Efficient Neural Network Training for an AI Radiologist on Intel® Xeon® based Supercomputers

Vikram A. Saletore, Ph.D.
Principal Engineer, AI Products Group, Intel

Valeriu Codreanu, Ph.D. and Damian Podareanu. MSc,
Research & Data Scientist, SURFsara B.V.

Lucas A. Wilson, Ph.D. and Alex Filby
HPC and AI Engineering, Dell EMC

The AI Conference, Sept. 4-7, San Francisco

Agenda

- AI Usages & Performance Drivers
- Efficient Scaling of Neural Network Training on Supercomputers
- AI Radiologist Trained on Intel® Xeon® Scalable Processors
- Call To Action

AI Usage Growth

Consumer

Smart Assistants
Chatbots
Search
Personalization
Augmented Reality
Robots

Health

Enhanced Diagnostics
Drug Discovery
Patient Care
Research
Sensory Aids

Finance

Algorithmic Trading
Fraud Detection
Research
Personal Finance
Risk Mitigation

Retail

Support
Experience
Marketing
Merchandising
Loyalty
Supply Chain
Security

Gov't

Defense
Data Insights
Safety & Security
Resident Engagement
Smarter Cities

Energy

Oil & Gas Exploration
Smart Grid
Operational Improvement
Conservation

Transport

Autonomous Cars
Automated Trucking
Aerospace
Shipping
Search & Rescue

Industrial

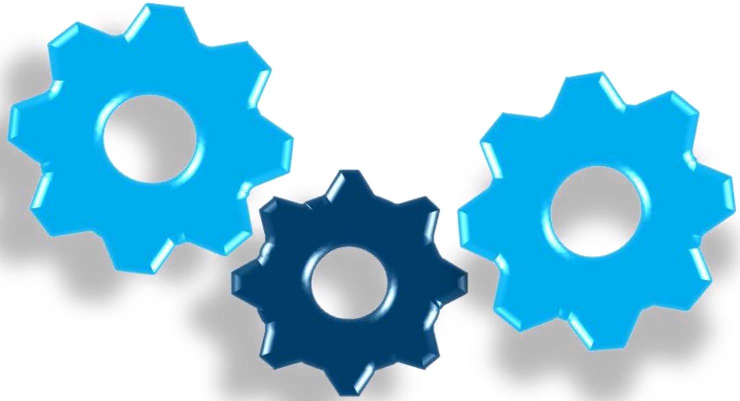
Factory Automation
Predictive Maintenance
Precision Agriculture
Field Automation

Other

Advertising
Education
Gaming
Professional & IT Services
Telco/Media
Space Exploration

Optimization notice

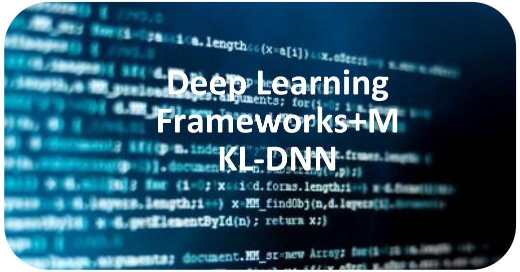
Performance Drivers for AI Workloads



Compute

The FOUNDATION for AI

SW Optimizations



Fabric

Intel® Omni-Path™ Architecture Fabric

AI Portfolio

Solutions

Data Scientists | Technical Services | Reference Solutions

Platforms

- Intel® AI Builders
- Intel® Deep Learning System‡
- Intel Saffron™ REASONING

Tools

- Intel® Deep Learning Studio‡
- Intel® Deep Learning Deployment Toolkit†
- OpenVINO ToolKit
- Intel® Movidius™ Software Development Kit (SDK)

Frameworks

TensorFlow* | Caffe* | mxnet* | BigDL on Spark* | neon | Caffe2*‡ | PYTORCH*‡ | Microsoft CNTK*‡ | PaddlePaddle*‡

libraries

Intel® MKL/MKL-DNN, clDNN, DAAL, Intel Python Distribution, etc.

DIRECT OPTIMIZATION

Intel® nGraph™ Compiler α

CPU Transformer† | NNP Transformer‡ | Other

technology

XEON inside | NERVANA inside‡ | ARRIA 10 inside | ATOM inside* | CORE i7 inside | Iris Graphics | Movidius

END-TO-END COMPUTE

SYSTEMS & COMPONENTS

αAlpha available
†Beta available
‡ Future
**Other names and brands may be claimed as the property of others.*
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

Intel® Xeon® Scalable Processors
the foundation of data center innovation

BCS: https://www.bsc.es/

***TACC (Texas Advanced Computing Center):** https://www.tacc.utexas.edu/*

DellEMC HPC and AI Innovation Lab

Architected for efficient, secure, and agile HPC Supercomputing center

Efficient scaling of Neural Network training on supercomputers

Valeriu Codreanu, Ph.D. and Damian Podareanu

Compute Services, SURFsara B.V.

Intel & SURFsara IPCC* Team

Valeriu Codreanu, Ph.D. (PI)., Senior HPC Consultant, SURFsara* B.V., The Netherlands, Intel Parallel Computing Center

Vikram Saletore, Ph.D. (Co-PI), Principal Engineer & Performance Architect, AI Products Group, Intel Corp.

Damian Podareanu, M.Sc., HPC Consultant, SURFsara* B.V., The Netherlands, Intel Parallel Computing Center

*IPCC: Intel® Parallel Computing Centers

IPCC@SURFsara: Scaling up Deep Learning

Research goals:

- Speeding up time-to-train for deep neural network models on large datasets
- Improve convergence accuracy
- Generalization of methodology across Intel® CPU architectures

Main Results

- Efficient scaling
- 512 Intel®2S Xeon® 8160 nodes, with a TTT of 44 minutes on ImageNet-1K
- Improved SOTA using a reduced number of epochs on ImageNet-1K

Accuracy vs Large Batch Size

Datasets

- ImageNet-1K | 1.2 million | 1000 categories => **~1200 examples / class**
- Chest-Xray14 | 0.07 million | 14 categories => **~200-20000 examples / class**

Training from scratch (< 2% accuracy degradation)

- ImageNet-1K | Batch size up to 32K | ~ 40 updates / epochs | 70-90 epochs

Fine tuning (< 2% accuracy degradation)

- Chest-Xray14 | Batch size up to 8K | ~ 10 updates / epoch | 70-90 epochs

Accuracy, Training Epochs, HW Scaling

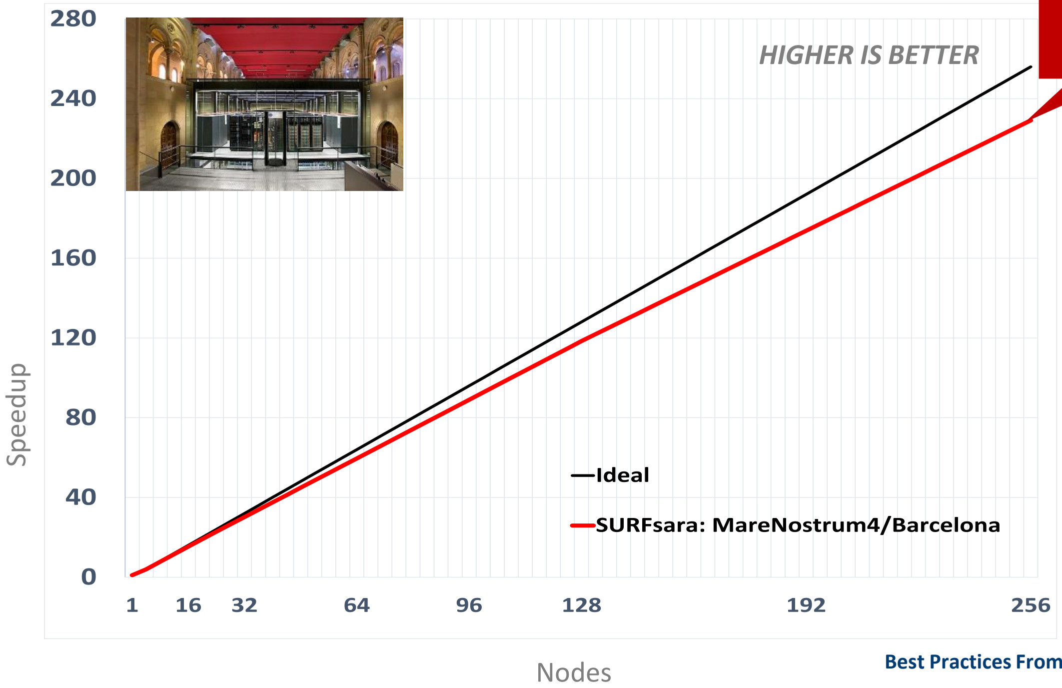
- **Achieving reasonably good to significantly better accuracy requires:**
  - Increased Training time with a fixed level of HW scaling
  - Increased HW scaling for a desired Training Time

- **We show results that trade-off accuracy with the number of training epochs**
  - >74.0% Top-1 Accuracy
  - >75.5% Top-1 Accuracy
  - >76.5% Top-1 Accuracy

- **Using several hardware architectures**
  - Intel® Xeon® Platinum Processor Family with Intel® Omni-Path® Architecture (Intel® OPA) Fabric

