds easy? ABSOLUTELY NO!

Technical Challenges:

1. Large batch problem
2. Performance Scalability (while keeping flexibility)
3. Troubles ☹

"Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes"
(arXiv:1711.04325)

Preferred Networks

Challenges in the “ImageNet-15min challenge”

1. The “large batch” problem
   - “Sharp minima”
   - Fewer training iterations
2. Performance scalability
3. Technical issues ☹

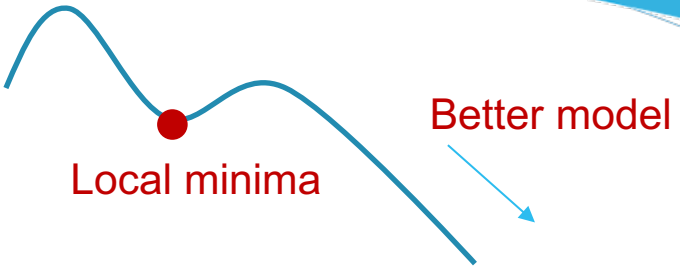
Preferred Networks

Challenges in the “ImageNet-15min challenge”

1. The “large batch” problem
   - “Sharp minima”
   - Fewer training iterations
2. Performance scalability
3. Technical issues ☹

Preferred Networks

Challenge 1:
The “large batch” problem



“It has been observed in practice that when using a larger batch there is a significant degradation in the quality of the model, as measured by its ability to generalize”

From Keskar et al.
“On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima”

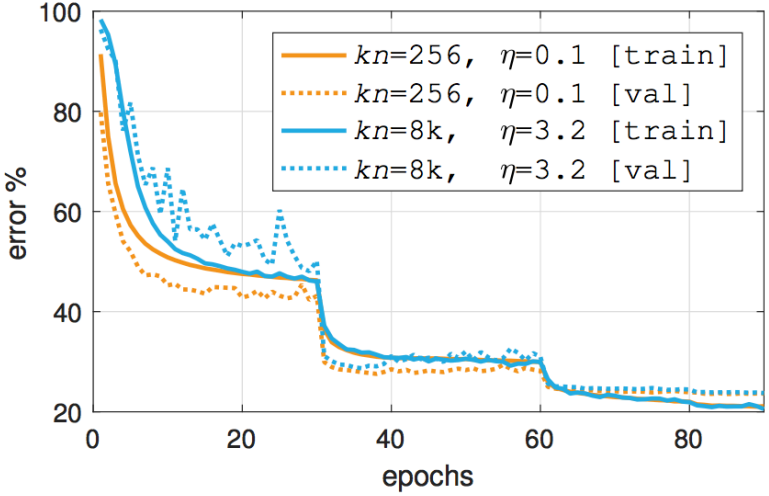
1. Computed gradients in each iteration is an average of larger number of samples
   → gradients are “less stochastic”, which makes it difficult to escape from local minima

2. Total number of iterations (=updates) is smaller

Preferred Networks

“Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour”

arXiv:1706.02677

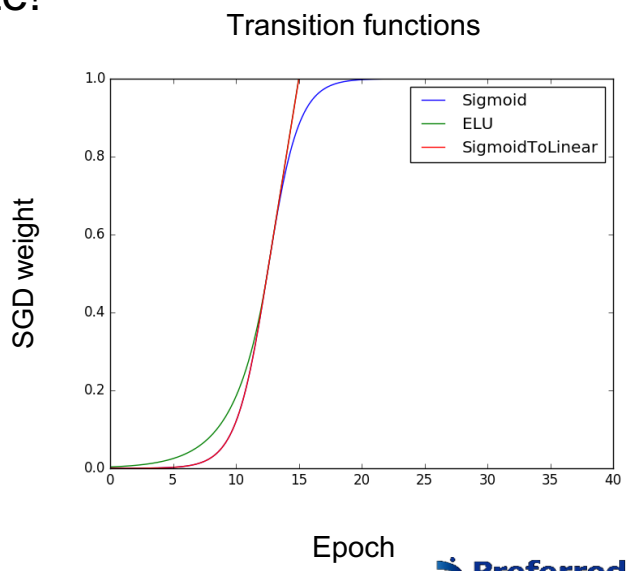


- Linear scaling rule:
  - “If minibatch-size is k times larger, increase learning rate by k times”
- Gradual warmup scheme

Preferred Networks

Additional techniques for 1024 GPUs:

- We needed to go further: 32*1024 = 32k batchsize!
- RMSprop Warmup
  - SGD: generalizes well, but converges slower.
  - We start the training with RMSprop, then gradually transition to SGD.
- Batch normalization without moving averages

