Enabling Scalable and Efficient Deep Learning on Supercomputers

Zhao Zhang
Data Intensive Computing
Texas Advanced Computing Center

TACC

Self Introduction

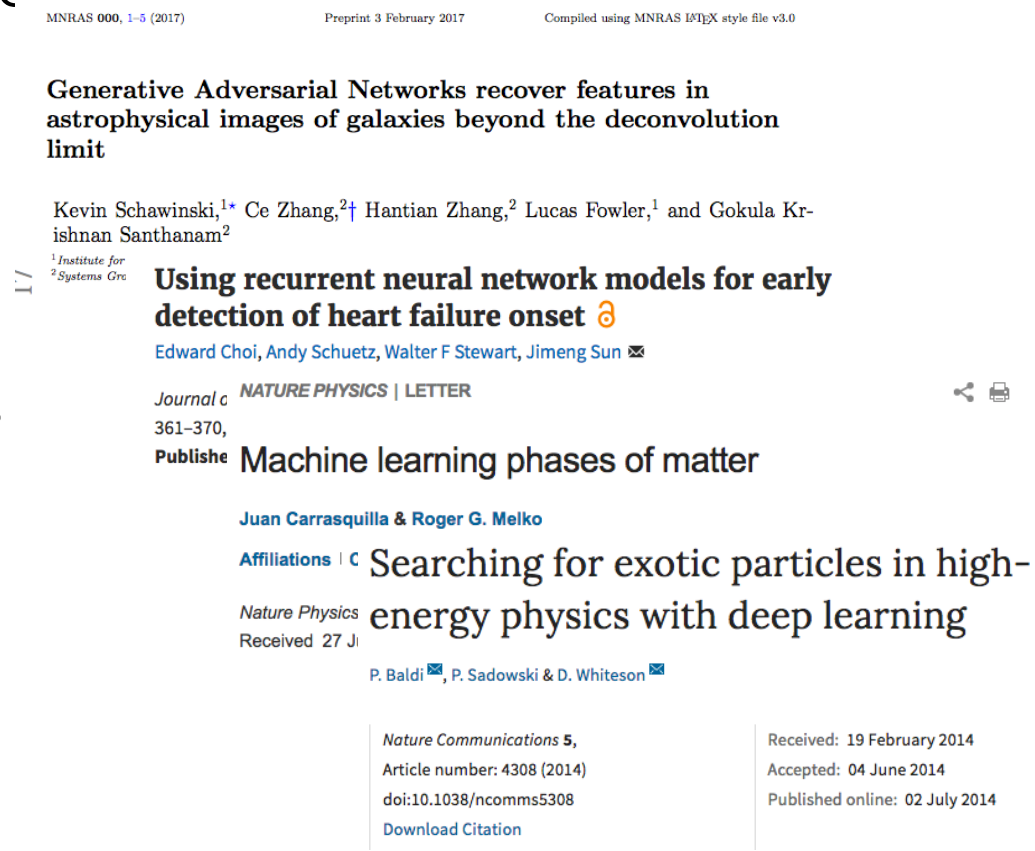
- Experience
  - Researcher in Data Intensive Computing Group at TACC, 2016 - present
  - Postdoc in AMPLab and Data Science Fellow in Berkeley Institute for Data Science, University of California, Berkeley, 2014 - 2016 (Advisor: Michael J. Franklin)
  - Ph.D in **Computer Science, University of Chicago** in 2014 (Advisor: Ian T. Foster)
- Research interest:
  - Enabling and expediting scientific discovery with computer systems and algorithms
- Current research:
  - Scalable deep learning on supercomputers
  - Machine learning interpretability
  - Memory error impact on DL training

Outline

- Trend Overview
- Motivating Applications
- Distributed Deep Learning
  - Computation
  - Communication
  - I/O
- Research
  - Scalable Training Algorithm
  - Scalable and Efficient I/O

Motivating Applications

- Many scientists are exploring and adopting deep learning as the data science methodology to tackle their domain research challenge
  - Astronomy
  - Drug discovery
  - Disease diagnosis
  - Molecular dynamics
  - Physics
  - Social science



MNRAS 000, 1–5 (2017) Preprint 3 February 2017 Compiled using MNRAS LATEX style file v3.0

Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit

Kevin Schawinski,[1]* Ce Zhang,[2]† Hantian Zhang,[2] Lucas Fowler,[1] and Gokula Krishnan Santhanam[2]

[1] Institute for
[2] Systems Gro

Using recurrent neural network models for early detection of heart failure onset

Edward Choi, Andy Schuetz, Walter F Stewart, Jimeng Sun

Journal o
361–370,
Publishe

NATURE PHYSICS | LETTER

Machine learning phases of matter

Juan Carrasquilla & Roger G. Melko

Affiliations | C

Nature Physics
Received 27 J

Searching for exotic particles in high-energy physics with deep learning

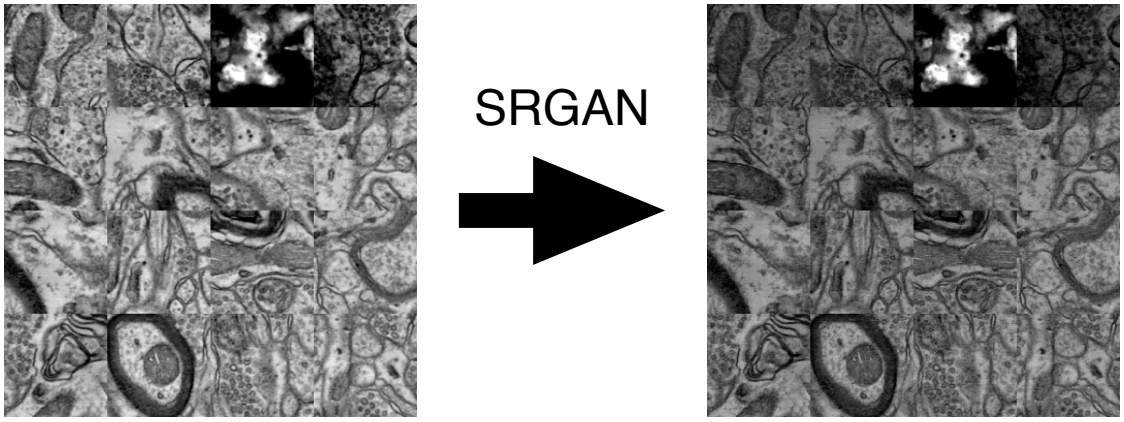
P. Baldi, P. Sadowski & D. Whiteson

Nature Communications **5**,
Article number: 4308 (2014)
doi:10.1038/ncomms5308
Download Citation

Received: 19 February 2014
Accepted: 04 June 2014
Published online: 02 July 2014

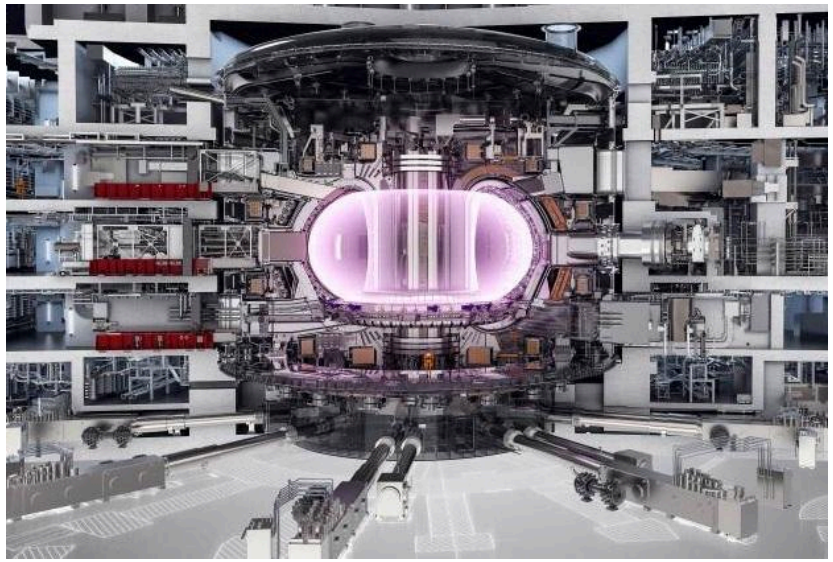
Motivating Applications

- Neural image resolution enhancement with super resolution generative adversarial network.
  - In collaboration with Salk Institute
  - ~600 GB neural image dataset
  - TensorLayer + TensorFlow + Horovod



Motivating Applications

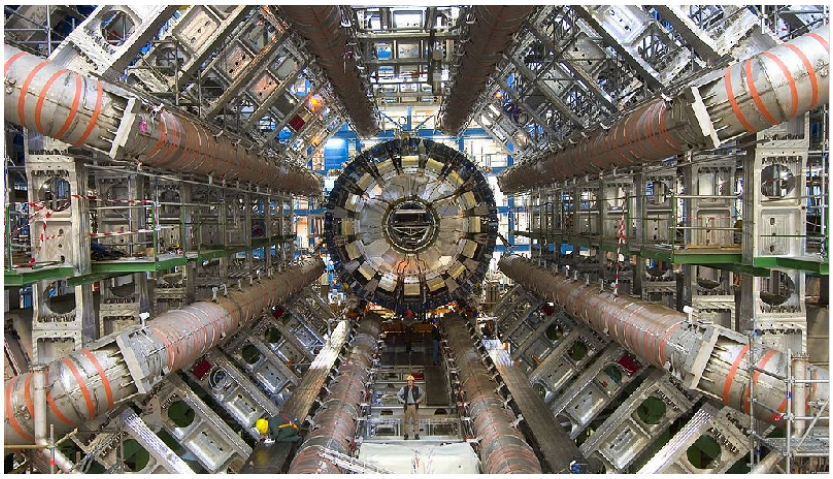
- Plasma reactor disruption prediction with fusion recurrent neural network.
  - In collaboration with ICES, UT Austin
  - ~1.7 TB text data
  - TensorFlow + MPI4Py