BCS: https://www.bsc.es/

ResNet-50 Scaling on 2S Intel® Xeon® Platinum 8160 Processor Cluster



Up to
90%
Scaling Efficiency

MareNostrum4 Barcelona Supercomputing Center

- 90% Scaling Efficiency
- Top-1/Top-5 > 74%/92%
- Global BS=8192
- Throughput: 15170 Img/sec
- Time-To-Train: 70 minutes

Best Practices From SURFsara B.V: https://surfdrive.surf.nl/files/index.php/s/xrEFLPvo7IDRARs

ResNet-50 Training Time to 74% Top-1 Accuracy

Intel® Xeon® Platinum 8160 Processor Cluster MareNostrum4*

Intel® Distribution of Caffe* with ImageNet-1K dataset

TRAINING TIME | **TRAINING ACCURACY**

56
Minutes

LOWER IS BETTER

74.0%
Top-1

HIGHER IS BETTER

400 Intel® Xeon® Processor Nodes

Global BS= 6400 & 50 Epochs
19047 Images/sec

Optimized DL Framework+MKL-DNN

TRAINING TIME | **TRAINING ACCURACY**

44
Minutes

LOWER IS BETTER

74.0%
Top-1

HIGHER IS BETTER

512 Intel® Xeon® Processor Nodes

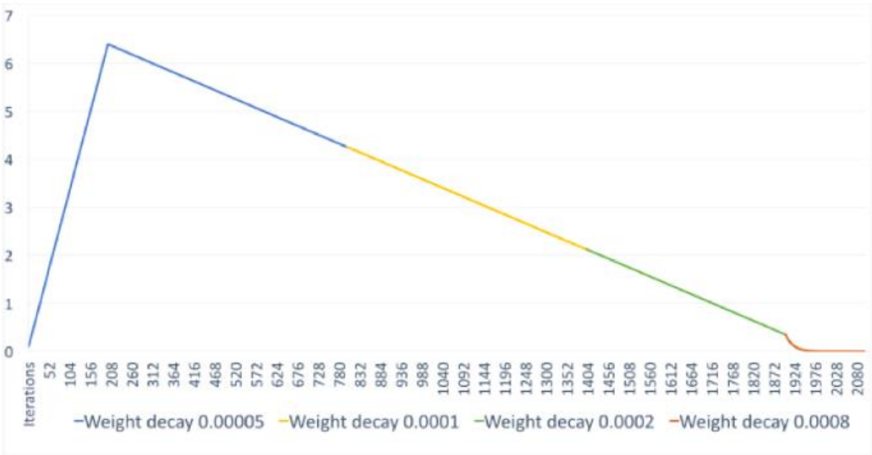
Global BS= 8192 & 50 Epochs
24240 Images/sec

****MareNostrum4*** *(Barcelona Supercomputing Center):* https://www.bsc.es/marenostrum

Extremely Large Batch Size convergence

Weight decay scaling throughout training eases the optimisation problem further

- 64K batch size: convergence in 2100 iterations to ~74% top-1 accuracy!



Batch size	8K	16K	32K	48K	64K
IBM [1]	75%	-	-	-	-
Facebook [2]	76.2%	75.2%	72.4%	-	66%
You et al. [3]	75.3%	75.3%	74.7%	-	72%
This work [4]	**76.6%**	**76.3%**	**75.3%**		**74%**

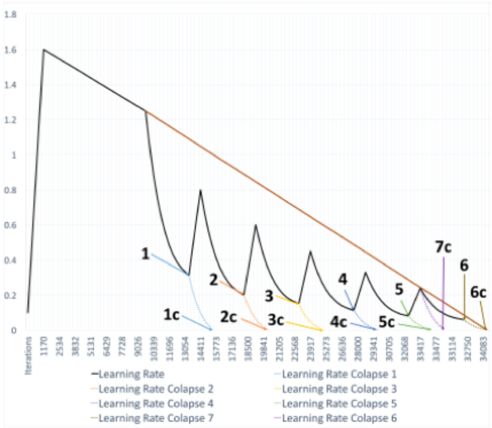
[1] Cho, M., Finkler, U., Kumar, S., Kung, D., Saxena, V., Sreedhar, D.: PowerAI DDL. arXiv

[2] Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch sgd: Training imagenet in hour. arXiv

[3] You, Y., Zhang, Z., Demmel, J., Keutzer, K., Hsieh, C.J.: Imagenet training in minutes. arXiv

[4] Codreanu, V., Podareanu, D., Saletore, V: Scale out for large minibatch SGD: Residual network training on ImageNet-1K with improved accuracy and reduced time to train, arXiv

Increasing Accuracy Using Collapsed Ensembles



No. on plot	Top-1 % acc.	Top-5 % acc.
1	68.33	88.71
1c	75.50	92.83
2	71.54	90.78
2c	76.15	93.17
3	73.28	91.58
3c	76.50	93.24
4	73.31	91.53
4c	76.57	93.24
5	73.89	91.97
5c	76.83	93.32
6	74.49	92.13
6c	76.81	93.32
7c	76.70	93.32

Fig. 3. Plot of learning rate behaviour when obtaining the ensemble snapshots

Collapsed ensembles

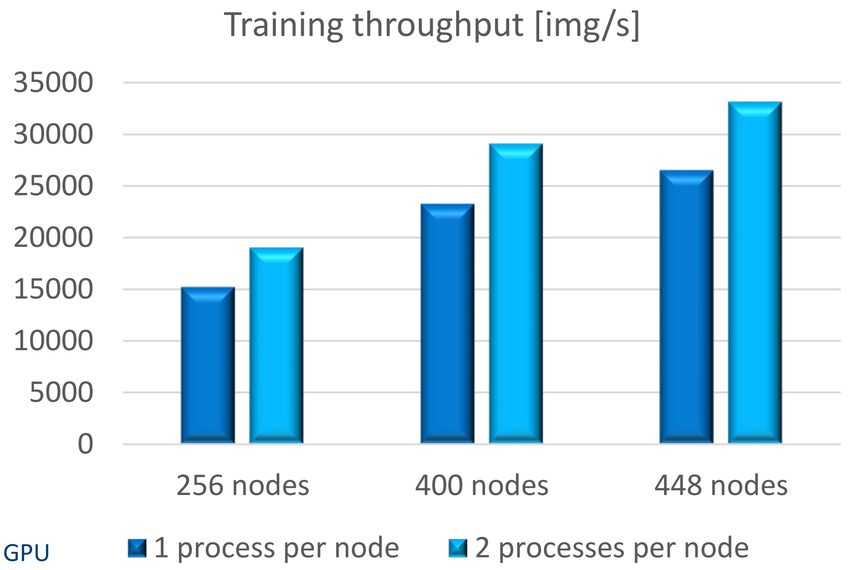
Similar in fashion to the learning-rate collapses:

- However, after performing a partial collapse, LR is again increased
- Cycling the LR:
  - Improves single-model accuracy faster
  - Ensemble of the collapsed points leads to 77.5% accuracy using a ResNet-50 regular training budget

https://github.com/sara-nl/caffe/tree/master/models/intel_optimized_models/multinode/resnet50_custom_lr

Improving Hardware Efficiency

- Using 2 training processes per node increases HW efficiency significantly!
  - Each process has a local batch size of 16. At 448 nodes, global batch size is 14336, so no convergence issues.
  - Each process is pinned to a separate NUMA domain
  - Scaling efficiency is not negatively impacted (until 512 nodes)
- Caffe achieves good HW efficiency now!



Comparing efficiency of CPU to GPU-based training of ResNet50. GPU peak performance **does not include** the CPU hosts

Work	HW type	# nodes (devices)	Peak [FP32]	TTT	HW eff.
This work	SKX 8160	448 (896)	2682 TF	58 min	12.36
Facebook [5]	NVIDIA P100	32 (256)	2658 TF	60 min	12.03
You *et al.* [26]	SKX 8160	1024 (2048)	6144 TF	48 min	6.51

https://github.com/sara-nl/caffe/tree/master/models/intel_optimized_models/multinode/resnet50_448nodes

Best Practices To Improve Accuracy of ResNet-50

Technique	Approximate top-1 accuracy
Default augmentation	74,0%
Warm-up of LR	75,4%
Polynomial decay	75,7%
Weight decay scaling	76,2%
Single collapse	76,6%
Collapsed ensembles	77,5%

Summary of Caffe Work

- Extensively evaluated Intel® Xeon® Platinum Processors with Intel® OPA Fabric on training ResNet-50 with ImageNet-1K:
  - >90% scaling efficiency up to 256 nodes to achieve Top-1 >74% Accuracy
  - >85% scaling efficiency from 256 to 512 SKX nodes & achieve 76.5% Top-1
  - NUMA-awareness improves throughput significantly

- Introduced several techniques to improve accuracy:
  - Collapses
  - Weight decay scaling
  - Achieve SOTA at batch sizes of up to 64K