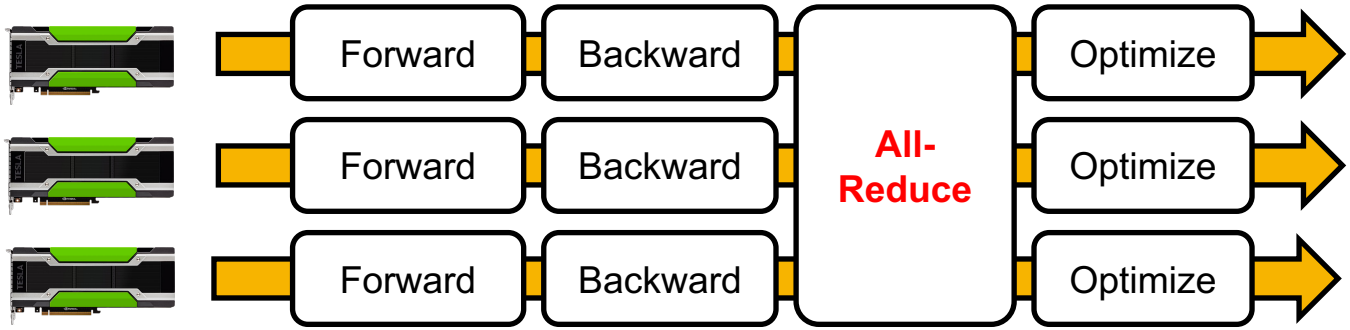


Challenges in the “ImageNet-15min challenge”

1. The “large batch” problem
   - “Sharp minima”
   - Fewer training iterations
2. Performance scalability
3. Technical issues ☹

Preferred Networks

Challenge 2:
Performance scalability



`Allreduce` operation is critical for scalability

How to overcome scalability challenge?

Improve the All-reduce bottleneck

- Use faster communication routines
- Reduce communication data

Preferred Networks

How to overcome scalability challenge?

Improve the All-reduce bottleneck

- Use faster communication routines
- Reduce communication data

Preferred Networks

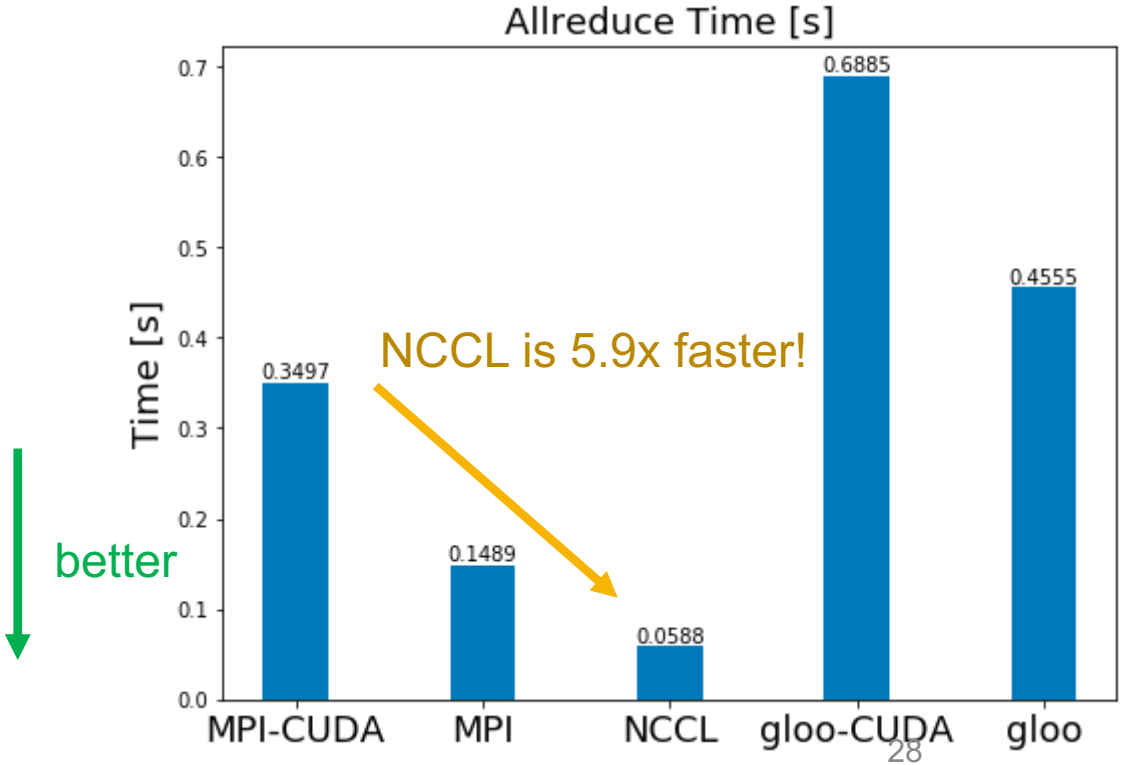
Faster communication routines

- ChainerMN is built on top of MPI
  - Just call `MPI_Allreduce()` and nothing else to do? (MPI should be well tuned… Agreed?)
  - Bandwidth efficiency of MPI_Allreduce with GPUDirect : **10%**

