LBANN: Livermore Big Artificial Neural Network HPC Toolkit

MLHPC 2015

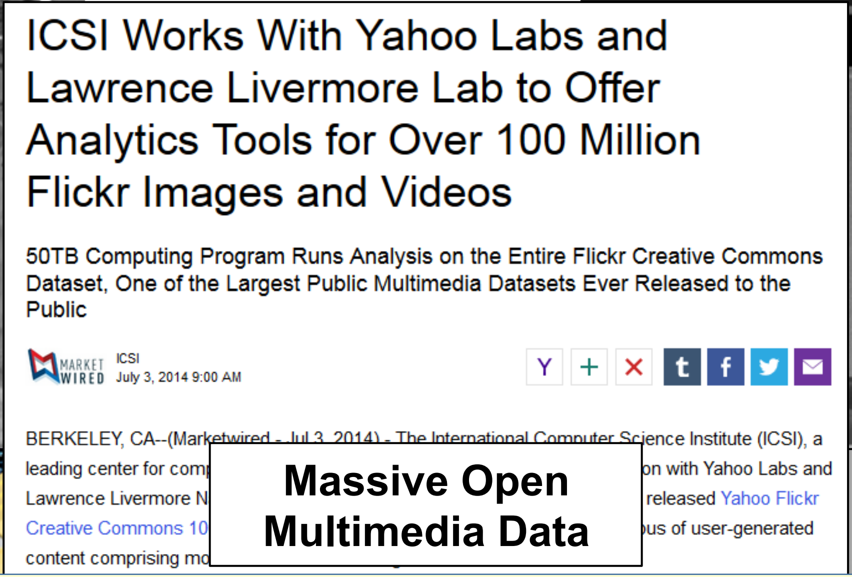
Brian Van Essen, Hyojin Kim, Roger Pearce, Kofi Boakye, Barry Chen
Center for Applied Scientific Computing (CASC) + Computational Engineering

Nov. 15, 2015

LLNL-PRES-679368
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

National Security, Science, and Economic Competitiveness Applications are Generating Ever-Growing Collections of Data

Aerial Platforms



ICSI Works With Yahoo Labs and Lawrence Livermore Lab to Offer Analytics Tools for Over 100 Million Flickr Images and Videos

50TB Computing Program Runs Analysis on the Entire Flickr Creative Commons Dataset, One of the Largest Public Multimedia Datasets Ever Released to the Public

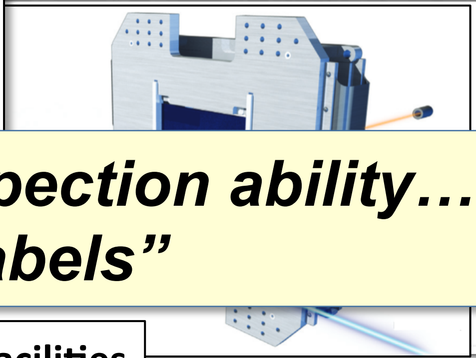
MARKET WIRED ICSI
July 3, 2014 9:00 AM

BERKELEY, CA--(Marketwired - Jul 3, 2014) - The International Computer Science Institute (ICSI), a leading center for comp... on with Yahoo Labs and Lawrence Livermore N... released Yahoo Flickr Creative Commons 10... us of user-generated content comprising mo...

Massive Open Multimedia Data

Advanced Manufacturing

Neural Interfaces



Science Facilities

Pace of collection far exceeds human inspection ability…
"We have Big Data but Small Labels"

Big Data … Small Labels

- Focus on training with unsupervised feature extraction
  - Stacked auto-encoders
  - Fine-tune with small labels

- Moving beyond strict image processing
  - Biological data sets
  - Sensors from large scientific instruments (e.g. NIF)
  - Incorporate imagery with additional sensor modalities (e.g. additive manufacturing)

- Preliminary focus on large, fully connected dense layers
  - Extend to unrolled RNN
  - Adding support for convolutional kernels

- Optimize for data-intensive HPC systems
  - Distributed memory algorithm
  - Low latency interconnect
  - Node-local NVRAM
  - State-of-the art distributed linear algebra library