- Models achieve a SOTA of >76.5%+ Top-1 accuracy for ResNet-50 Benchmark

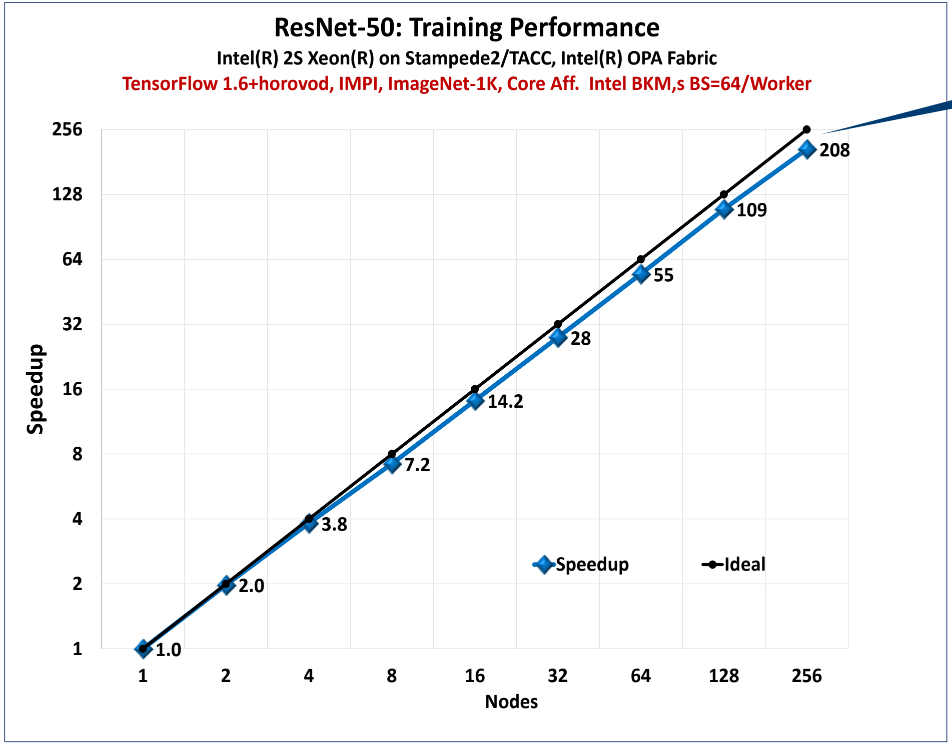
- Collapsed ensemble techniques lead to 77.5% accuracy using ResNet-50

Extending to Tensorflow and to scientific disciplines

Tensorflow Scalability on Intel® Xeon® Processors

ResNet-50 Scaling Efficiency With TensorFlow

Intel® Xeon® Platinum 8160 processor Cluster Stampede2 at TACC



81% Efficiency with TensorFlow+horovod

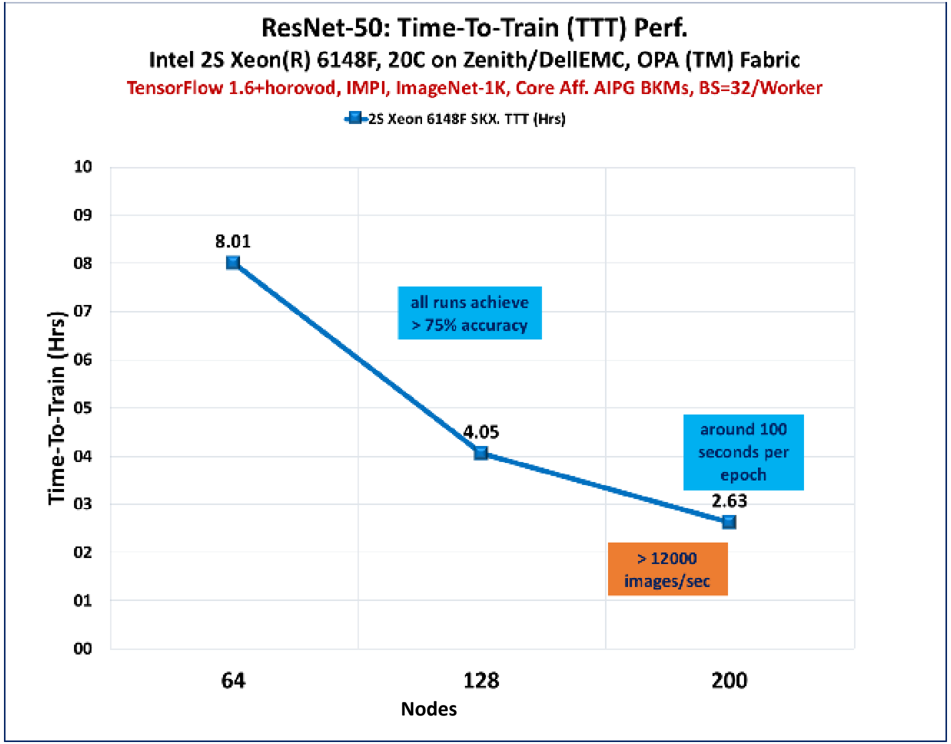
ResNet-50 with ImageNet-1K on 256 Nodes on Stampede2/TACC:

- Improved single-node perf with multi-workers/node
- 81% scaling efficiency
- Batch size of 64 per worker: Global BS=64K
- 16400 Images/sec on 256 nodes
- 26700 images/sec on 512 nodes
- Time-To-Train: ~2 Hrs on 256 Nodes

First to achieve convergence with state-of-the-art accuracy with TensorFlow on 256 node Intel® Xeon® cluster

Scaling up Training On ImageNet-1K

Intel® Xeon® Gold 6148F processor Zenith* cluster at DellEMC



DenseNet-121 Training at Scale

Global batch size	# nodes	# epochs	Time/epoch (secs/epoch)	Time-To-Train	% Top1 Accuracy
8192	64	90	346 s	8h40m	74.9
16384	128	64	187.5 s	3h20	74.5

These models are to be further fine-tuned on the real-world dataset: Chest-Xray14

Transfer Learning Using Highly Accurate Benchmark for Real Use Case

Fine-tuned ResNet-50 that was pre-trained on ImageNet using the Zenith cluster.

To increase accuracy:

- When picking a pre-trained checkpoint do not pick the last one.
- Start with the learning rate at which the model was training when it was checkpointed.
- Perform gradual warmup of the learning rate, proportionally to the global batch size.

Comparative timings for 128-node fine-tuning run

Global batch size	Framework	# nodes	Time/epoch
4096	Keras	128	**85 s**
4096	Tensorflow	128	**18 s**

An AI Radiologist Trained on Intel® Xeon® Scalable Processors

Automatically Identifying Thoracic Pathologies in Chest X-rays

Lucas A. Wilson, Ph.D. and Alex Filby

HPC and AI Engineering, DellEMC

DellEMC AI Engineering Team – Intel Projects

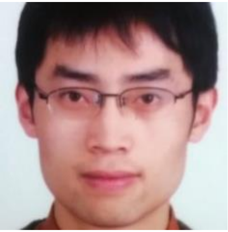
Onur Celebioglu
Director, HPC and AI Engineering

Quy Ta
Manager, AI Engineering

Lucas A. Wilson
Artificial Intelligence Research

Vineet Gundecha
AI Software Principal Engineer

Srinivas Varadharajan
AI Software Principal Engineer



Pei Yang
AI Software Principal Engineer

Alex Filby
Sr. Systems Development Engineer

The Importance of Early Detection

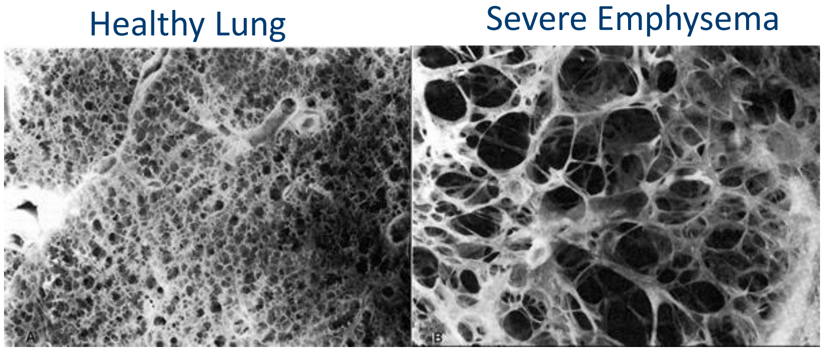
Emphysema is estimated to affect more than
1. 3 million people in the U.S.
2. 65 million people worldwide

- Severe emphysema (types 3 / 4) are life threatening
  - Early detection is important to try to halt progression

Pneumonia affects more than 1 million people each year in the U.S.3, and more than 450 million4 each year worldwide.