(as of the experiment, Open MPI 2.1.2, Infiniband FDR)

Preferred Networks

NCCL : Nvidia Collective Communication Library



64MB Allreduce (MPI_SUM), 2 processes,

- Open MPI 2.1.2
  (default configuration: no advanced tuning)
- Over Infiniband FDR(4x)

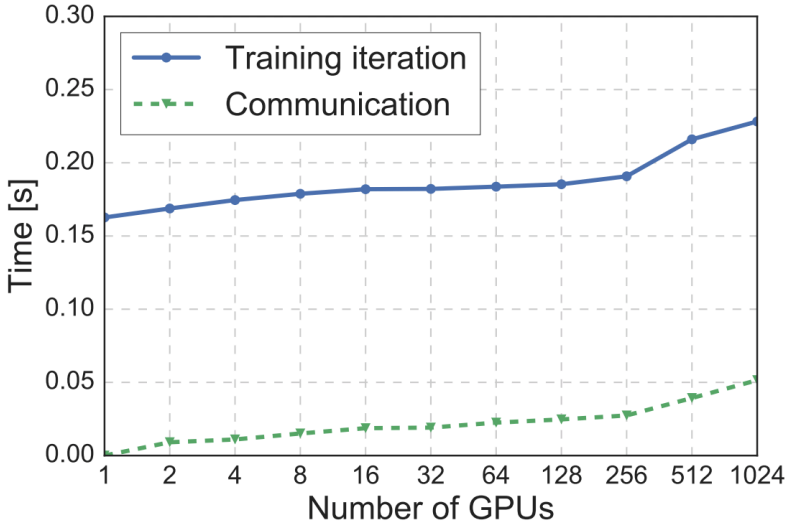
”MPI” :
Allreduce an array on host memory
(ordinary MPI_Allreduce)

“MPI-CUDA” :
Allreduce an array on GPU’s device memory
(You can pass device memory pointer to MPI routines)

Preferred Networks

Further optimizations for NCCL:

- Improve network performance
  - GPU Direct P2P & RDMA
  - Manual ring configuration



Preferred Networks

How to overcome scalability challenge?

Improve the All-reduce bottleneck

- Use faster communication routines
- Reduce communication data

Preferred Networks

Reduce communication data: use FP16

Compute gradients

Convert FP32 to FP16

Allreduce (with NCCL)

Convert FP16 to FP32 and update

The accuracy degradation is negligible!!

Preferred Networks

Challenges in the “ImageNet-15min challenge”

1. The “large batch” problem
   - “Sharp minima”
   - Fewer training iterations
2. Performance scalability
3. Technical issues ☹

Preferred Networks

Crash… Crash… Crash…

- The more you buy, the more you crash

- ≧ 192 GPUs: Crash → NCCL2: too many file descriptors
- ≧ 784 GPUs: Crash → Bug in ChainerMN
- ≧ 944 GPUs: Crash → NCCL2: stack overflow

- Some GPUs were broken, as well

(As of NCCL 2.0.5)