Courtesy image from https://www.newsweek.com/nuclear-fusion-reactor-sustainable-clean-energy-edges-closer-627995

Motivating Applications

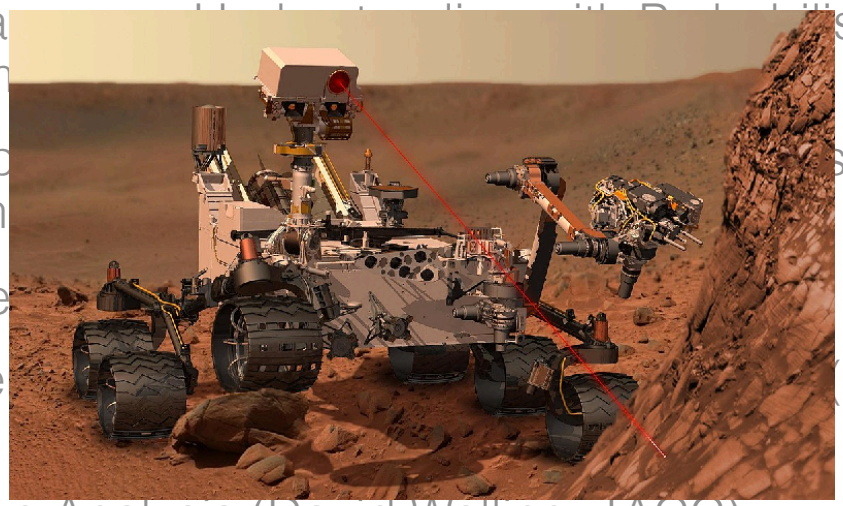
- LHC particle detector material design
  - In Collaboration with CERN and SURFSara
  - ~100 GB data in HDF5
  - Keras + TensorFlow + Horovod



Courtesy image from https://www.pbs.org/wgbh/nova/article/one-quadrillion-lhc-collisions-lead-to-a-rare-discovery/

Motivating Applications

- Power Control in Mars Rover (Chris Mattmann, NASA)
- MRI image analysis for multiple sclerosis patient management (Ponnada Narayana and et al., Texas Medical Center and UT Tyler)
- Deep Natural La stic Logic and Distributional Sin
- Functional Mapp s and the Observable States (Alex Sun
- Cancer Drug Tre Anderson)
- Hyper-paramete (Sanjay Shakkott, UT Austin)
- Geological Image Analysis (David Walling, TACC)
- Ancient Biographical Text Analysis, (David Walling, TACC)
- Austin Traffic Analysis (Weijia Xu, TACC)



Courtesy image from https://en.wikipedia.org/wiki/Mars_rover

Motivating Applications

- Distributed Deep Learning Training is
  - Computation Intensive. E.g., the classic 90-epoch ResNet-50 training with ImageNet dataset has 10^{18} single precision floating operations
  - Communication Intensive. Inter-node communication can dominate the back-propagation phase
  - I/O Intensive. Training datasets can have 10^{6} small files (KB - MB) and several TBs in total.

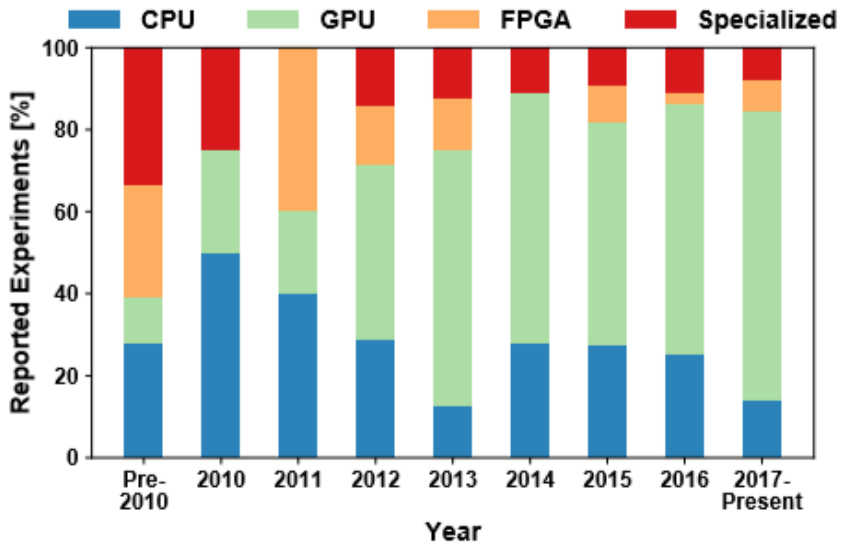
Motivating Applications

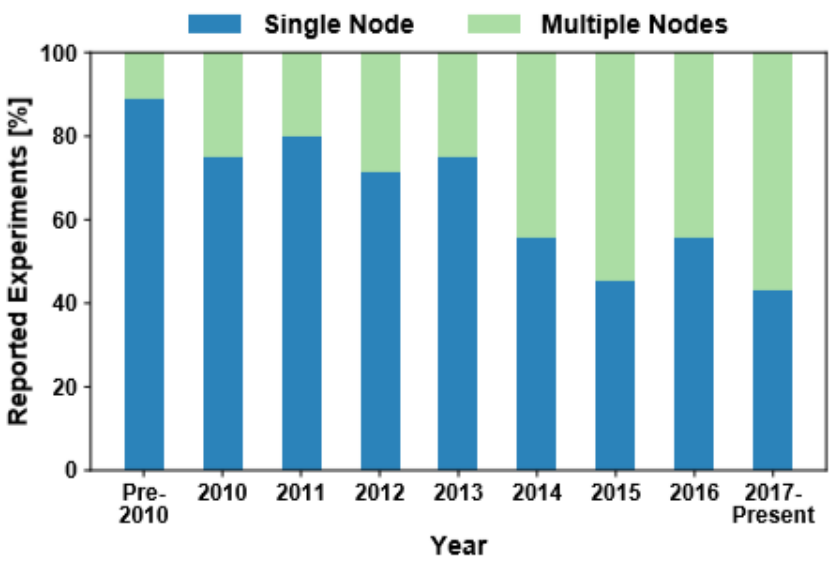
- Model Serving can be
  - Time critical, depending on use cases, e.g., autonomous driving and telescope survey
  - Highly concurrent, e.g., a surrogate model used in simulation
  - Far from data center, e.g., censors deployed in ocean, IoT, smart surveillance cameras

Deep Learning is Supercomputing

Trends in deep learning: hardware and multi-node

The field is moving fast – trying everything imaginable – survey results from 227 papers in the area of parallel deep learning





Deep Learning is largely on distributed memory today!

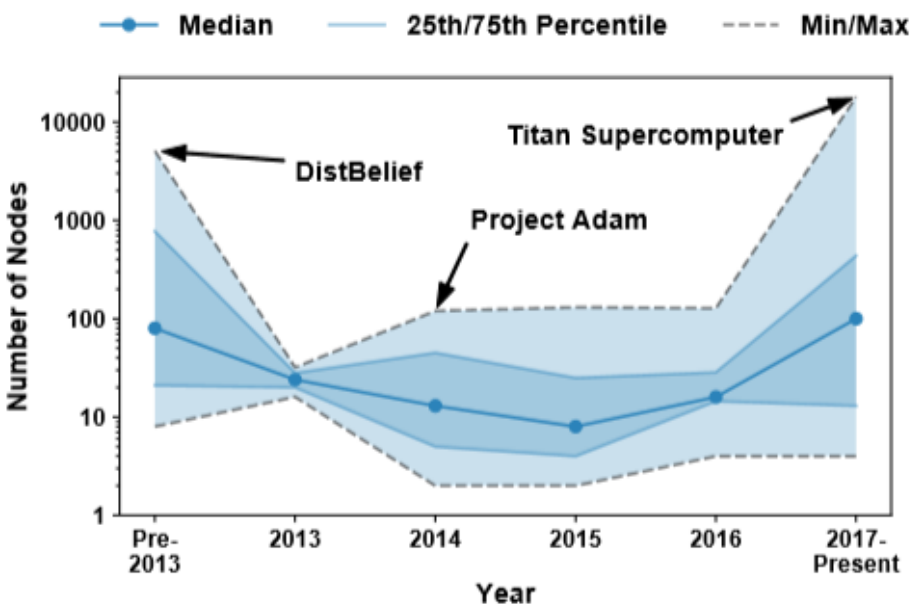
T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

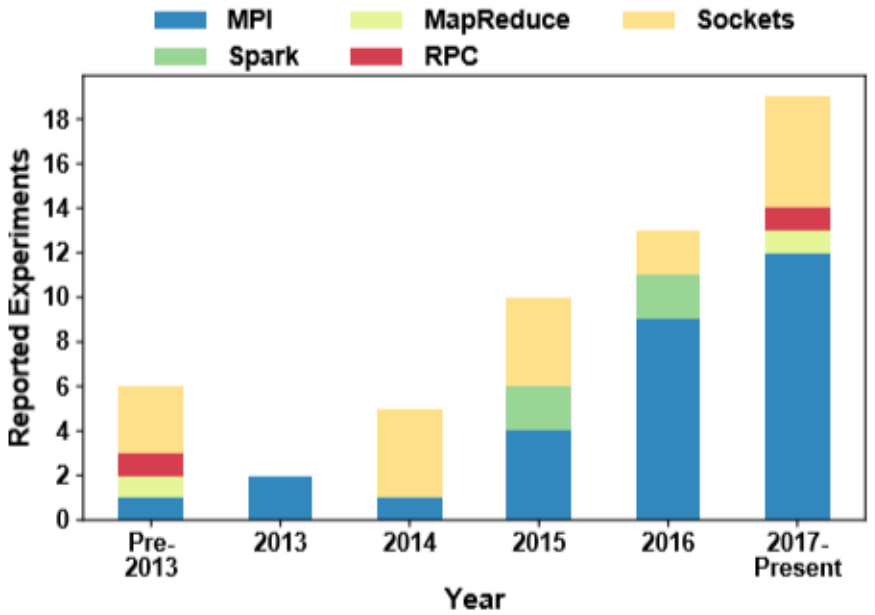
Courtesy Slide from https://htor.inf.ethz.ch/publications/img/ddl-slides.pdf

Deep Learning is Supercomputing

Trends in distributed deep learning: node count and communication

The field is moving fast – trying everything imaginable – survey results from 227 papers in the area of parallel deep learning





Deep Learning research is converging to MPI!