- 1.4 million deaths per year worldwide
  - Treatable with early detection

1. www.emphysemafoundation.org/index.php/the-lung/copd-emphysema
2. http://www.who.int/respiratory/copd/burden/en/
3. https://www.cdc.gov/features/pneumonia/index.html
4. https://doi.org/10.1016%2FS0140-6736%2810%2961459-6



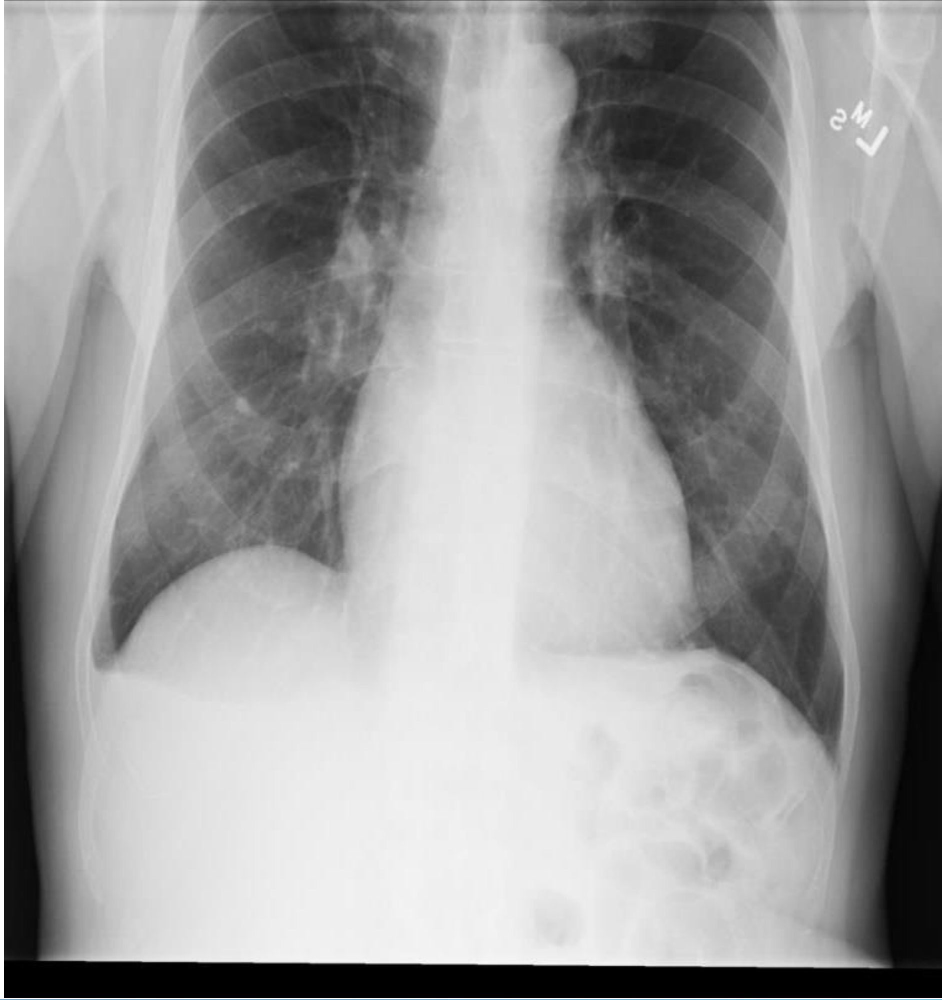
https://www.ctsnet.org/article/airway-bypass-stenting-severe-emphysema

CheXNet

Developed at Stanford University, CheXNet is a model for identifying thoracic pathologies from the NIH ChestXray14 dataset

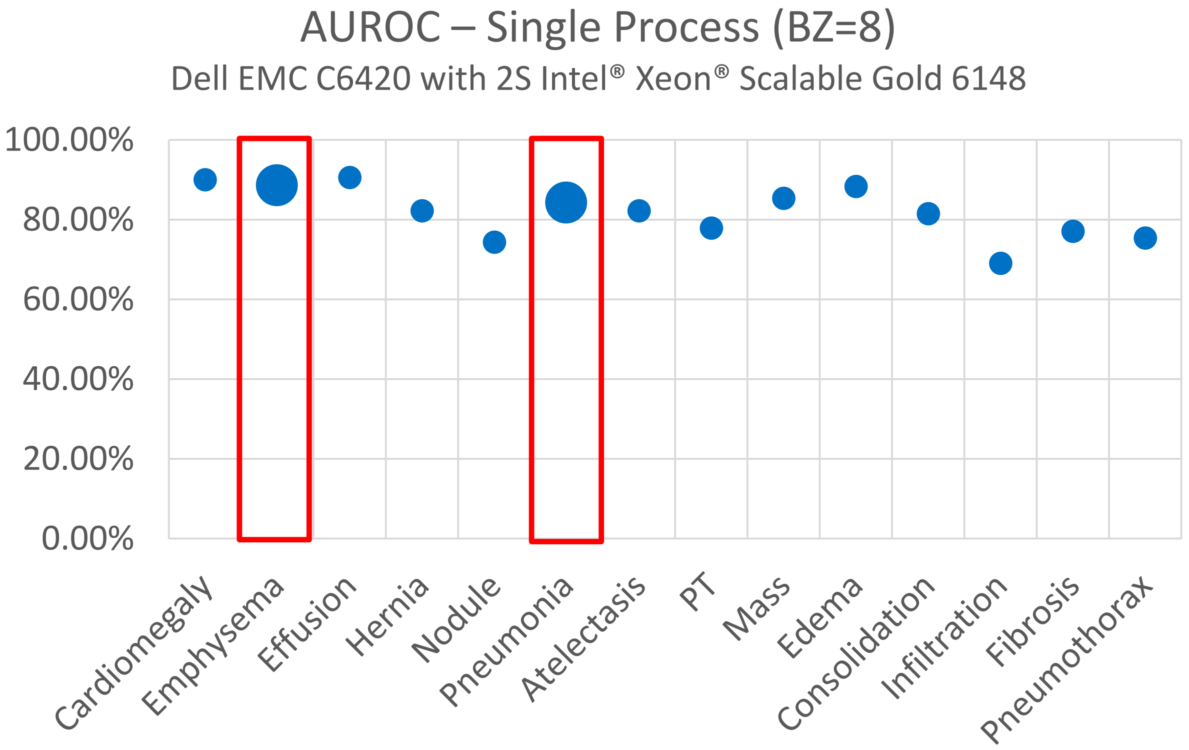
- DenseNet121 topology
  - Pretrained on ImageNet
- Dataset contains 112K images
  - Multicategory / Multilabel
  - Unbalanced

http://academictorrents.com/details/557481faacd824c83fbf57dcf7b6da9383b3235a
https://stanfordmlgroup.github.io/projects/chexnet/



Building CheXNet

Training CheXNet



High-accuracy model

- ✓ 84% accuracy identifying pneumonia
- ✓ 89% accuracy identifying emphysema

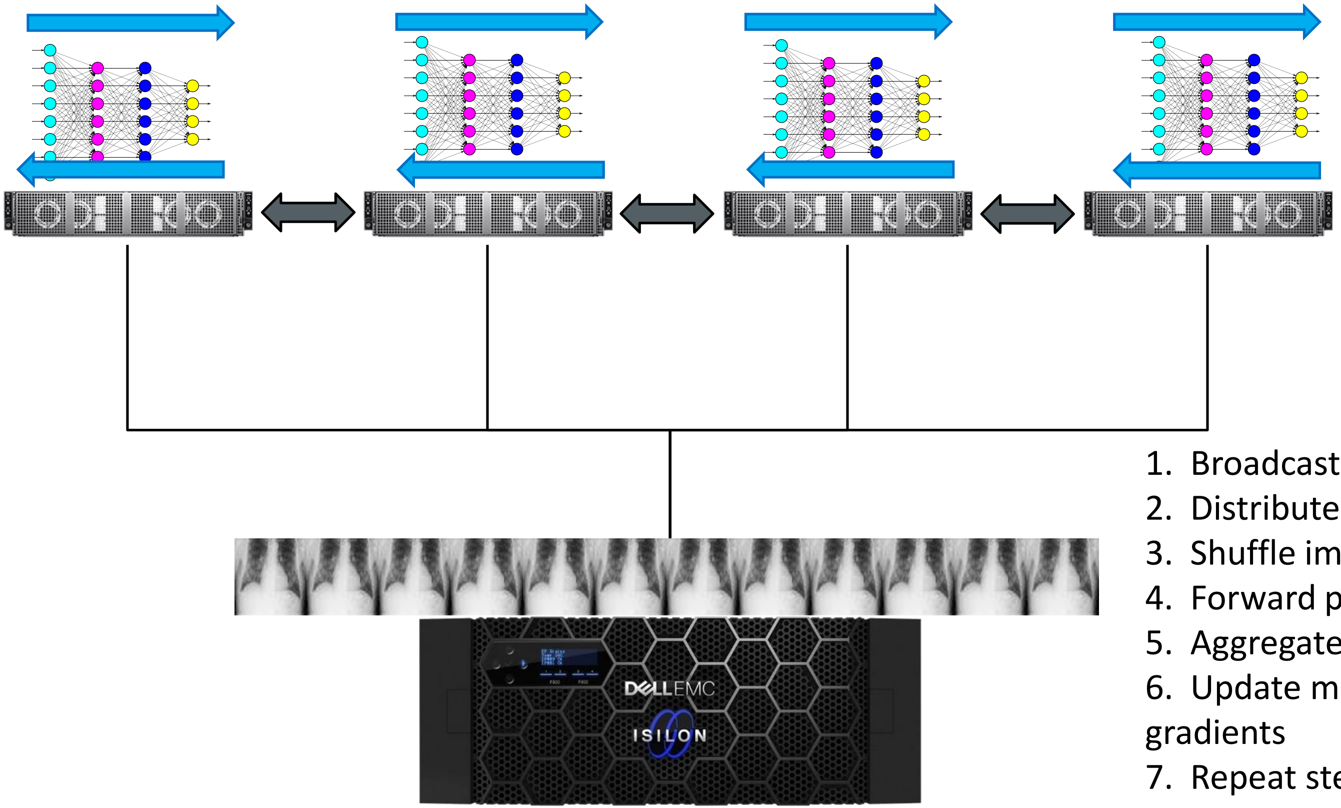
Baseline performance on CPUs

- 4 images per second
- 1 epoch takes 5 hours!