Extracting Parallelism

- Model Parallelism (train a single model faster)
  - Distributed algorithm across multiple HPC nodes
  - Future work will extend this to include attached GPU accelerators

- Data Parallelism (process data faster)
  - Larger mini-batches reduce synchronization steps
  - Leverage node-local NVRAM for data staging
    - Overlap communication with computation

- Future work:
  - GPU-offload
  - Train multiple models concurrently
  - Node-local data amplification

Computational Horsepower is Required for the Deep Learning

- Andrew Ng's Deep Learning Rocket Analogy:
  - Powerful engine: Use large ***Low Bias*** models
  - Rocket fuel: ***Minimize Variance*** with vast training data

HPC resources enable the training of massive-scale Deep Learning networks

ICSI Works With Yahoo Labs and Lawrence Livermore Lab to Offer Analytics Tools for Over 100 Million Flickr Images and Videos

50TB Computing Program Runs Analysis on the Entire Flickr Creative Commons Dataset, One of the Largest Public Multimedia Datasets Ever Released to the Public

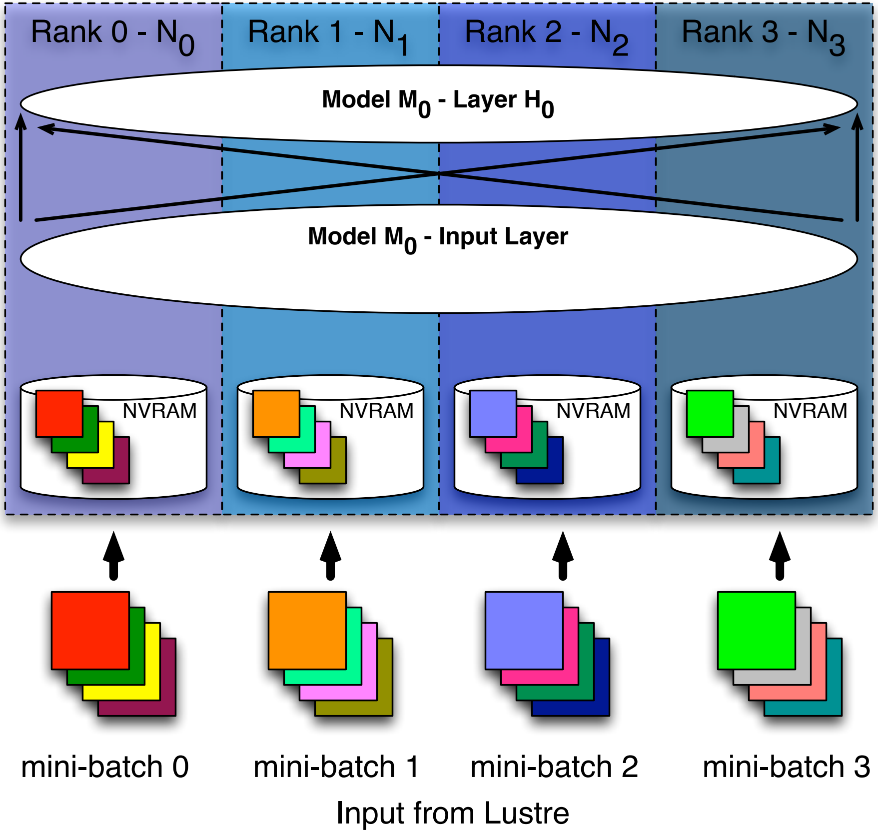
MARKET WIRED ICSI
July 3, 2014 9:00 AM

BERKELEY, CA--(Marketwired - Jul 3, 2014) - The International Computer Science Institute (ICSI), a leading center for computer science research, today announced a collaboration with Yahoo Labs and Lawrence Livermore National Laboratory to process and analyze the recently released Yahoo Flickr Creative Commons 100 Million (YFCC100M) dataset, a publicly available corpus of user-generated content comprising more than 100 million images and videos.