T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

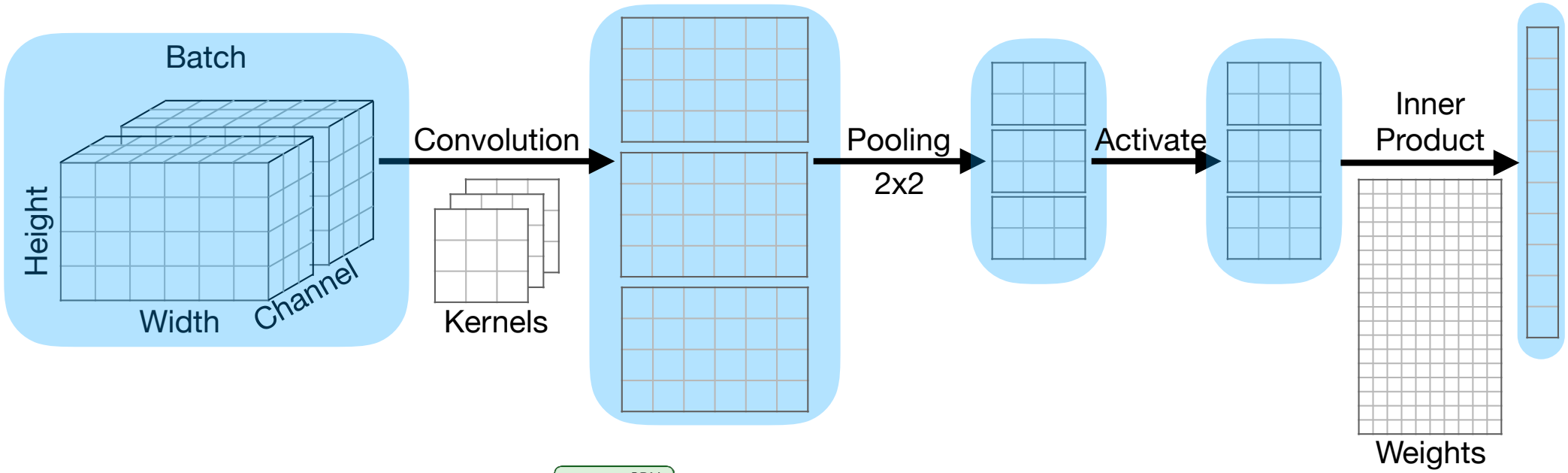
Courtesy Slide from https://htor.inf.ethz.ch/publications/img/ddl-slides.pdf

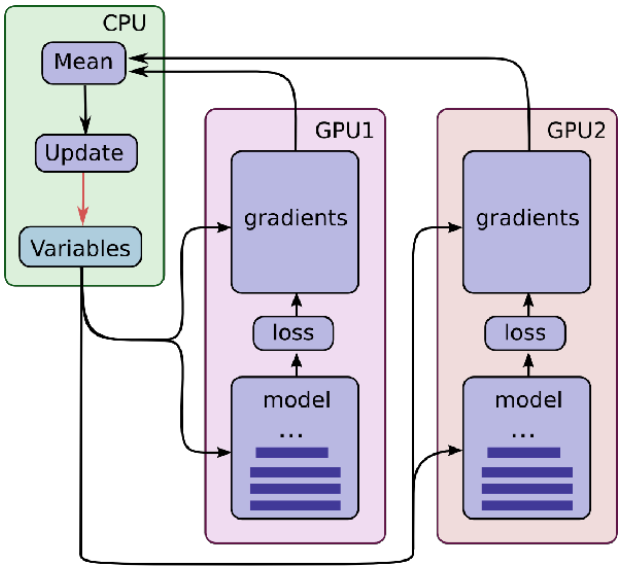
Distributed Deep Learning

- Performance is critical for hyper-parameter tuning
  - *Minutes, Hours*
    - *— Interactive research! Instant gratification!*
  - *1-4 days*
    - *Tolerable*
    - *Interactivity replaced by running many experiments in parallel*
  - *1-4 weeks*
    - *High value experiments only*
    - *Progress stalls*
  - *>1 month*
    - *Don't even try*