Preferred Networks

Crash… Crash… Crash…

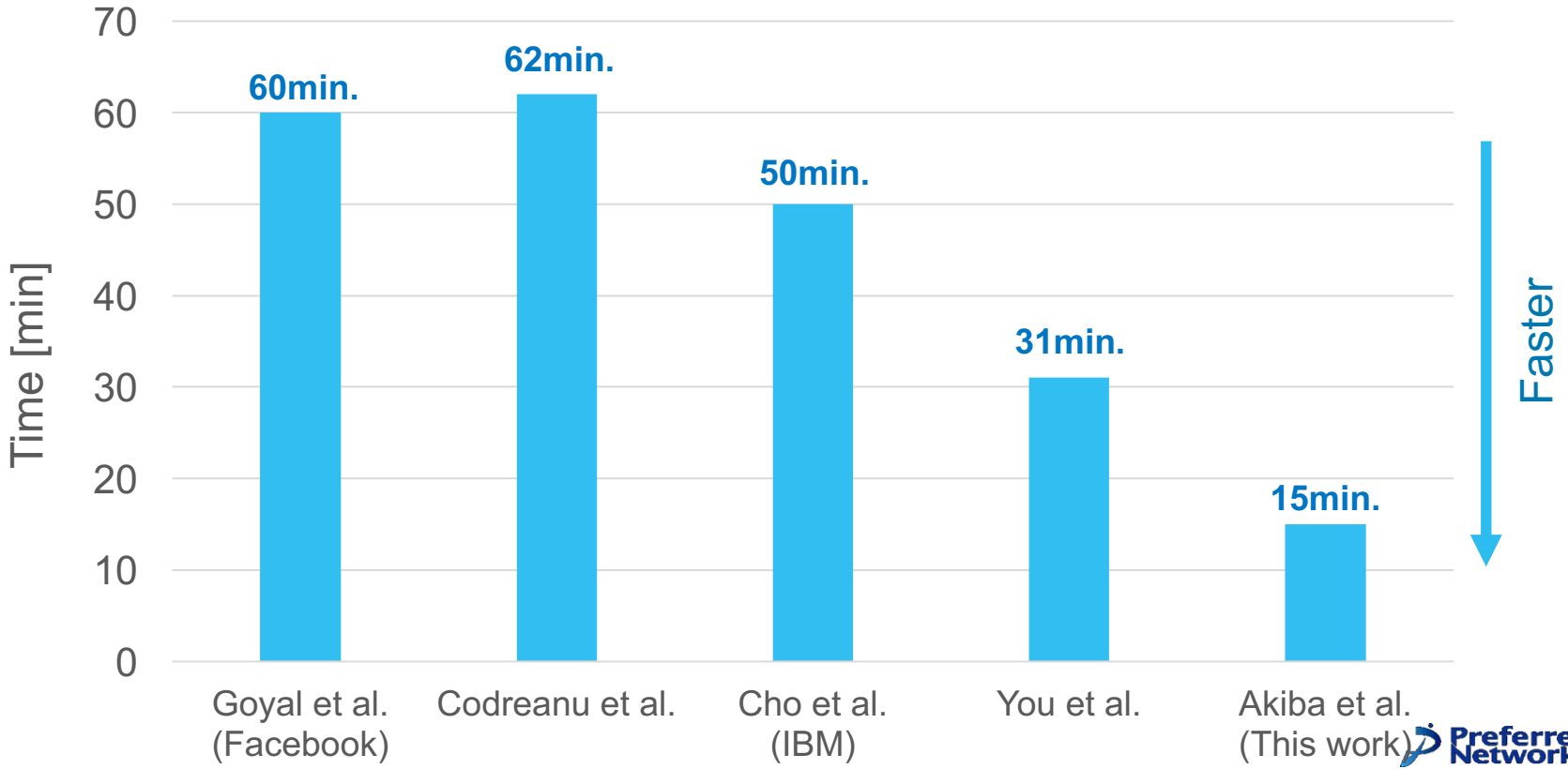
Tips for users of NCCL v2 with >1000 GPUs:

- NCCL v2 opens a large number of file descriptors.
  - `ulimit -n unlimited`, or will see `'unhandled system error'`
- NCCL v2 uses huge amount of stack.
  - `ulimit -s unlimited`, or will see SEGV
- When it suddenly starts to claim 'unhandled system error', just **reboot** all nodes.

(As of NCCL 2.0.5)

Preferred Networks

Training time of ResNet-50 (90 epochs) on ImageNet



Preferred Networks

Training ResNet-50 on ImageNet in 15 mins

Team	Hardware	Software	Batchsize	Time	Accuracy
He et al.	P100 × 8	Caffe	256	29 hr	75.3 %
Goyal et al.	P100 × 256	Caffe2	8,192	1 hr	76.3 %
Codreanu et al.	KNL 7250 × 720	Intel Caffe	11,520	62 min	75.0 %
Cho et al.	P100 × 256	Torch	8,192	50 min	
You et al.	Xeon 8160 × 1600	Intel Caffe	16,000	31 min	75.3 %
This work	**P100 × 1024**	**Chainer**	**32,768**	**15 min**	**74.9 %**

T. Akiba, et al. "Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes" (modified)

- **Dataset:** ImageNet-1k
- **Accuracy:** single-crop top-1 validation accuracy
- **Training duration:** 90 epochs (common configuration for ResNet50)

We achieved a total training time of 15 minutes while maintaining a comparable accuracy of 74.9%.

Preferred Networks

Maybe you are thinking:

We don’t have so many GPUs…
Our GPU cluster does not have Infiniband…

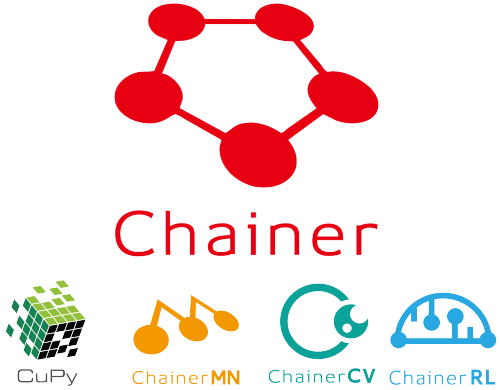
It’s not for us 🙁

Preferred Networks

ChainerMN is for you.