Vast collections of data fuel the Deep Learning engine

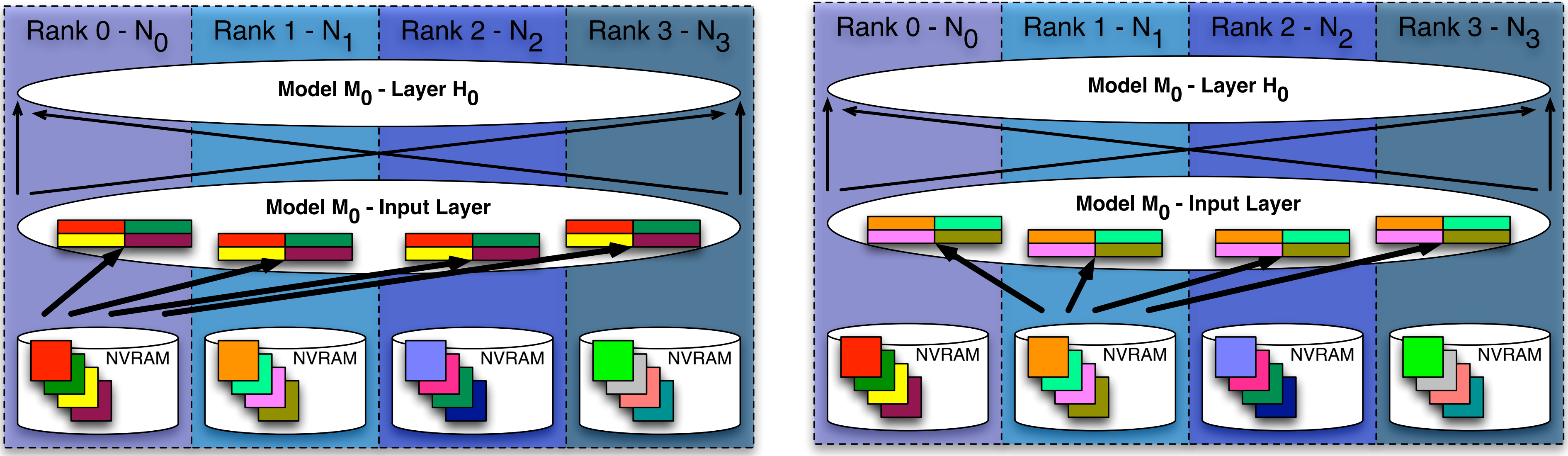
Distributing DNN across HPC nodes

- Each layer of model is distributed across nodes
  - Distributed matrix library (Elemental) provides dense matrix operations
- Input data is staged into node-local NVRAM
  - Each node stages a separate mini-batch



Distributing data

- Active mini-batch is replicated from source node to each MPI rank
- First layer multiplies distributed matrix with replicated input data



Experimental setup & Learning Task

- LLNL Catalyst HPC system (324 nodes)
  - 24 Xeon EP X5660 cores, 128 GB DRAM, and 800GB of node-local NVRAM
  - Aggregate bandwidth of 24-32 GB/s to a Lustre parallel file system
  - 48 Hyper-Threaded cores per node

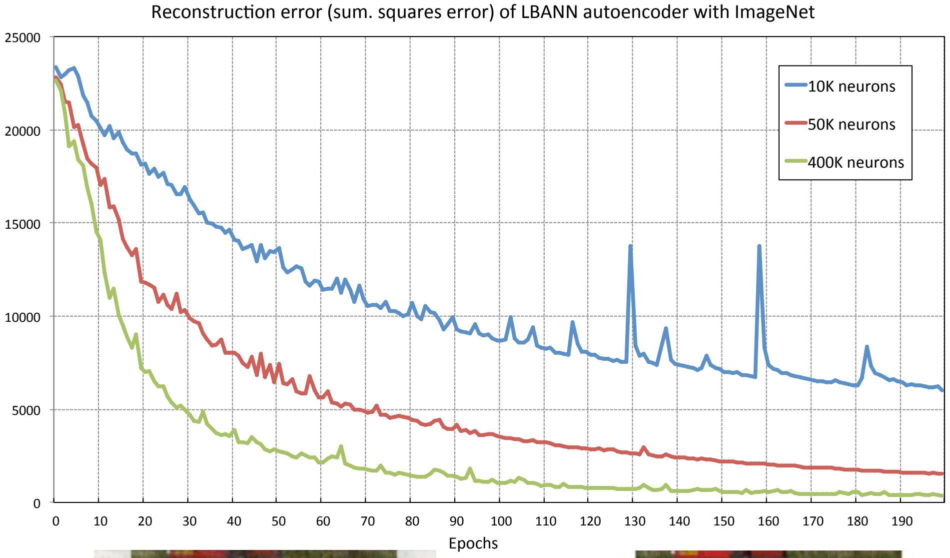
- ILSVRC2012 data set
  - Image size: $256 \times 256 \times 3 = 196{,}608$