— Jonathan Hseu, Google Brain Team

Intra-node Parallelism

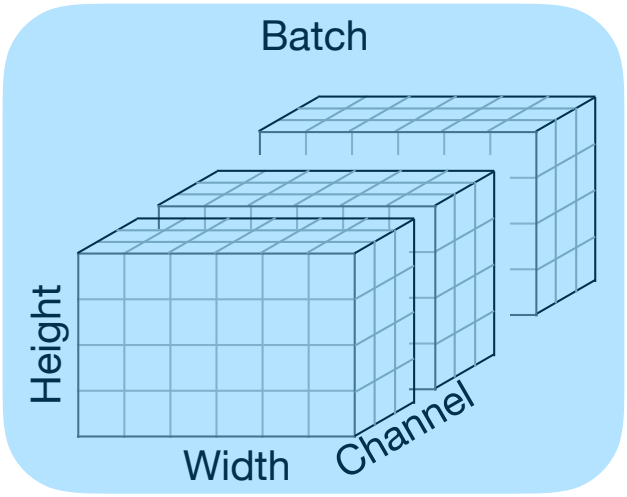


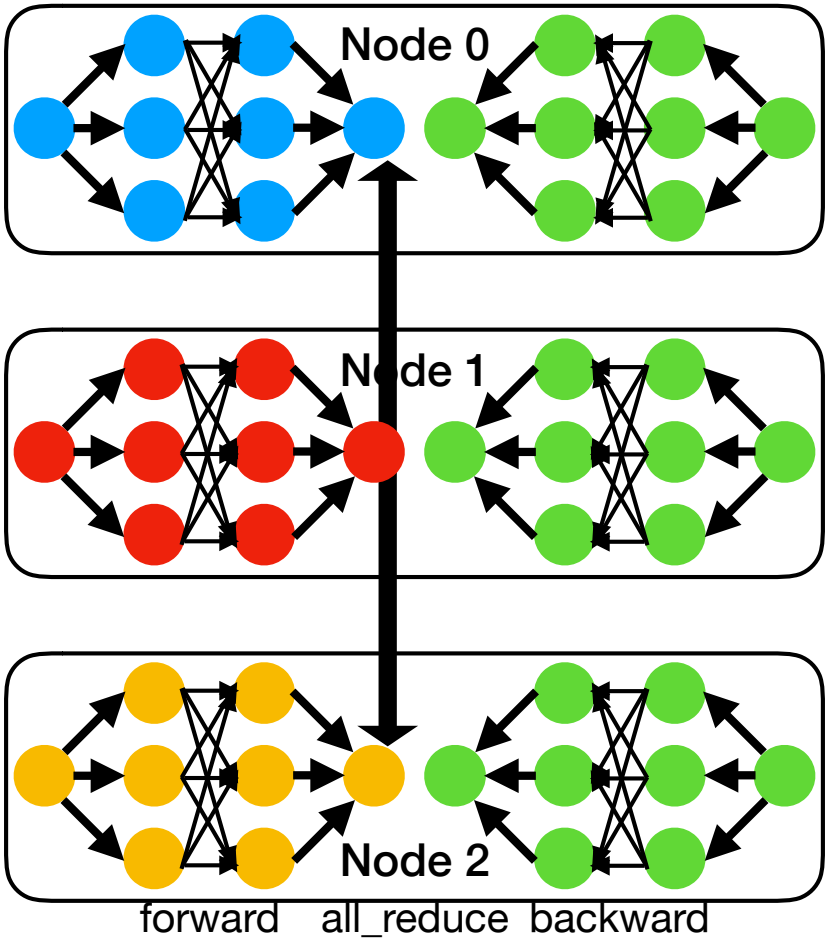


Courtesy image from https://www.tensorflow.org/tutorials/images/deep_cnn

Inter-node Parallelism

- Data Parallel

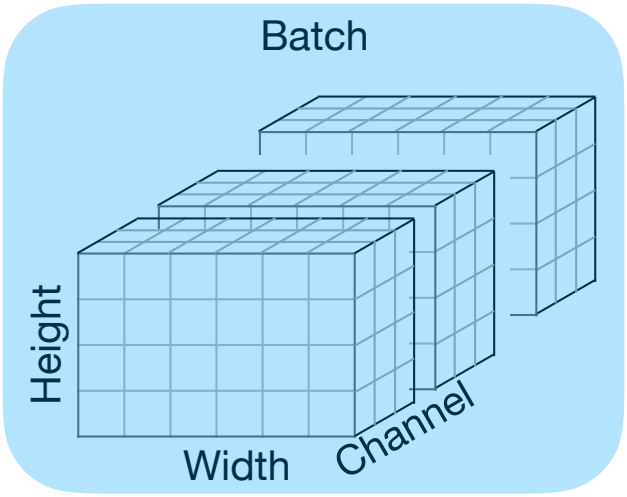


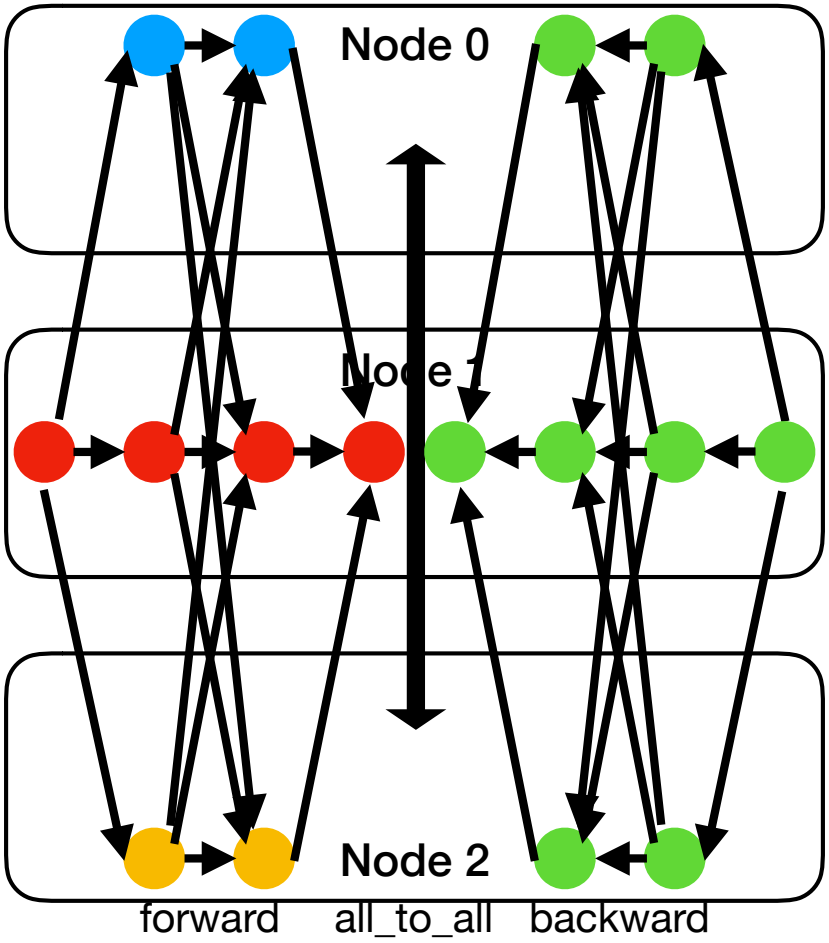


- **Parameters (Model) are duplicated on all nodes**

Inter-node Parallelism

- Model Parallel

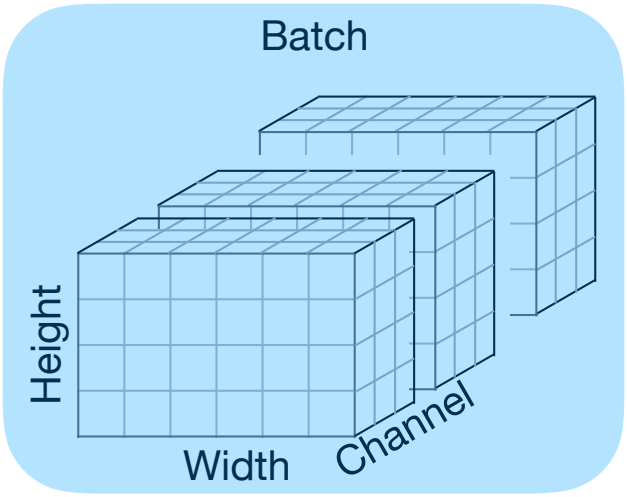


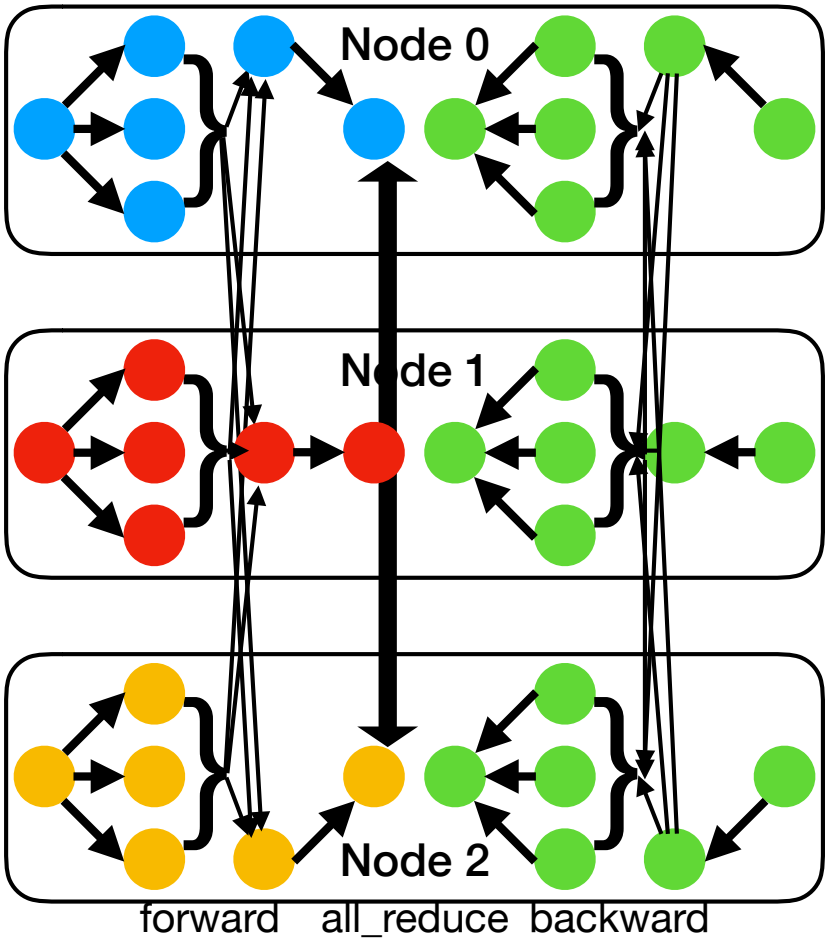


- **Weights are distributed to all nodes**
- **Back-propagation requires all-to-all communication for every layer**

Inter-node Parallelism

- Hybrid Parallel





- **Often specific to layer-types**

Distributed Deep Learning

- Data parallel is widely adopted, it is supported in almost all distributed DL frameworks
- Model parallel is suited for DL with large model sizes, e.g., RNN

Parameter Update Consistency

- Synchronous
  - All parameters in the model are updated every iteration
  - Better convergence performance
- Asynchronous
  - Not all parameters are updated every iteration
  - Does not guarantee to converge

Scalable Training Algorithm

- Problem Statement
  - To yield high utilization at scale, we need to feed enough data (computation), which results in large batch size
  - Validation (test) accuracy is sensitive to batch size.
  - Large batch size results in degraded validation accuracy

Scalable Training Algorithm

- Previous work
  - Accurate, large minibatch SGD: training imagenet in 1 hour
  - warmup
  - learning scaling

Scalable Training Algorithm

- Layer-wise Adaptive Rate Scaling (LARS)
  - Intuition: learning rate should be adjusted according to the norm of the weights in each layer

SGD with LARS

$$w = w - [\mu a + l\lambda \frac{||w||_2}{||\nabla w||_2 + \beta||w||_2}(\nabla w + \beta w)]$$