Preferred Networks

Want to try Chainer + ChainerMN?

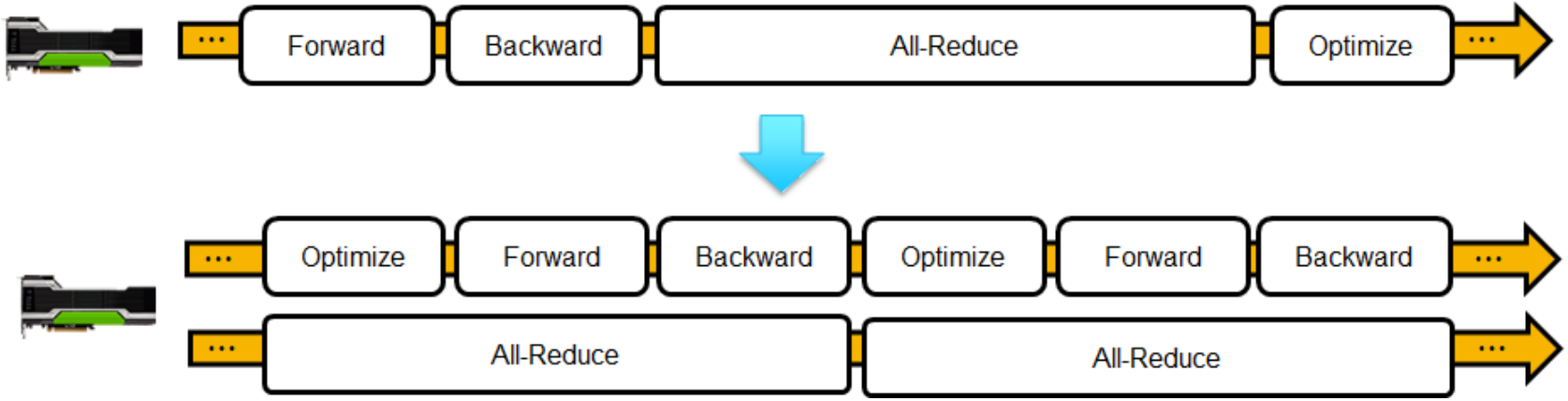


+

Cloud formation support is coming soon!