- Neural network topology ~197K – X – ~197K,
  - X is the number of neurons in a fully connected hidden layer
  - Network sizes: 50K, 100K, 400K neurons
  - Matrix sizes: 9.8B, 19.7B, 78.6B parameters
  - Weight matrix sizes: 73GB, 147GB, 293GB (double FP)

- Software stack (C++)
  - Elemental distributed linear algebra library
  - MPI communication
  - Intel multi-threaded BLAS library

Training an auto-encoder

- Visualizing auto-encoder learning
  - Reconstruction cost
  - Reconstructing training image after 100 and 200 epochs



10K neurons
200 epochs

50K neurons
200 epochs

400K neurons
200 epochs

Original
image

Strong Scaling: Time per unit work (mini-batch)

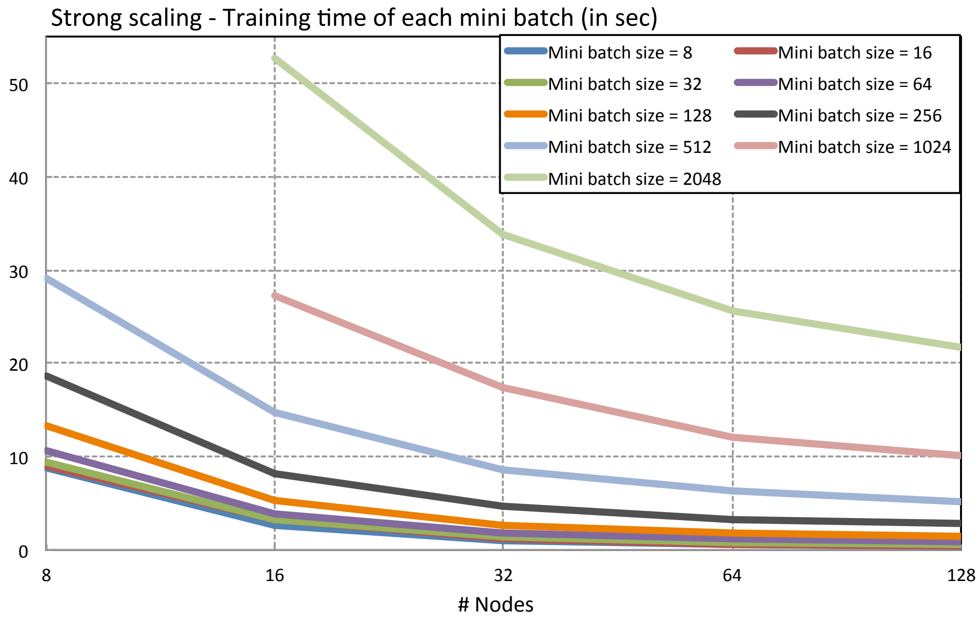
- # nodes versus mini-batch training time

- Processing multiple images per step
  - Reduces number of synchronizations per epoch
  - Computes a better gradient
  - Balancing # of steps versus quality of step

- Test
  - 50K neurons
  - 8 - 128 nodes, 12 ranks per node
  - mini-batch sizes from 8 - 2048 images