Parallelizing CheXNet

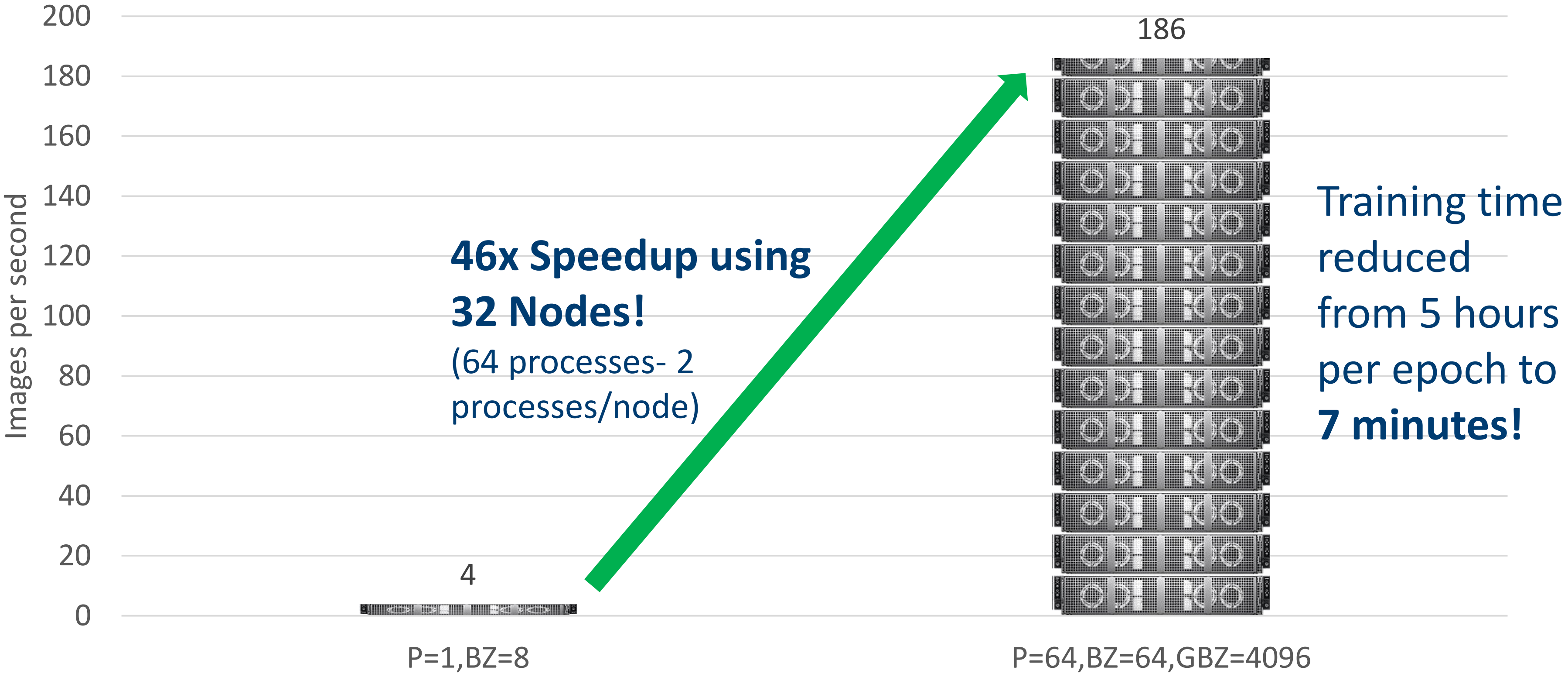
K + TensorFlow + HOROVOD

Faster Model Development with Distributed Deep Learning



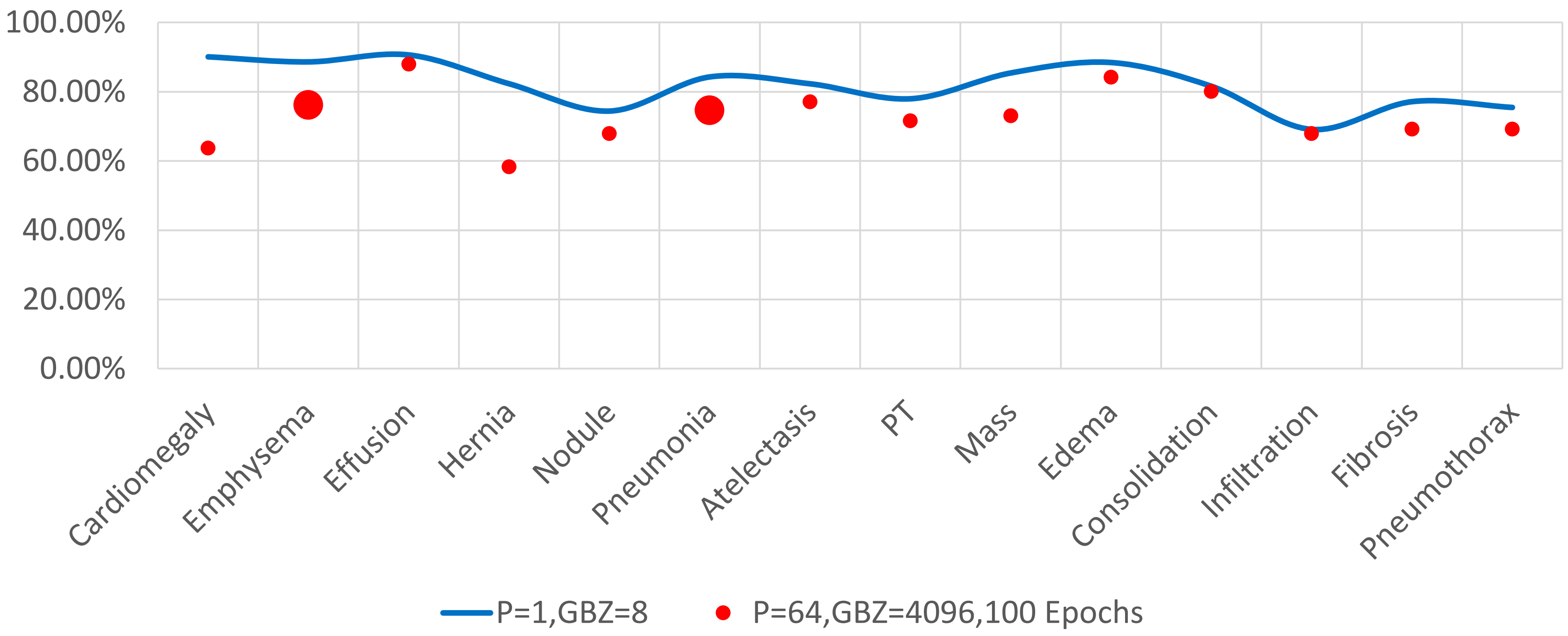
1. Broadcast initial model
2. Distribute images
3. Shuffle images
4. Forward pass (inference)
5. Aggregate Gradients
6. Update model based on aggregated gradients
7. Repeat steps 3-6

CheXNet – Parallel Speedup

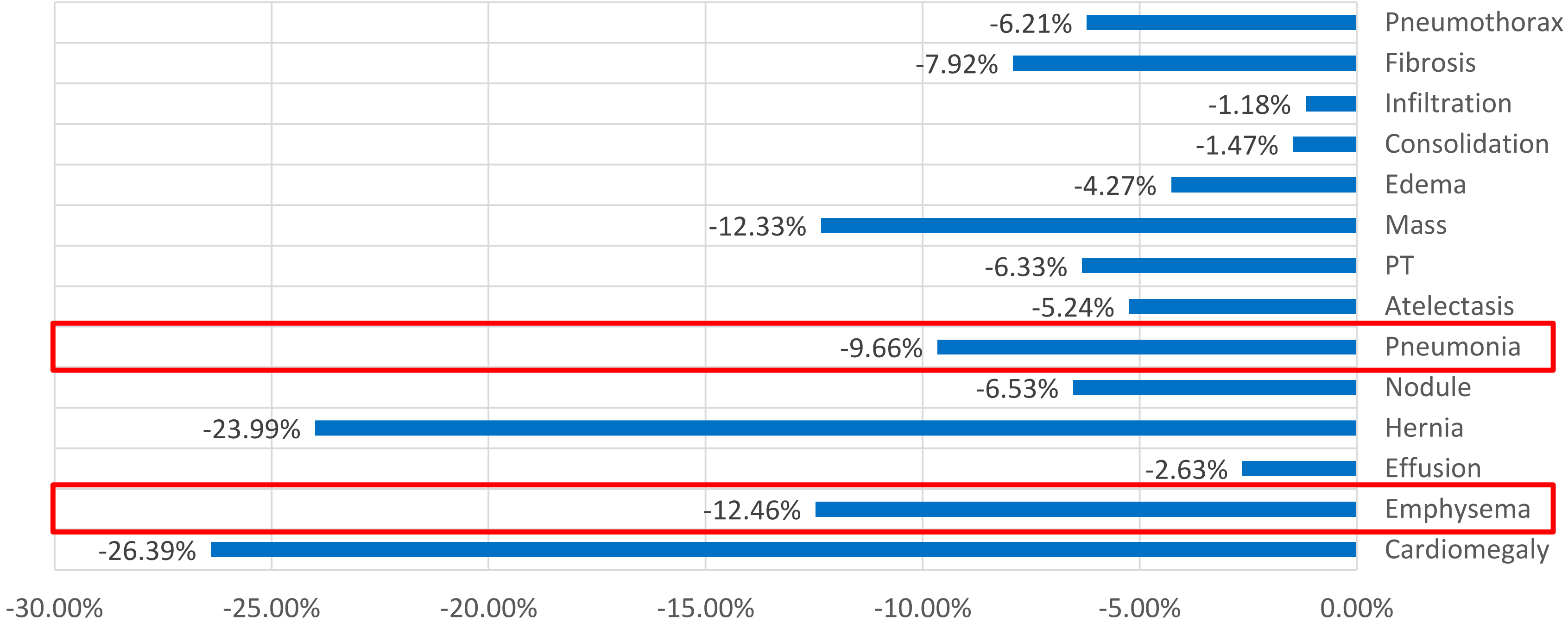


Dell EMC PowerEdge C6420 with dual Intel® Xeon® Scalable Gold 6148 on Intel ® Omni-Path fabric.