$$a$$

momentum
0.9

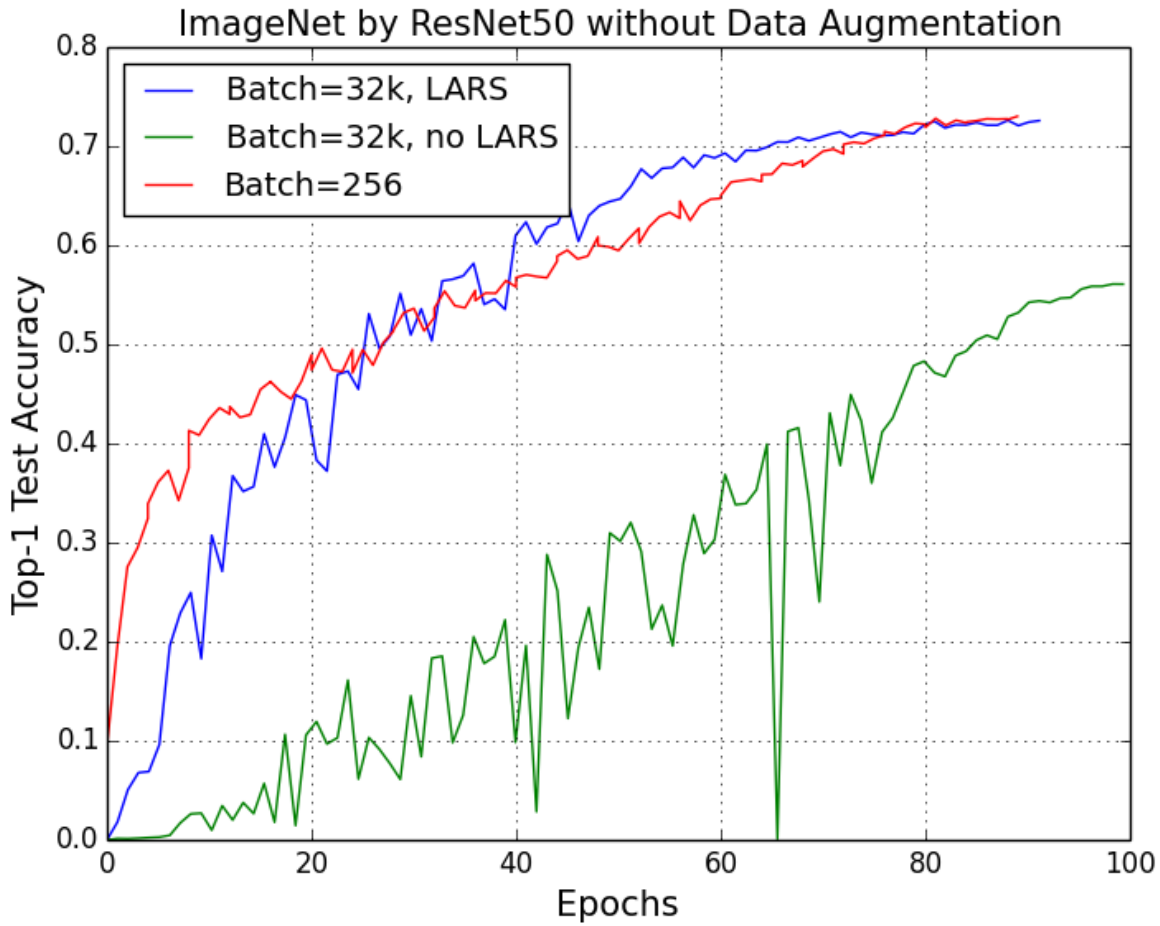
scaling factor
0.001

learning rate

weight decay:
0.0005

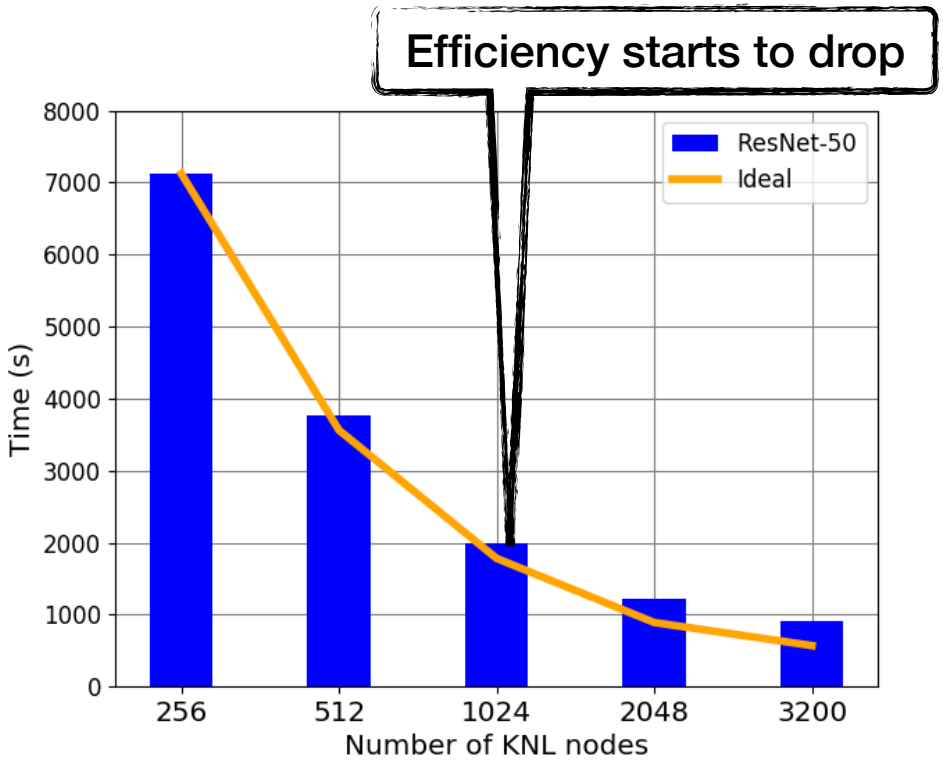
Scalable Training Algorithm

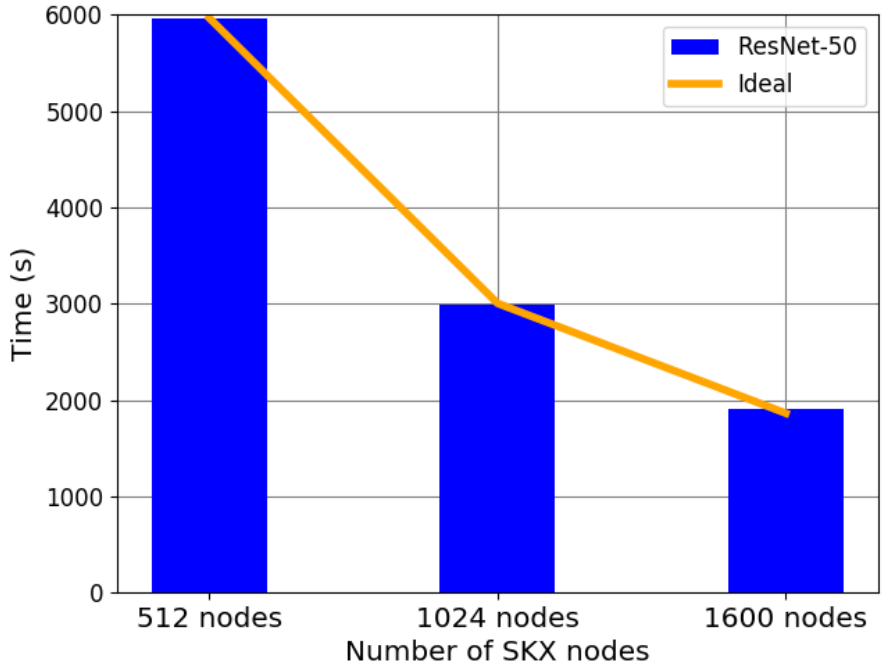
- Using batch size of 32K while preserving validation accuracy



Scalable Training Algorithm

- Using batch size of 32K on Intel Xeon Phi 7250 (KNL) and Intel Xeon Platinum 8160 (SKX) nodes

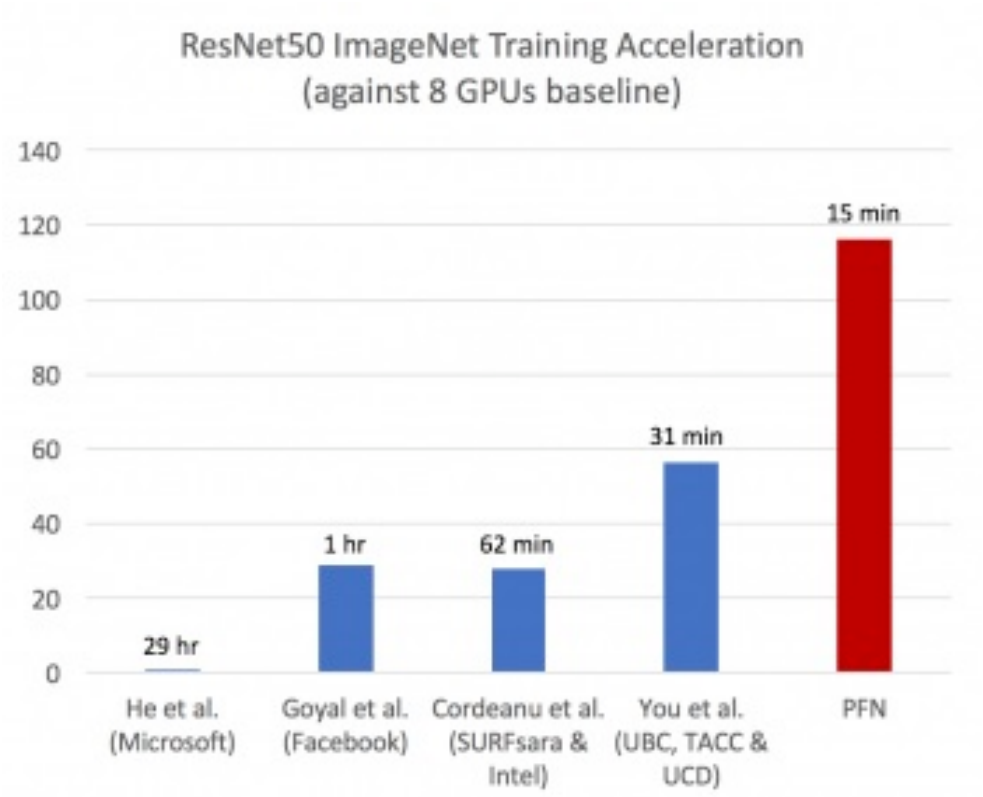




You, Yang, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. "ImageNet training in minutes." In Proceedings of the 47th International Conference on Parallel Processing, p. 1. ACM, 2018. Best paper

Scalable Training Algorithm

- Result
  - 90-epoch ResNet-50 training finished in 20 mins on 2,048 KNL with 74.9% top-1 accuracy



Scalable Training Algorithm

- Follow-on Work
  - 6.6 minutes using 2,048 Tesla P40 GPUs from researchers in Tencent and Hong Kong Baptist University
    - Half-precision for forward computation and back propagation, single precision for LARS
  - 224 seconds using 2,176 Tesla V100 GPUs from Sony researchers
    - 2D-Torus optimization with LARS
  - 2.2 minutes with 1024 chip Google TPUv3 Pod

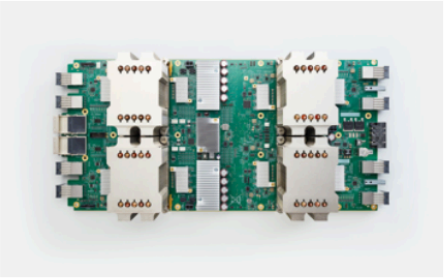


Figure 1: Cloud TPU v2 device with four chips, **180 teraFLOPS** of peak floating point throughput and **64 GB** of High Bandwidth Memory (HBM).