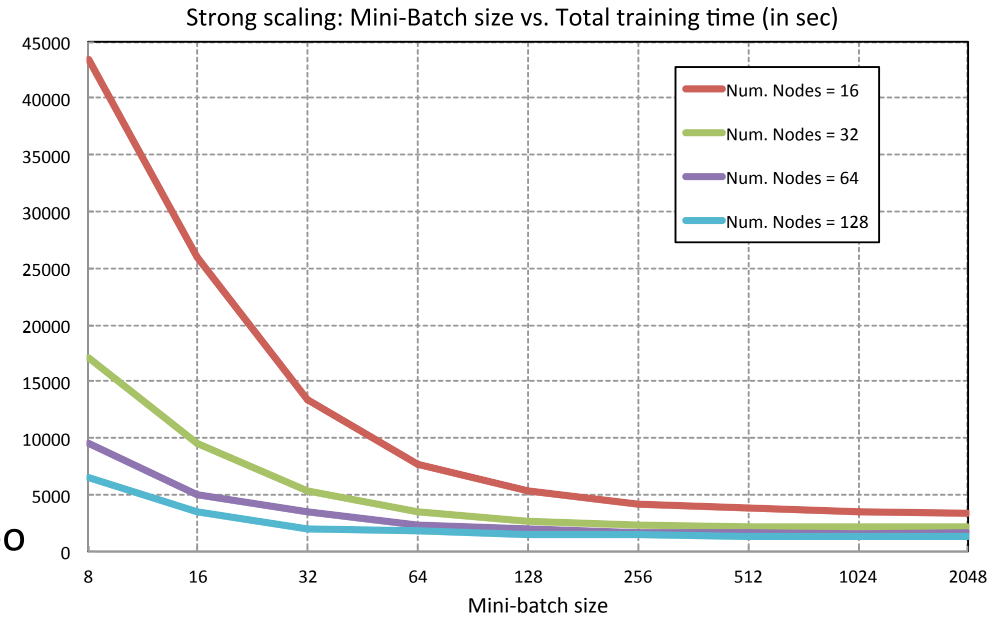
- Large mini-batches benefit greatly from additional nodes
  - Good scaling up to 64 nodes

- Smaller mini-batches have limited improvement beyond 16 or 32 nodes
  - Insufficient work to effectively amortize communication overheads

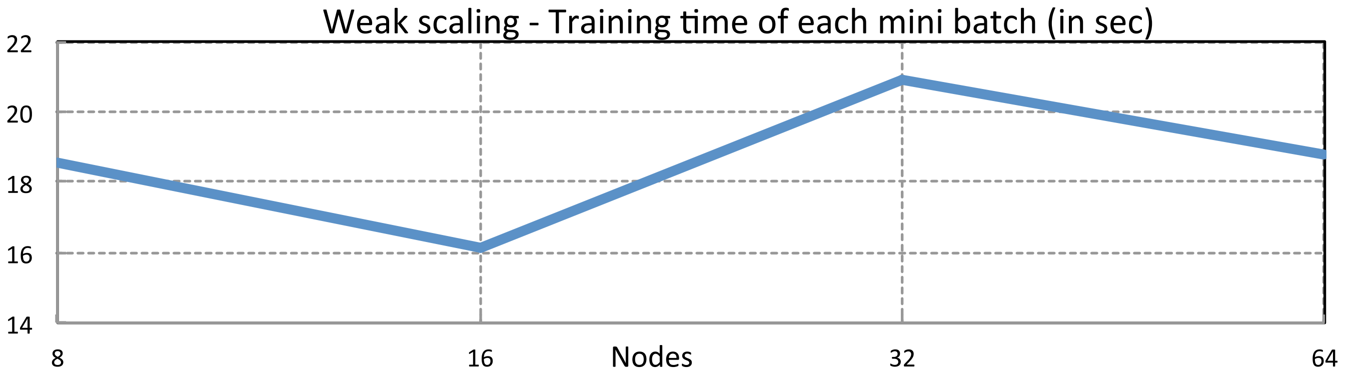


Strong Scaling: Total time for fixed amount of work

- Mini-batch size versus wall clock time
  - Fixed number of epochs
- Test
  - 50K neurons
  - 8 - 128 nodes
  - 12 ranks per node
  - mini-batch sizes from 8 - 2048 images
- Good scaling with smaller node counts
  - Diminishing returns for MB > 128
- On larger node counts problem size is too small to leverage resources
  - Too little work per node to offset communication overhead
  - Diminishing returns after 32 MB > 32