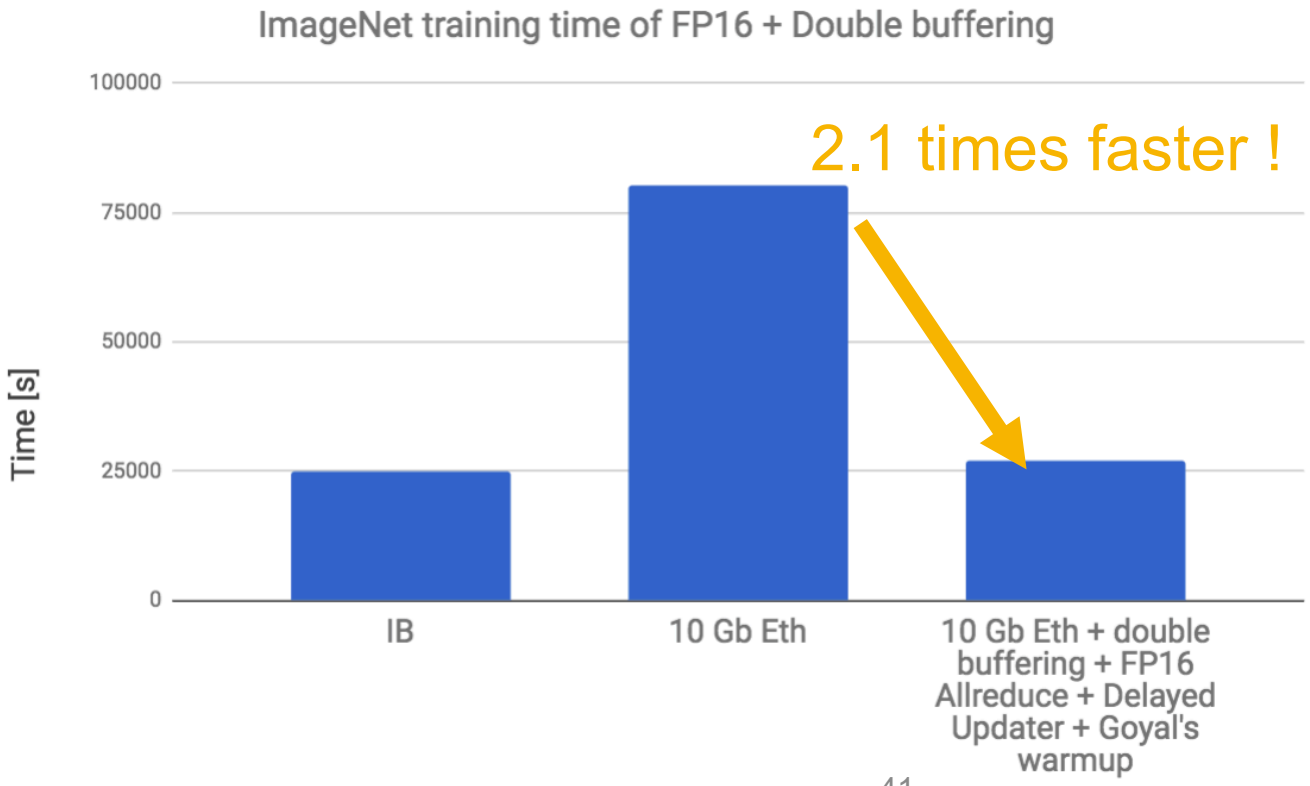
Preferred Networks

Optimization technique for non-IB environment: Double buffering



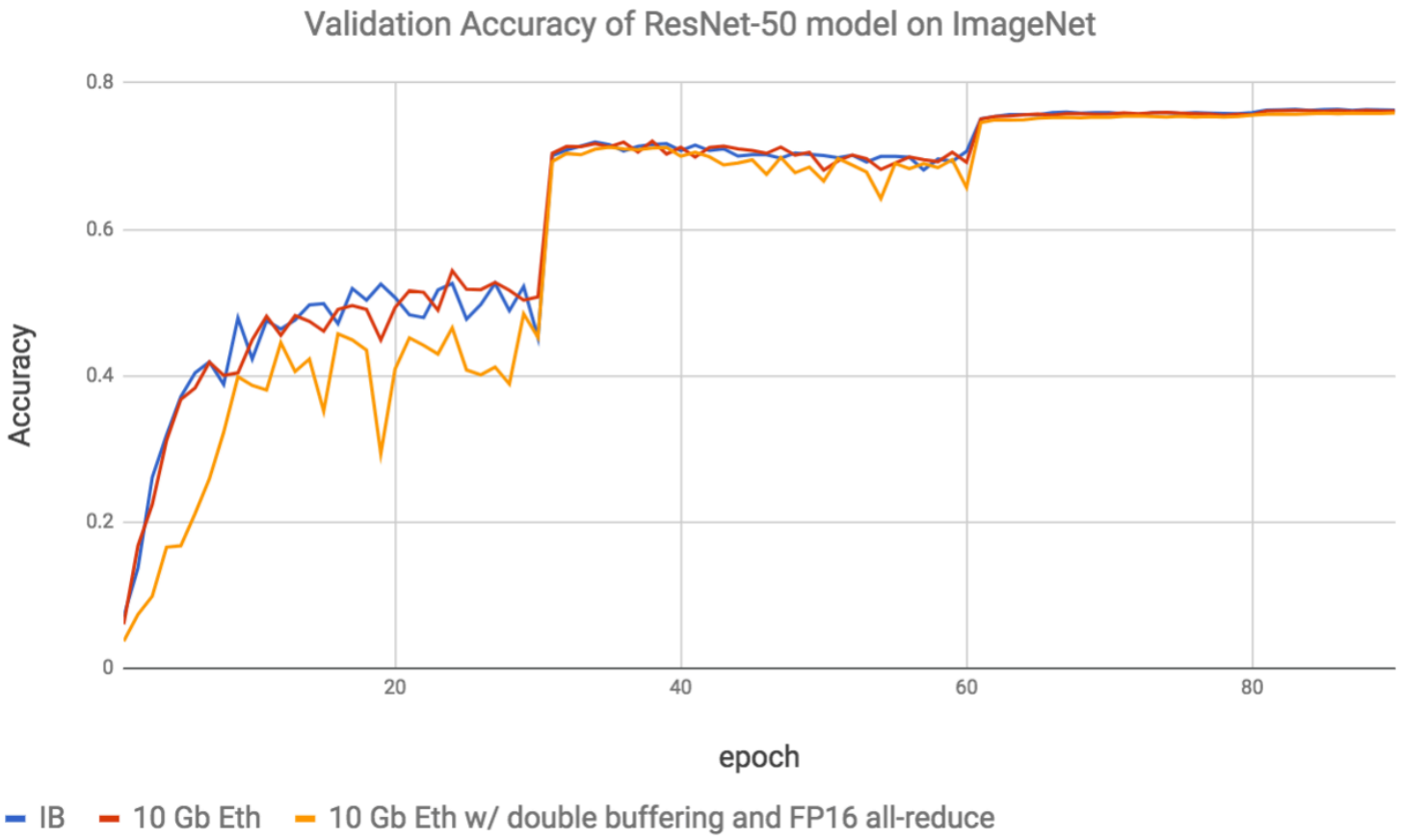
- Each update uses the gradients from previous iteration (1-step stale grad.)