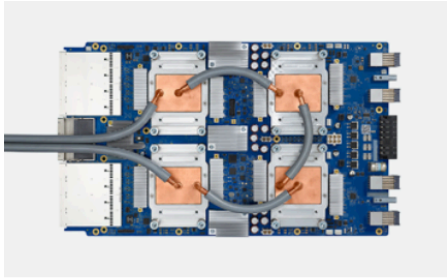


Figure 2: Liquid-cooled Cloud TPU v3 device with four chips, **420 teraFLOPS** of peak floating point throughput and **128 GB** of HBM.

State-of-the-art SGD Techniques

- Layer-wise Adaptive Rate Scaling algorithm
  - Intuition: learning rate should be adjusted according to the norm of the weights in each layer
- Adaptive Batch Size
  - Intuition: decaying learning rate schedules can be directly converted into increasing batch size schedules
- Mixed Precision
  - Intuition: relaxing precision to increase arithmetic throughput
- Deep Gradient Compression
  - Intuition: synchronizing only the largest 0.1% weights every iteration

Questions

Deep Learning I/O

- Problem Statement
  - DL's long lasting, repeated, high volume, and highly concurrent file access can easily saturate the metadata and data service of traditional shared file system.
  - ResNet-50 with Keras, TensorFlow, and Horovod on 16 nodes, each with 4 GPUs
    - 128K stat() and readdir() operations with 64-way concurrent access
    - 117M stat(), open(), close() operations with 256-way concurrent access
    - ~180M read() operations with same concurrency
    - ~ 8 hour duration

Deep Learning I/O

- Example Datasets

Dataset	# files	# dirs	total_size	file_size
ImageNet	1.3 million	2002	140 GB	KB-MB
Neural Image	0.6 million	6	500 GB	MB
Reactor Status	0.17 million	1	65 GB	KB

Deep Learning I/O

- Many scientific applications have datasets in POSIX files and directories

- Directory metadata is accessed before training starts
- Directory metadata is accessed by many processes concurrently

- File metadata is accessed before each iterations
- File metadata is accessed by many processes concurrently

Deep Learning I/O

- File data can be accessed asynchronously with the current iteration, e.g., prefetch in Caffe and data generator in Keras
- When a file is read, it is read completely and sequentially
- Files are accessed randomly
  - If the dataset can not be cached completely, all caching algorithms may behave as a random implementation
- Files can be accessed concurrently by many processes

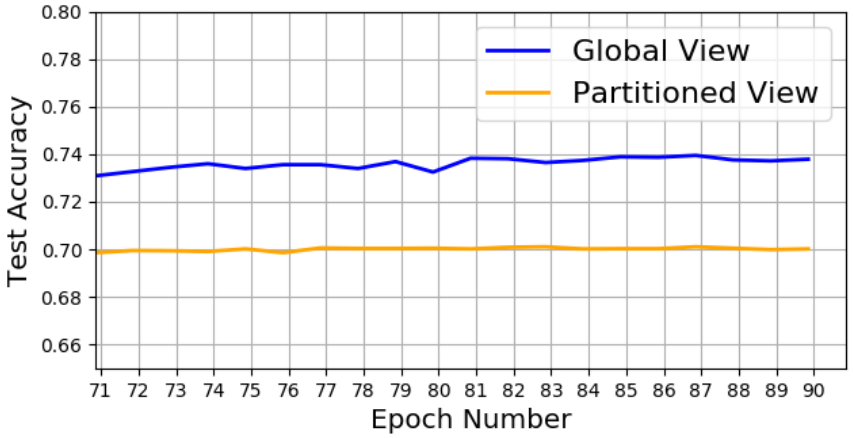
Deep Learning I/O

- Checkpoints are written from the process with Rank 0
- Checkpoints are usually named with epoch number, as previous models can have better generality performance than later ones
- GANs may write sample outputs for human examination

- Output files are not accessed by training programs unless resuming from a checkpoint
- Output files can be needed during training when debugging
- Output files can be needed only by the end of training

Deep Learning I/O

- `



Deep Learning I/O

- Sometimes, we are limited by the local storage size
  - Hosting a 455GB neural image dataset requires 8 nodes on a cluster with each node having 60GB storage space

Deep Learning I/O

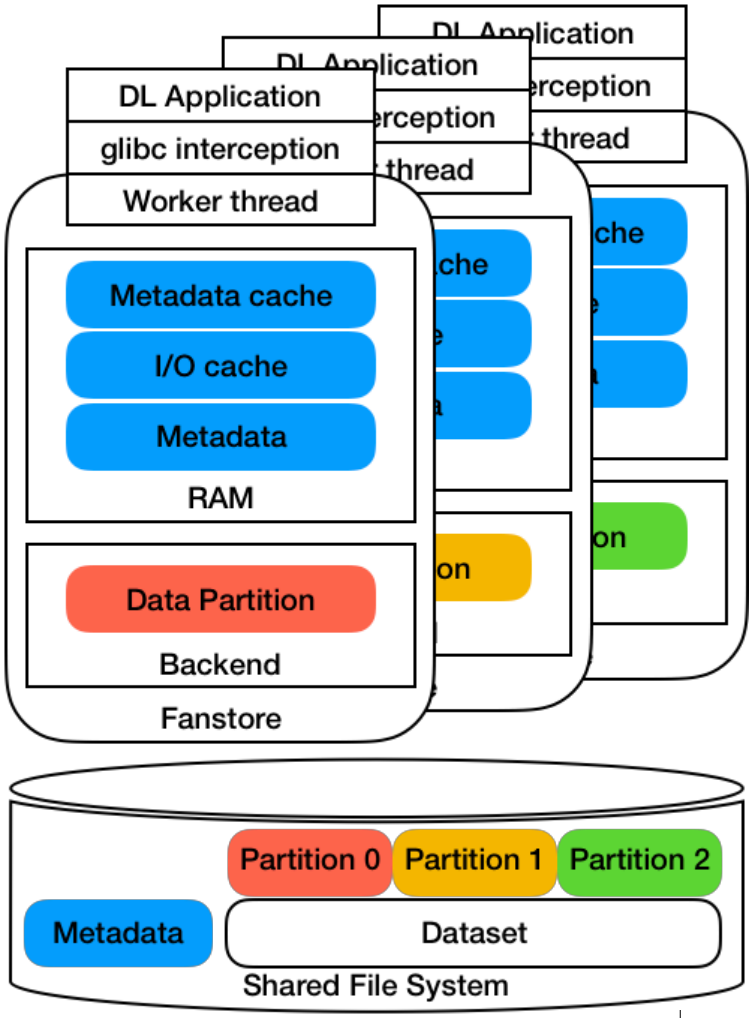
- Previous work to enable scalable I/O for HPC and Workflow applications
  - Distributed metadata server design
    - Giga+, ZHT, GlusterFS
  - File pre-creation
    - PVFS
  - I/O system design
    - ZOID, CIOD
  - Limiting write() semantic
    - HDFS

FanStore Design

- Leverages the local storage and interconnect to reduce the I/O traffic between compute nodes and shared file system
- Place the metadata and file data across nodes to enable high performance read and write
- Preserve a POSIX-compliant interface for user's convenience
- Enhance the storage capacity of existing hardware

FanStore Design

- FanStore is a transient runtime file system that optimizes I/O for distributed DL training.
- Data is partitioned (optionally compressed) and spread across local storage space
- File access functions are intercepted and handled in user space
- Remote file access is in the form of MPI round-trip message