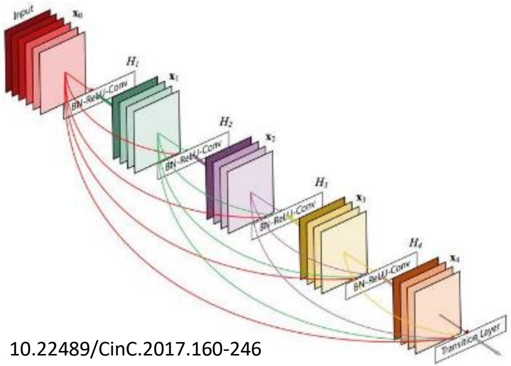
Parallelizing CheXNet - Accuracy



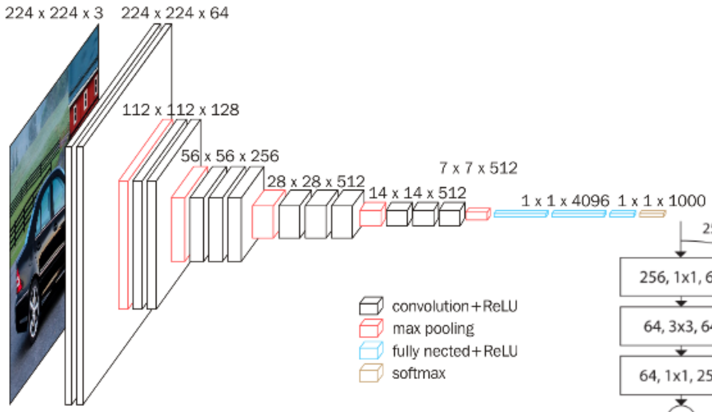
Parallelizing CheXNet – Accuracy Relative to single-process

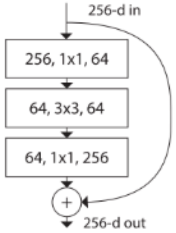


Can We Do Better?



10.22489/CinC.2017.160-246





DenseNet121 is a very deep topology with lots of batch normalization

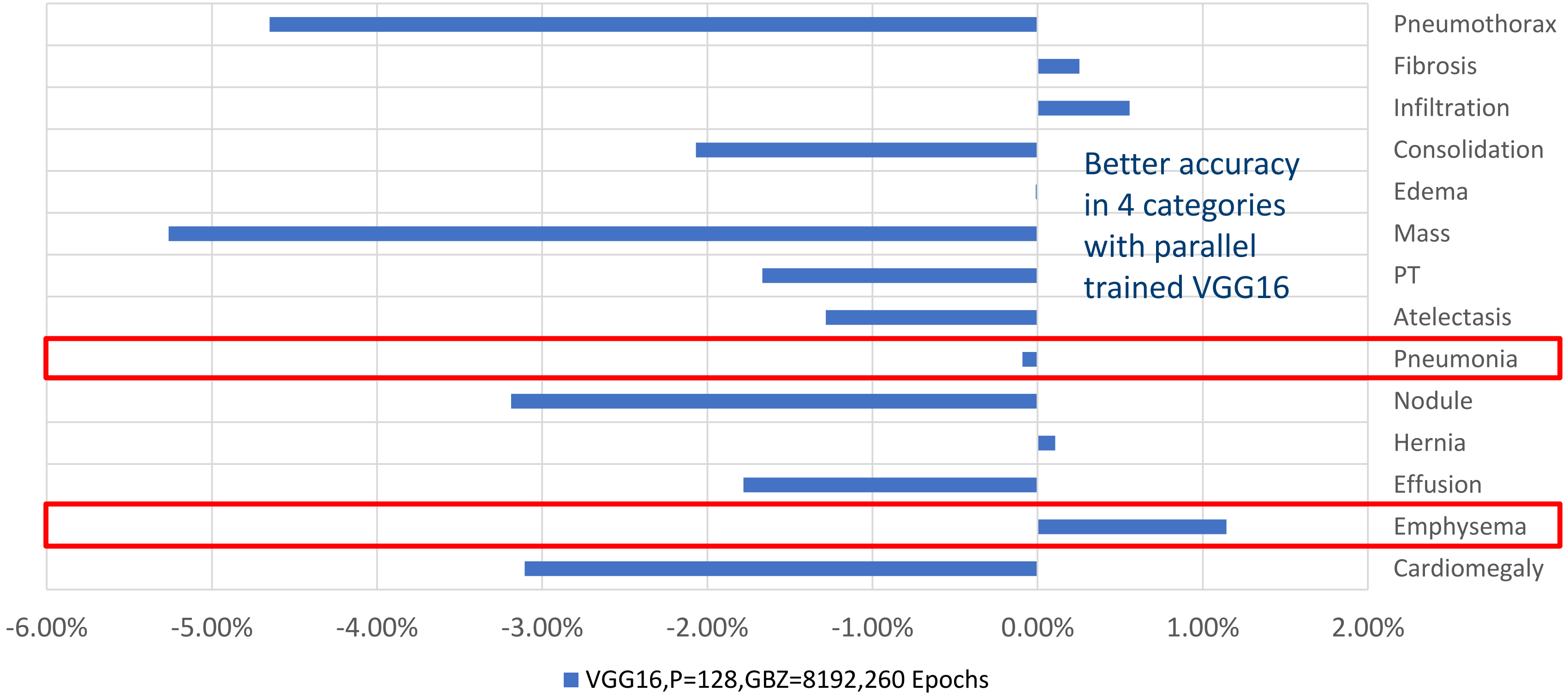
- Batch normalization with large batches (thousands) can hinder convergence

VGG16 and ResNet50 are shallower topologies with less batch normalization

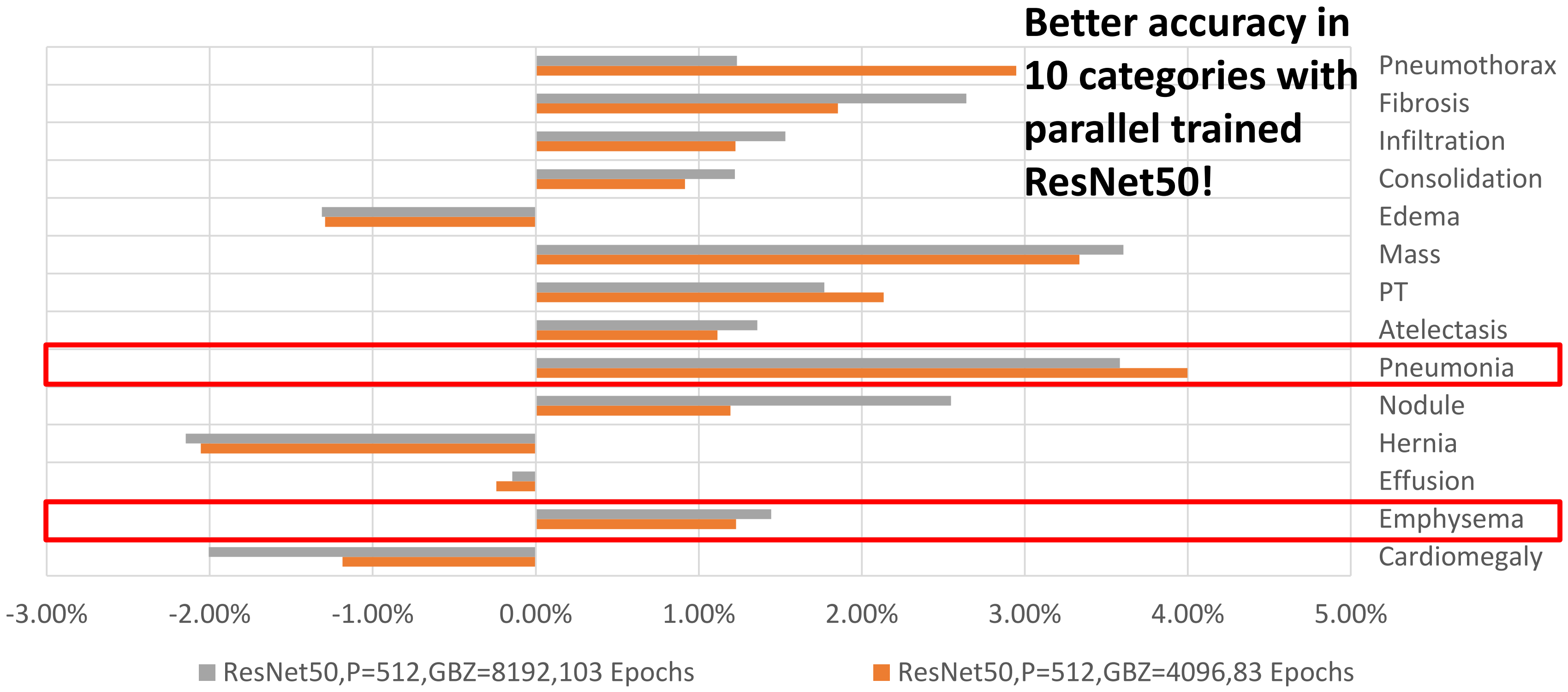
- ResNet50 contains less than half the batch normalization layers of DenseNet121
- VGG16 has no batch normalization

Why not try another topology?