Computing time of ImageNet training with Double Buffering + FP16 communication

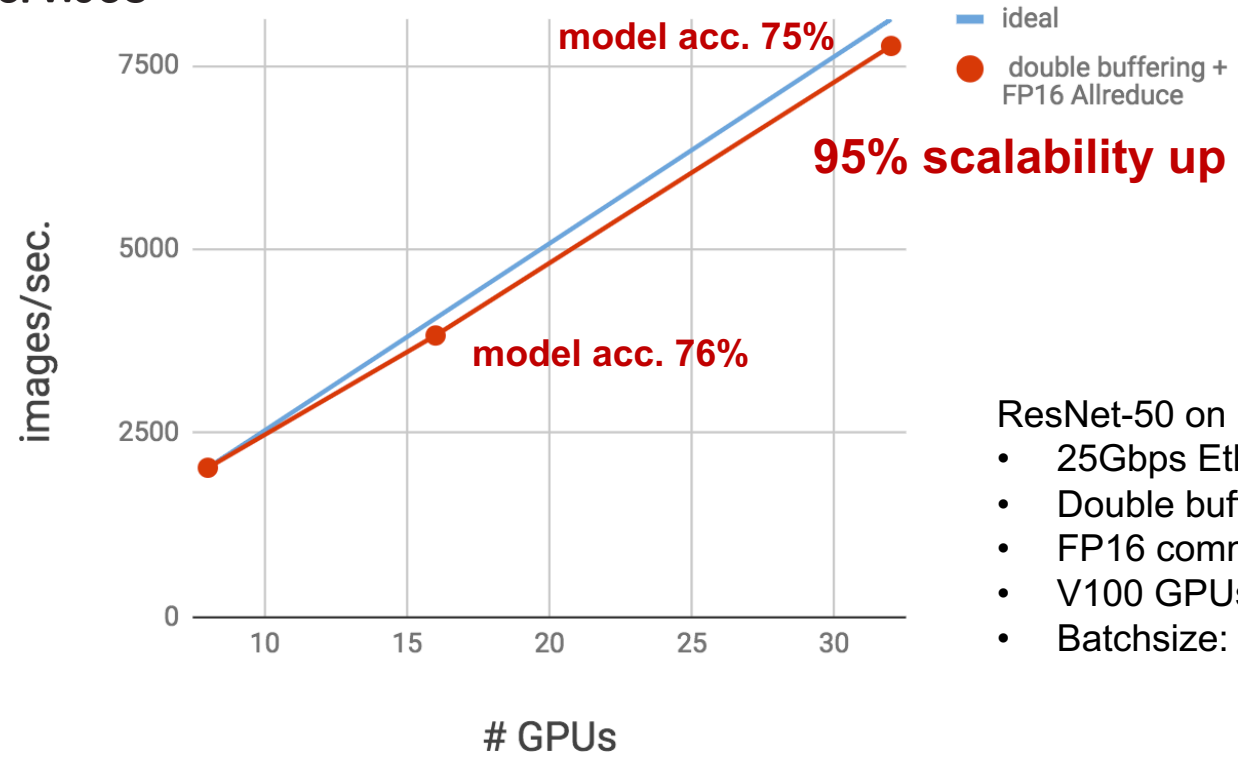


- Local batchsize: 64
- 32 processes
- NCCL for Allreduce

Preferred Networks



Preferred Networks



95% scalability up to 32 GPUs !!

ResNet-50 on ImageNet training

- 25Gbps Ethernet
- Double buffering
- FP16 communication (NCCL)
- V100 GPUs
- Batchsize: 64/GPU

Next step?

"ImageNet is the new MNIST"
by Chris Ying (Google Brain)

How to move towards larger, more complex models?