Weak Scaling: Increasing problem size and compute resources

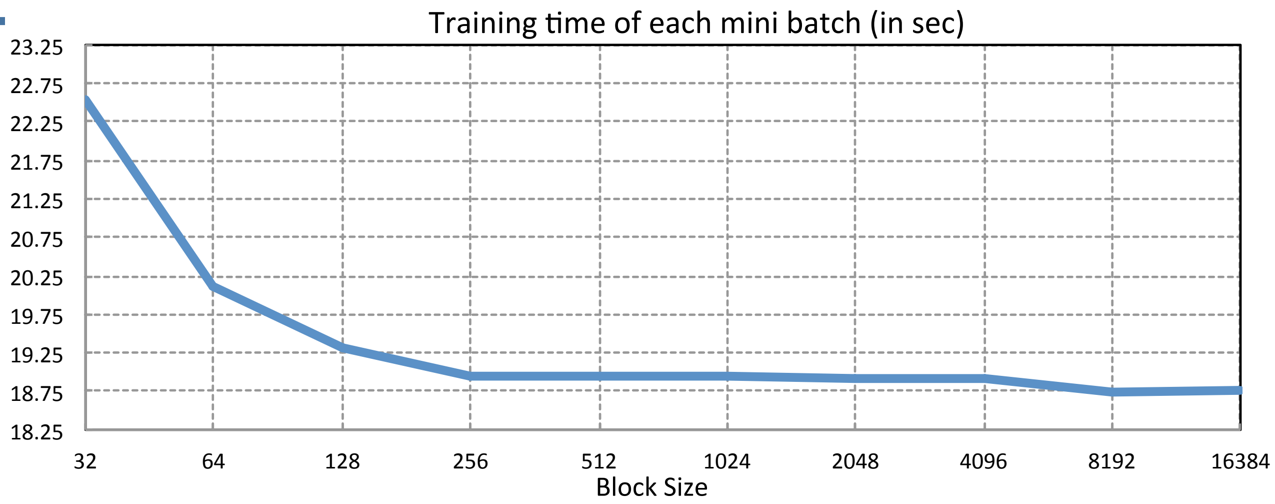


- Scaling the number of neurons from 50K to 400K
  - 8 nodes to 64 nodes, respectively
  - Mini-batch size is 256 images.

- Processing time of each mini-batch if fairly constant as problem size and available resources increase
  - ~10% variation in MB processing time

- Scaling up model sizes: matrices become more rectangular as # neurons increases
  - 2D partitioning scheme for data distribution

Tuning Elemental library algorithms

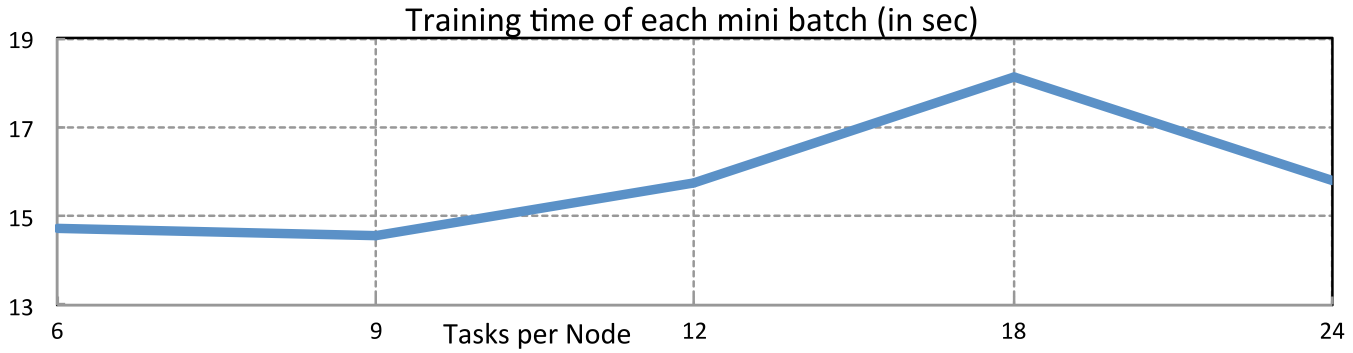


- Test
  - 400K neurons
  - 256 image mini-batch

- Data is distributed element-wise
  - Exploring new block distributed implementation in v0.86-git