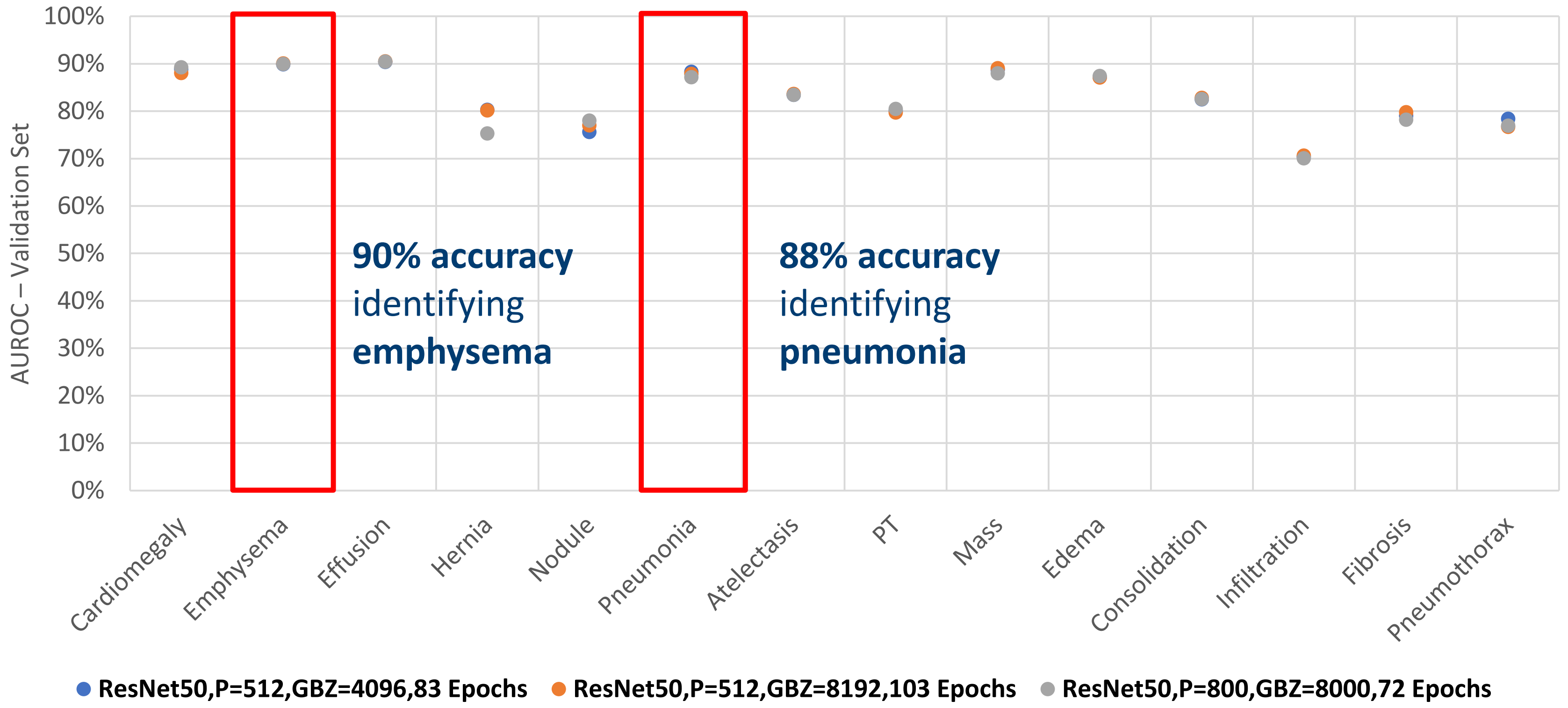
Accuracy of VGG16 relative to DenseNet-121



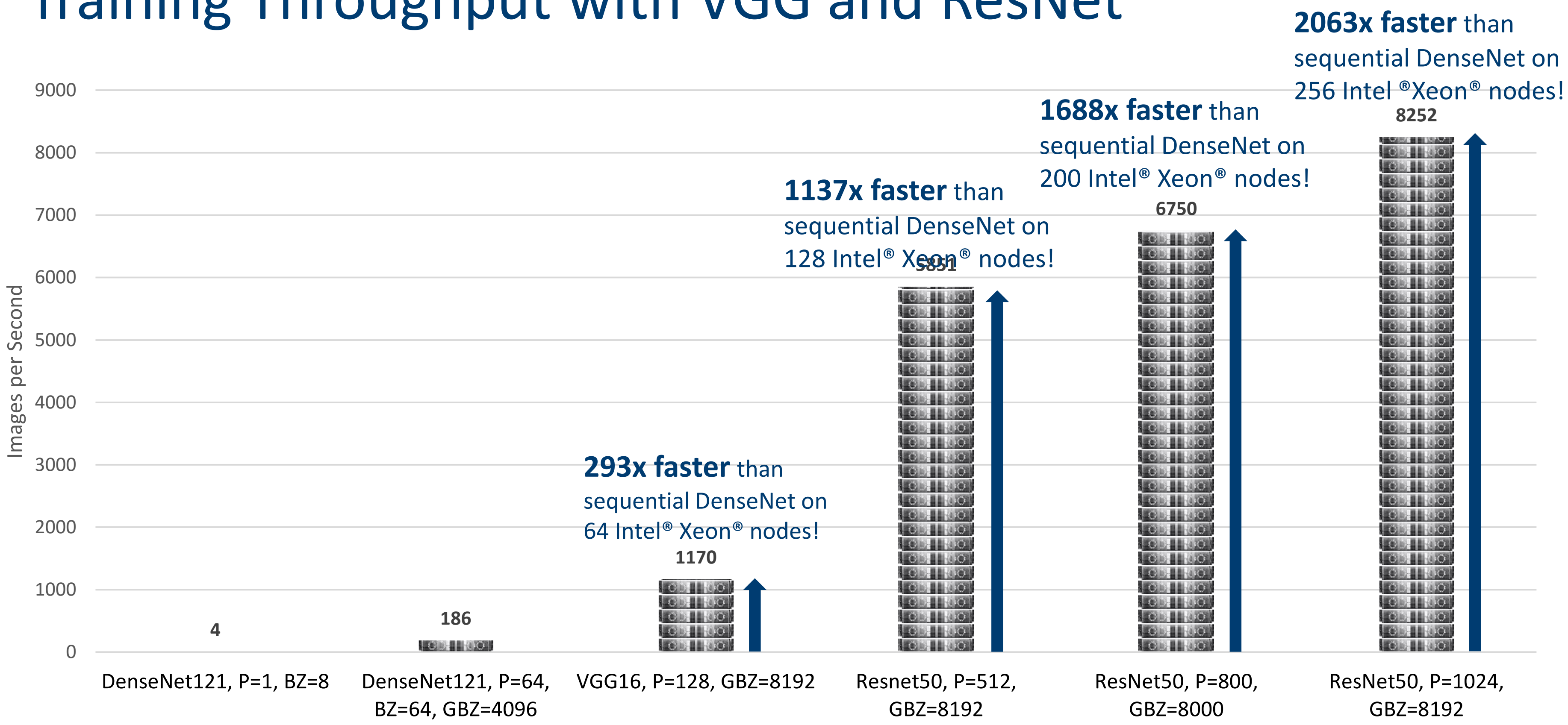
Accuracy of ResNet50 relative to DenseNet-121