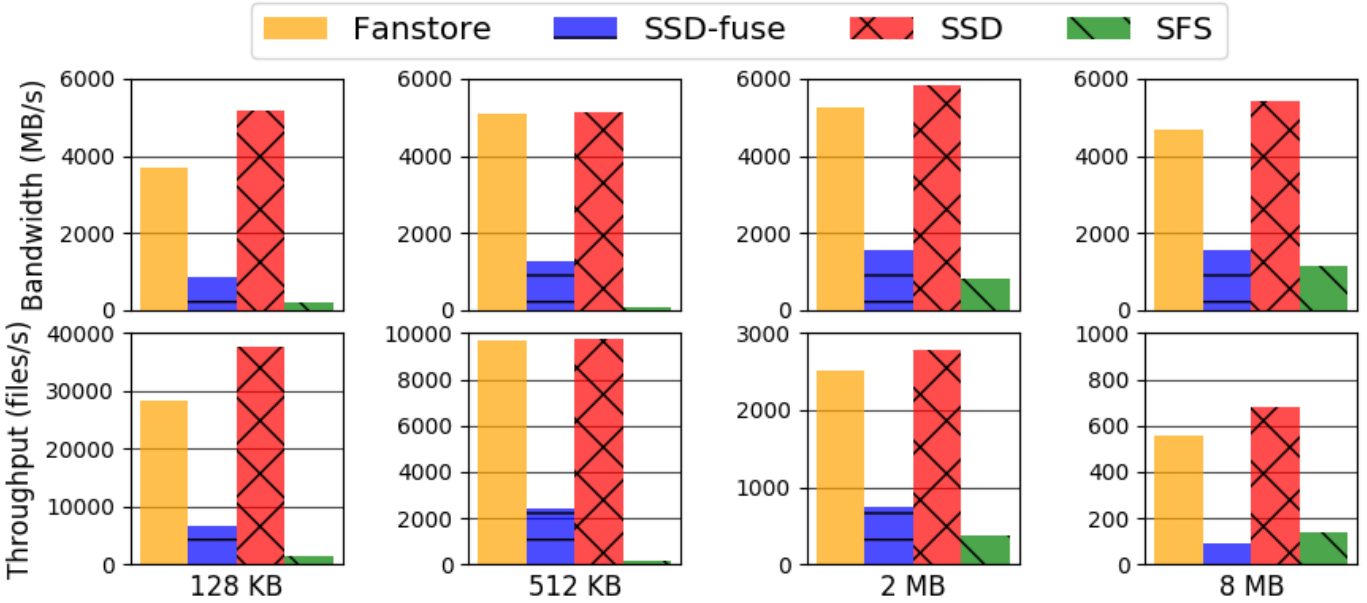


Platforms

- GPU cluster
  - 24 nodes, each with 4 Nvidia GTX 1080 Ti GPUs
  - Infiniband FDR
  - 60 GB local SSD
- CPU cluster
  - 1,736 Intel Xeon Platinum 8160 Dual Sockets nodes
  - Omnipath
  - 144 GB local SSD

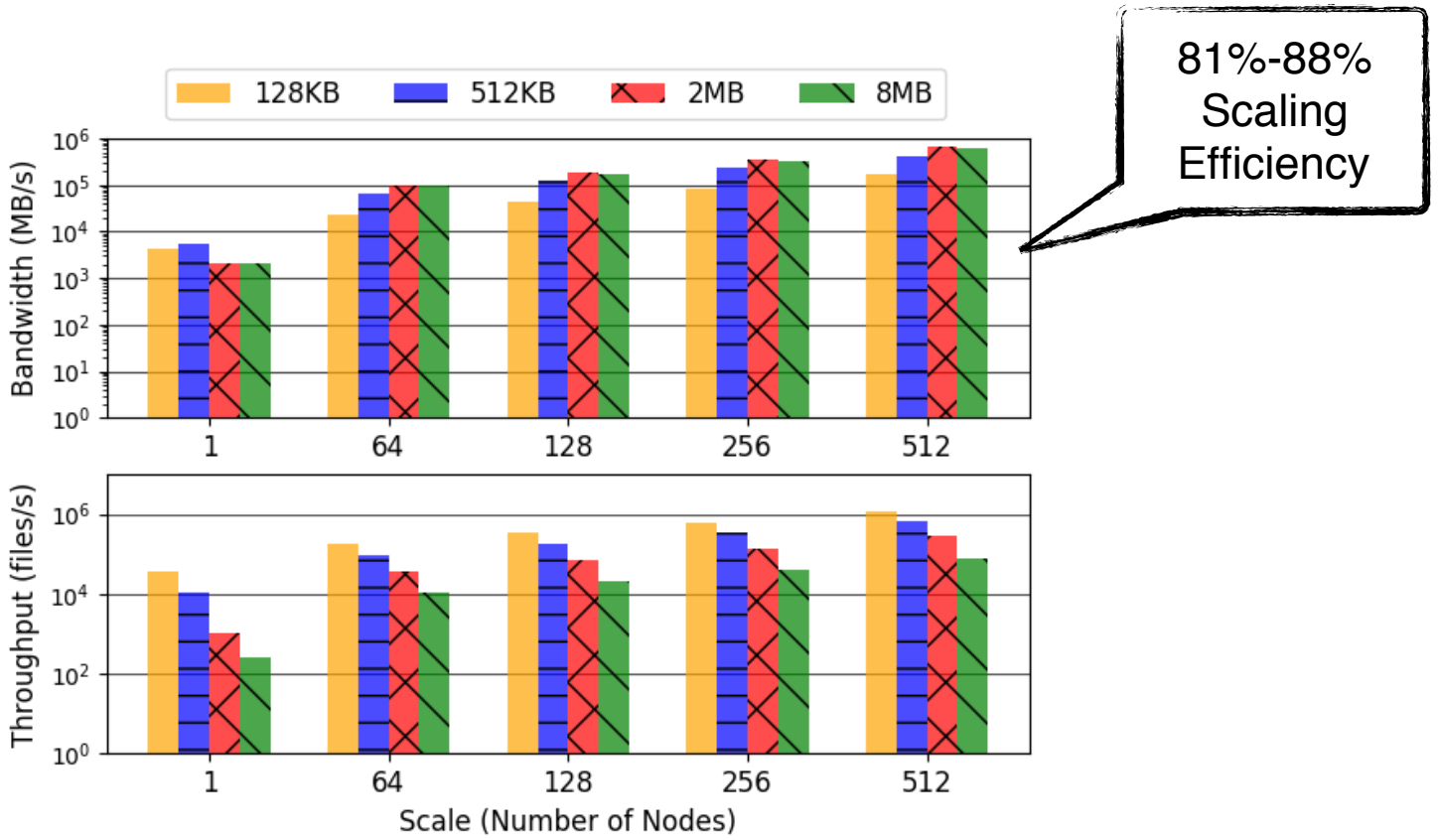
Benchmark Results

- {128KB, 512KB, 2MB, 8MB} file size on both the GPU and CPU cluster
- Goal: to verify the efficiency of FanStore design and implementation on a single node

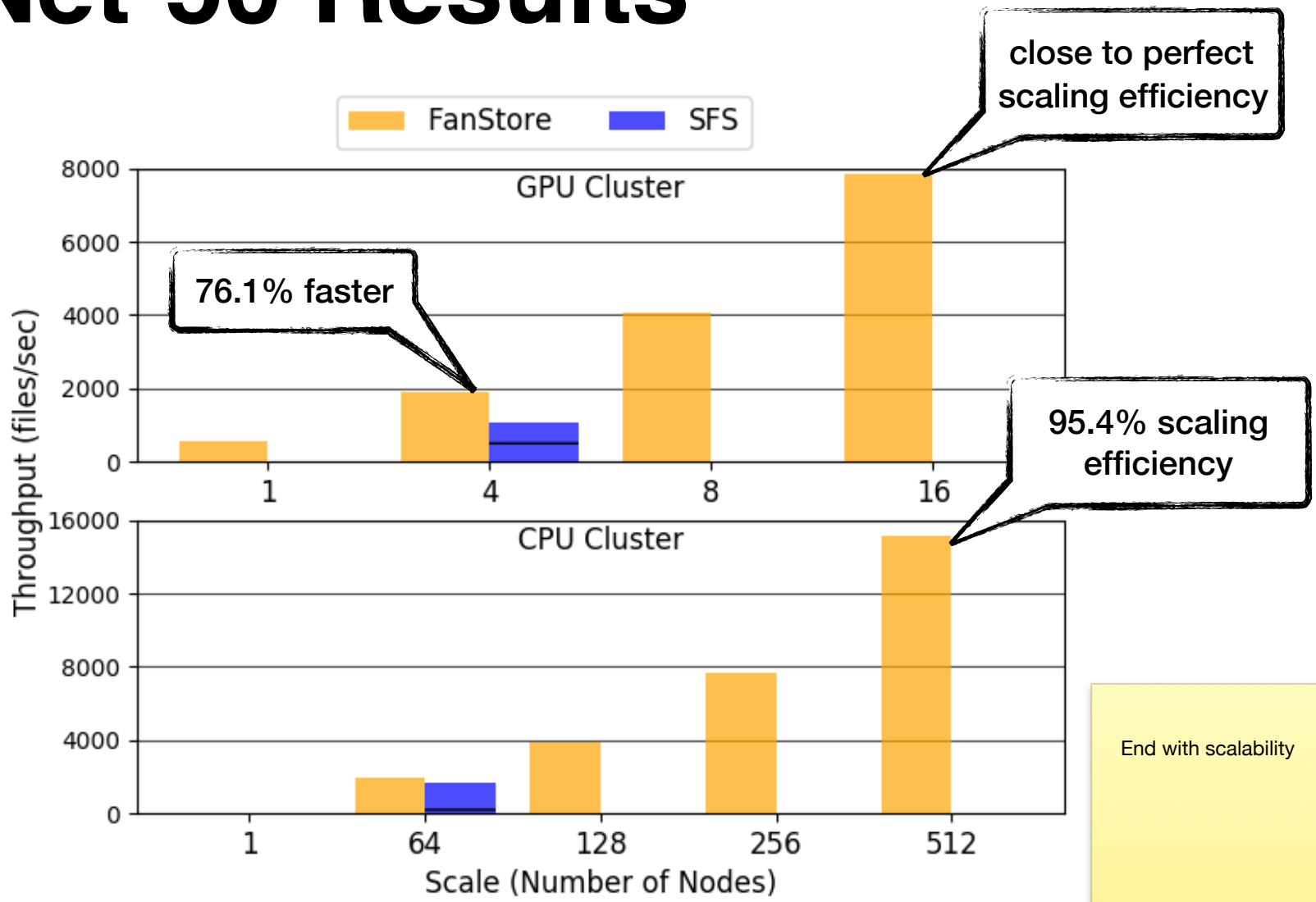


Benchmark Results

- {128KB, 512KB, 2MB, 8MB} file size on both the CPU cluster
- Goal: to verify the scalability of FanStore design and implementation