Preferred Networks

Taxonomy of distributed deep learning

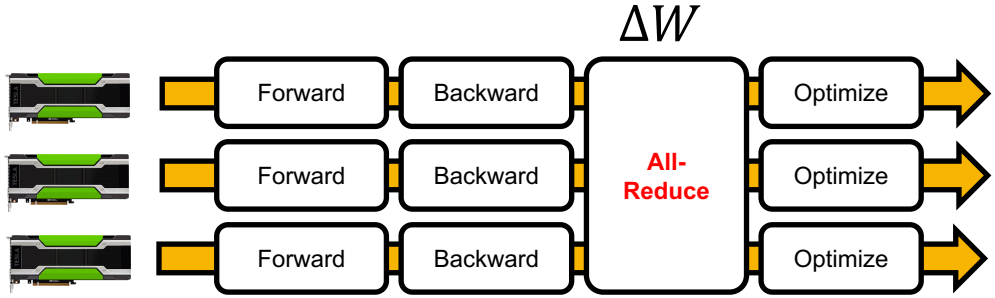
- Data-parallelism
  - Synchronous
  - Asynchronous
- Model-parallelism
  - Fine-grained
  - Coarse-grained

Main focus (currently)

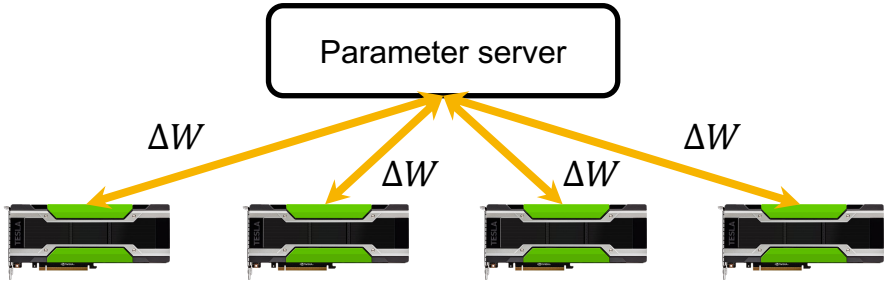
Preferred Networks

Data parallel: sync vs. async

Synchronous:

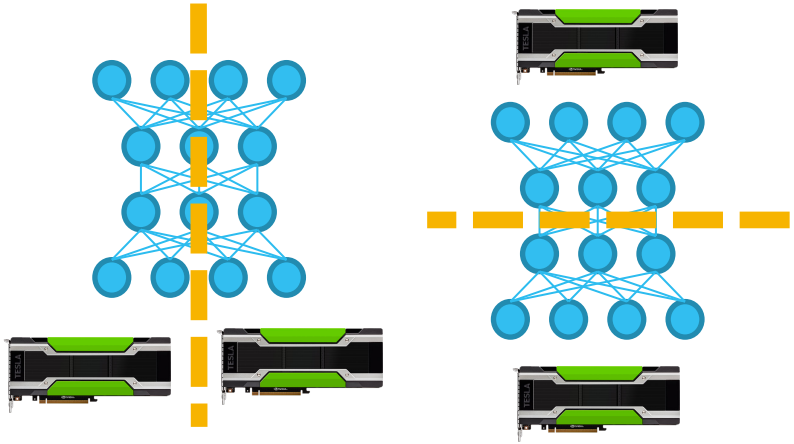


Asynchronous:

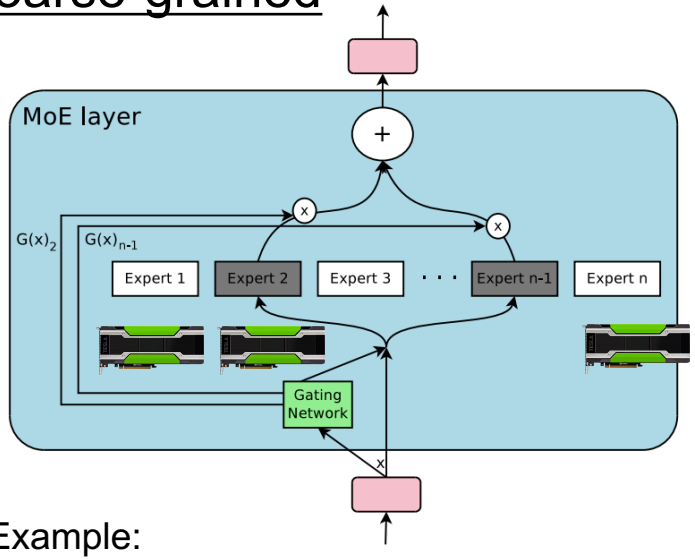


Model parallelism

Find-grained



Coarse-grained



Example:
Mixture-of-Experts [Shazeer+(Google Brain), ICLR'17]

Preferred Networks

ChainerMN's focus, now and future

Data-parallelism
- Synchronous — Main focus (currently)
- Asynchronous

Model-parallelism
- Fine-grained — Under active development
- Coarse-grained — Basic components ready to use

Preferred Networks

Conclusion

- We finished training of ResNet-50 on ImageNet in 15 min.
- We achieved both of speed, accuracy, and productivity

We will continue tackling hard problems !

Preferred Networks

Thank you!