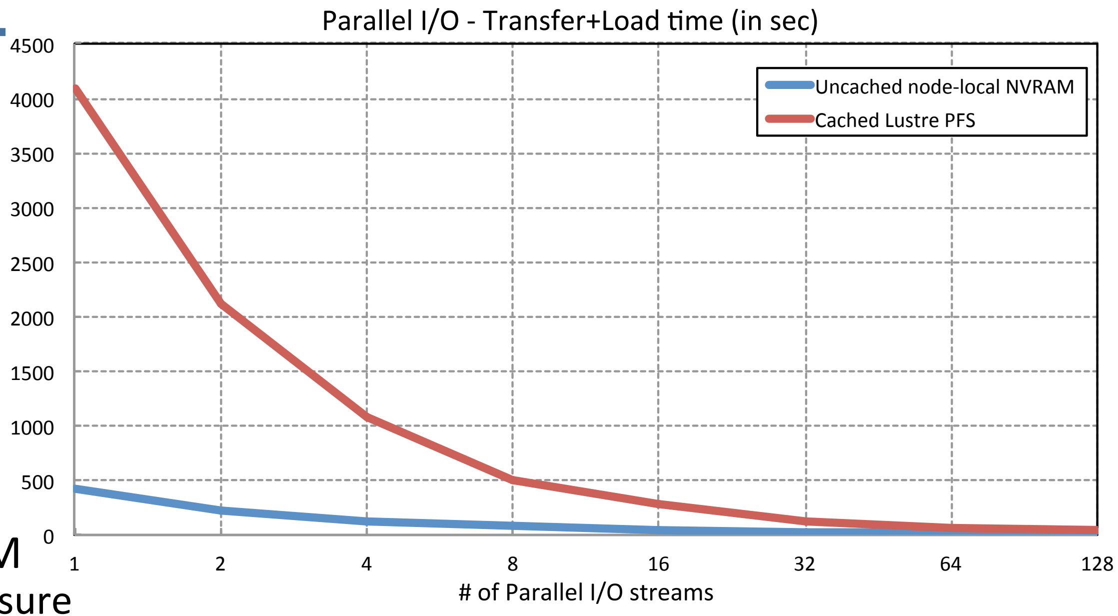
- Algorithmic block size affects operator implementation
  - Performance levels out once a sufficiently large block size is reached for local BLAS libraries
  - 19% performance difference between block size of 32 and 256+

Load-balancing distributed algorithm versus node-local math lib.



- Test
  - 400K neurons
  - 128 image mini-batch
- Balancing # of tasks versus # of threads per task
- Tuning the available resources to Intel BLAS library
  - BLAS library uses free cores for thread-parallel math operations
- 48 HyperThreaded cores per node
  - # of tasks should evenly divide # cores
  - 8 threads per task provided peak performance
  - 18 tasks per node is 19% worse than average training time

Local data staging & Parallelizing I/O



- Test
  - 50K neurons
  - 128 image mini-batch
  - 32 nodes
- Stage data to node-local NVRAM
  - Avoids additional memory pressure
  - ~12.9x faster than PFS with 128 I/O streams
  - Includes 18.56s overhead for copying and untar'ing the data from the Lustre PFS
  - Dovetails into future data augmentation techniques

Sufficient I/O parallelism significantly amortize data movement

Summary: LBANN is a work in progress

- LBANN toolkit is optimized for:
  - Unsupervised feature extraction
  - Data-intensive High Performance Computing systems
  - Large, distributed neural network models
  - Elemental library provides scalable, distributed linear algebra library

- Next Steps:
  - Convolutional, local receptive fields
  - Integrate GPU accelerators (including multiple per node)
  - Open source release
  - Explore training multiple models in parallel