Categorical Accuracy of ResNet-50 based AI Radiologist

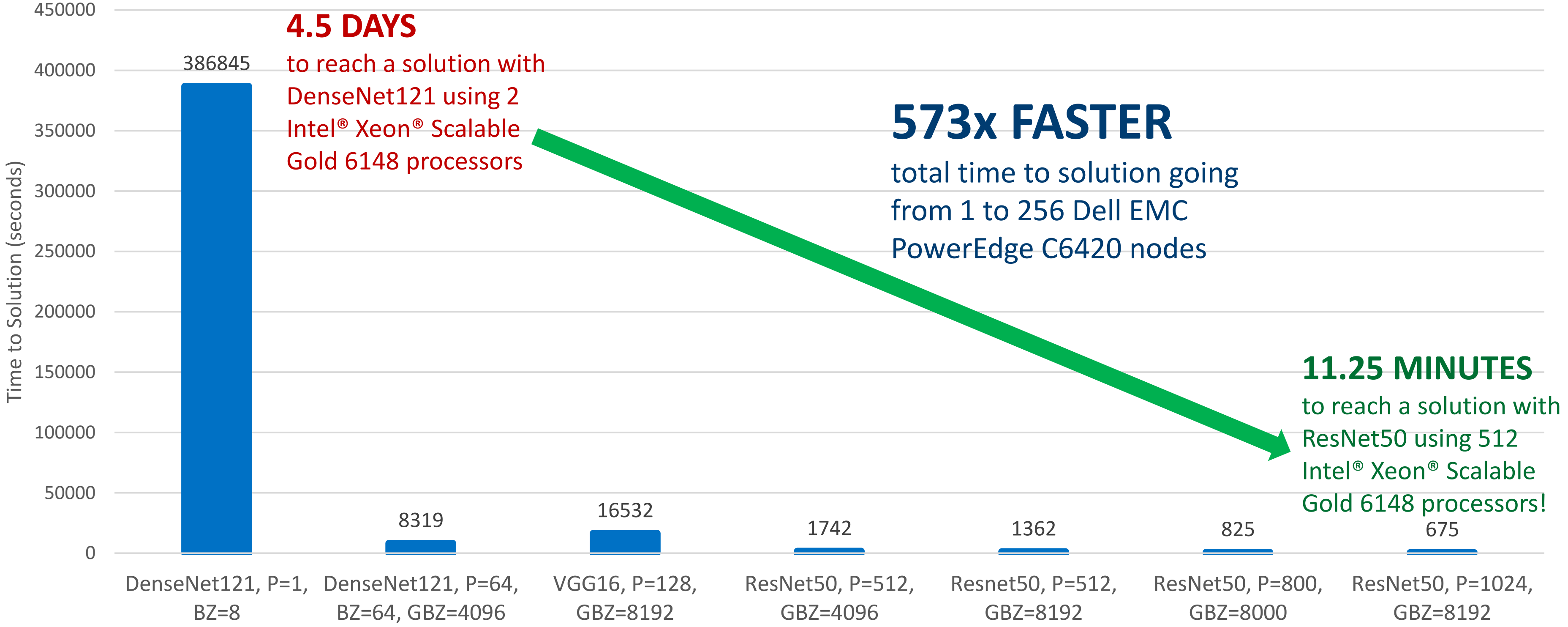


Training Throughput with VGG and ResNet



Dell EMC PowerEdge C6420 with dual Intel® Xeon® Scalable Gold 6148 on Intel ® Omni-Path fabric. ResNet50 tests performed with TensorFlow+Horovod

Time to Solution DenseNet vs VGG vs ResNet



Dell EMC PowerEdge C6420 with dual Intel® Xeon® Scalable Gold 6148 on Intel ® Omni-Path fabric. ResNet50 tests performed with TensorFlow+Horovod

Call To Action

learn

More information at
ai.intel.com/framework-optimizations/

- Tensorflow: https://ai.intel.com/tensorflow/
- Blog: http://www.techenablement.com/surfsara-achieves-accuracy-performance-breakthroughs-deep-learning-wide-network-training/
- SURFsara-Caffe* Blog: SURFsara* Caffe* blog: https://blog.surf.nl/en/imagenet-1k-training-on-intel-xeon-phi-in-less-than-40-minutes/
- SURFsara-Intel Paper: https://arxiv.org/pdf/1711.04291.pdf
- Intel Blog: https://ai.intel.com/accelerating-deep-learning-training-inference-system-level-optimizations/
- SURFsara* Best Practices for Caffe*: https://github.com/sara-nl/caffe
- SURFsara Best Practices for TensorFlow: https://surfdrive.surf.nl/files/index.php/s/xrEFLPvo7IDRARs
- Dell EMC Ready Solutions for AI Blog: https://community.dellemc.com/community/products/rs_for_ai

explore

engage

Use Intel's performance-optimized libraries & frameworks

Contact us/Intel for help and collaboration opportunities

Vikram Saletore, Ph.D.
vikram.a.saletore@intel.com

Valeriu Codreanu, Ph.D.
valeriu.codreanu@surfsara.nl

Lucas A. Wilson, Ph.D.
luke_wilson@dell.com
@lucasawilson

Damian Podreanu, MSc
damian.podareanu@surfsara.nl

Alex Filby
alex_filby@dell.com

intel® AI

Legal Notices & Disclaimers

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer. No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Stampede2*/TACC* Configuration Details

***Stampede2/TACC**: https://portal.tacc.utexas.edu/user-guides/stampede2

Compute Nodes: 2 sockets Intel® Xeon® Platinum 8160 CPU with 24 cores each @ 2.10GHz for a total of 48 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux 6.7.

TensorFlow 1.6: Built & Installed from source: https://www.tensorflow.org/install/install_sources

Model: Topology specs from https://github.com/tensorflow/tpu/tree/master/models/official/resnet (ResNet-50); Batch size as stated in the performance chart

Convergence & Performance Model: https://surfdrive.surf.nl/files/index.php/s/xrEFLPvo7IDRARs

Dataset: ImageNet2012-1K: http://www.image-net.org/challenges/LSVRC/2012/

Performance measured on 256 Nodes with:

```
OMP_NUM_THREADS=24 HOROVOD_FUSION_THRESHOLD=134217728 export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2 \
mpirun -np 512 -ppn 2 python resnet_main.py --train_batch_size 8192 --train_steps 14075 --num_intra_threads 24 --num_inter_threads 2 --mkl=True  --data_dir=/scratch/04611/valeriuc/tf-1.6/tpu_rec/train --model_dir model_batch_8k_90ep --use_tpu=False --kmp_blocktime 1
```

*DellEMC Zenith Cluster Configuration Details

***DellEMC Internal Cluster**:

Compute Nodes: 2 sockets Intel® Xeon® Gold 6148F CPU with 20 cores each @ 2.40GHz for a total of 40 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux 6.7.

TensorFlow 1.6: Built & Installed from source: https://www.tensorflow.org/install/install_sources

ResNet-50 Model: Topology specs from https://github.com/tensorflow/tpu/tree/master/models/official/resnet
DenseNet-121Model: Topology specs from https://github.com/liuzhuang13/DenseNet

Convergence & Performance Model: https://surfdrive.surf.nl/files/index.php/s/xrEFLPvo7IDRARs

Dataset:
ImageNet2012-1K: http://www.image-net.org/challenges/LSVRC/2012/
ChexNet: https://stanfordmlgroup.github.io/projects/chexnet/

Performance measured with:

```
OMP_NUM_THREADS=24 HOROVOD_FUSION_THRESHOLD=134217728 export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2 \
mpirun -np 512 -ppn 2 python resnet_main.py --train_batch_size 8192 --train_steps 14075 --num_intra_threads 24 --num_inter_threads 2 --
mkl=True  --data_dir=/scratch/04611/valeriuc/tf-1.6/tpu_rec/train --model_dir model_batch_8k_90ep --use_tpu=False --kmp_blocktime 1
```

MareNostrum4/BSC* Configuration Details

***MareNostrum4/Barcelona Supercomputing Center**: https://www.bsc.es/

Compute Nodes: 2 sockets Intel® Xeon® Platinum 8160 CPU with 24 cores each @ 2.10GHz for a total of 48 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux 6.7.

Intel® Distribution of Caffe*: http://github.com/intel/caffe/, revision 8012927bf2bf70231cbc7ff55de0b1bc11de4a69.
Intel® MKL version: mklml_lnx_2018.0.20170425; Intel® MLSL version: l_mlsl_2017.1.016

Model: Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (ResNet-50) and modified for wide-RedNet-50. Batch size as stated in the performance chart

Time-To-Train: measured using "train" command. Data copied to memory on all nodes in the cluster before training. No input image data transferred over the fabric while training; Performance measured for node count: 128, 192, 256, 400, 512 & Performance projected for node count: 1-64.

Performance measured with:
export OMP_NUM_THREADS=44 (the remaining 4 cores are used for driving communication), export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2

OMP_NUM_THREADS=44 KMP_AFFINITY="proclist=[0-87],granularity=thread,explicit" KMP_HW_SUBSET=1t MLSL_NUM_SERVERS=4 mpiexec.hydra -PSM2 -l -n $SLURM_JOB_NUM_NODES -ppn 1 -f hosts2 -genv OMP_NUM_THREADS 44 -env KMP_AFFINITY "proclist=[0-87],granularity=thread,explicit" -env KMP_HW_SUBSET 1t -genv I_MPI_FABRICS tmi -genv I_MPI_HYDRA_BRANCH_COUNT $SLURM_JOB_NUM_NODES -genv I_MPI_HYDRA_PMI_CONNECT alltoall sh -c 'cat /ilsvrc12_train_lmdb_striped_64/data.mdb > /dev/null ; cat /ilsvrc12_val_lmdb_striped_64/data.mdb > /dev/null ; ulimit -u 8192 ; ulimit -a ; numactl -H ; /caffe/build/tools/caffe train --solver=/caffe/models/intel_optimized_models/multinode/resnet_50_256_nodes_8k_batch/solver_poly_quick_large.prototxt -engine "MKL2017"