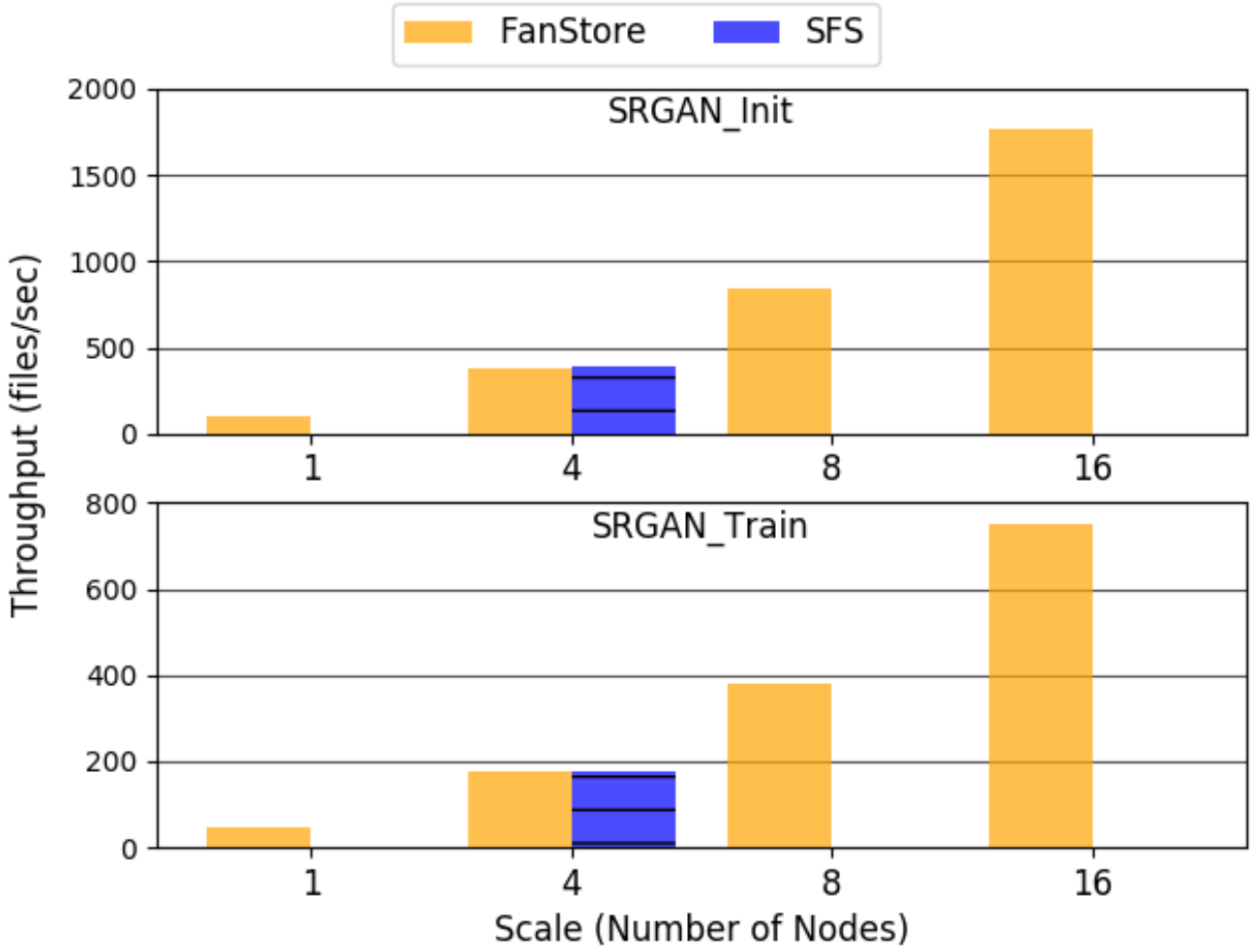
ResNet-50 Results



End with scalability

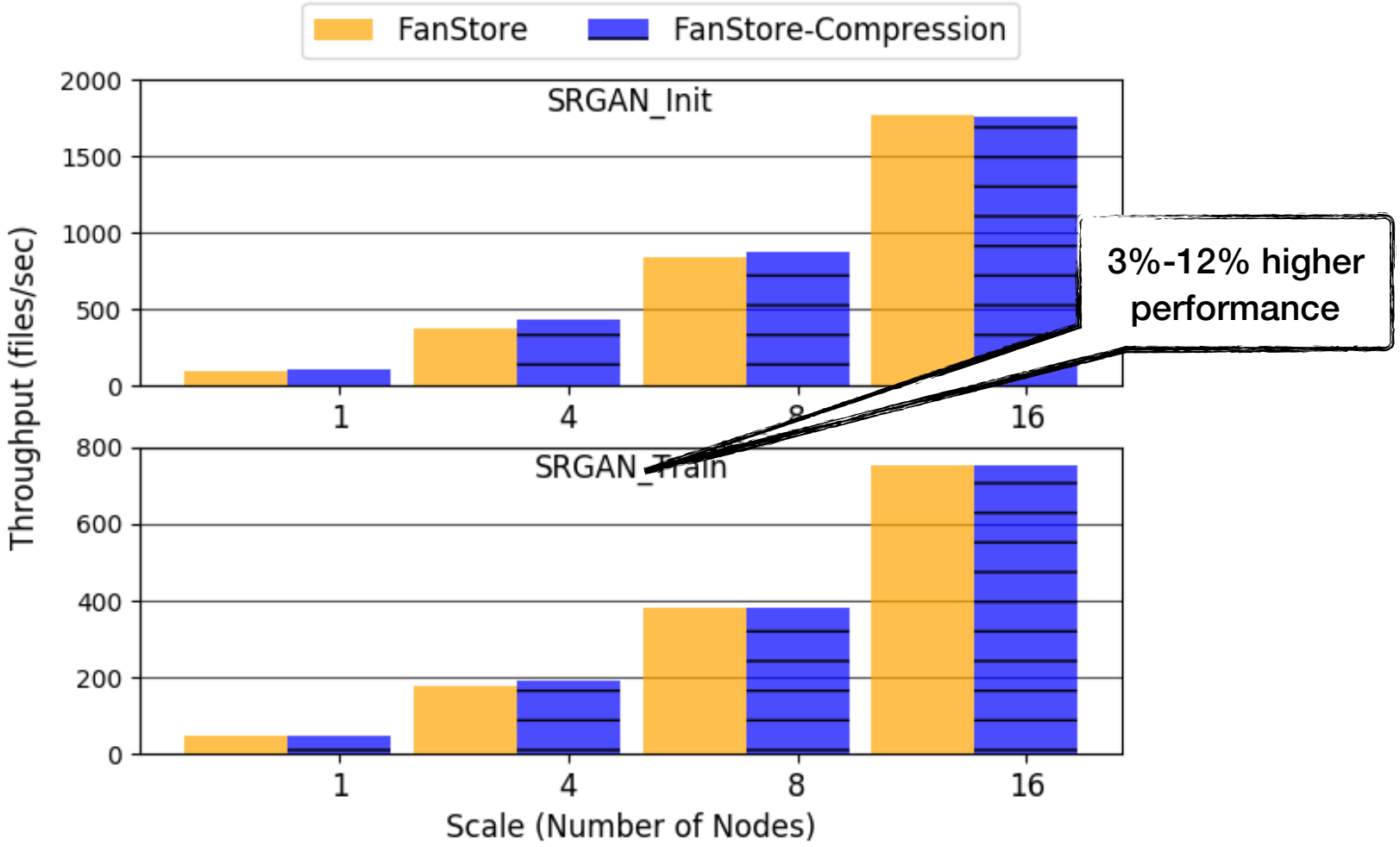
Zhao Zhang, Lei Huang, Uri Manor, Lingjing Fang, Gabriele Merlo, Craig Michoski, John Cazes, Niall Gaffney.
"FanStore: Enabling Scalable and Efficient I/O for Distributed Deep Learning".
Preprint: https://arxiv.org/abs/1809.10799

SRGAN Results



Zhao Zhang, Lei Huang, Uri Manor, Lingjing Fang, Gabriele Merlo, Craig Michoski, John Cazes, Niall Gaffney.
"FanStore: Enabling Scalable and Efficient I/O for Distributed Deep Learning".
Preprint: https://arxiv.org/abs/1809.10799

SRGAN Results with Compression



Zhao Zhang, Lei Huang, Uri Manor, Lingjing Fang, Gabriele Merlo, Craig Michoski, John Cazes, Niall Gaffney.
"FanStore: Enabling Scalable and Efficient I/O for Distributed Deep Learning".
Preprint: https://arxiv.org/abs/1809.10799

Discussion

- The average file size for ImageNet is 108KB
- It is only 7.8-9.5% of the peak throughput of 128KB benchmark on the GPU cluster
- ResNet-50 has 7.7 billion single precision floating operations per image
- FanStore can keep the scaling performance for neural networks that are 10x smaller than ResNet-50

Conclusion

- We profile I/O traffic of three representative applications of CNN, RNN, and GAN
- We design and implement FanStore to speedup deep learning training and reduce I/O between compute nodes and shared file system
- FanStore scales to 512 nodes with over 90% scaling efficiency for real applications

Other Research

- Deep Learning Performance Modeling
- Deep Learning Interpretability

Questions

- zzhang@tacc.utexas.edu