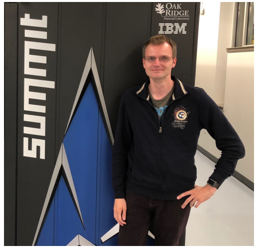
Exascale Deep Learning for Climate Analytics

Thorsten Kurth*, Sean Treichler, Joshua Romero, Mayur Mudigonda, Nathan Luehr, Everett Phillips, Ankur Mahesh, Michael Matheson, Jack Deslippe, Massimiliano Fatica, Prabhat, Michael Houston

ASCAC Meeting
12/12/18

The Team



Thorsten Kurth


Sean Treichler


Joshua Romero


Mayur Mudigonda


Nathan Luehr


Everett Phillips


Ankur Mahesh


Michael Matheson


Jack Deslippe


Massimiliano Fatica


Prabhat


Michael Houston

Relevance to DOE and ASCR

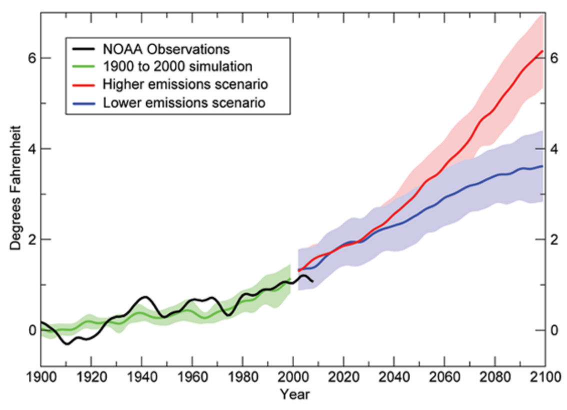
- why DOE/ASCR?
  - requires national lab compute facilities
  - requires domain specific knowledge in climate sciences
  - requires HPC knowledge only available in ASCR

- benefits to DOE
  - successful collaboration between national lab and hardware vendor
  - successful cross-lab effort
  - visibility and recognition helps talent acquisition

Understanding Climate

- How will the global weather develop by 2100?
  - will the globe warm up by 1.5 or 2.0 C?
  - will the sea level rise by 1 or 2 feet?

- How will extreme weather develop by 2100?
  - will there be more hurricanes?
  - will they become more intense?
  - will they make landfall more often?
  - will atmospheric rivers carry more water?
  - will they make landfall over California?
  - will they mitigate droughts?
  - will they cause heavy precipitation and flooding?

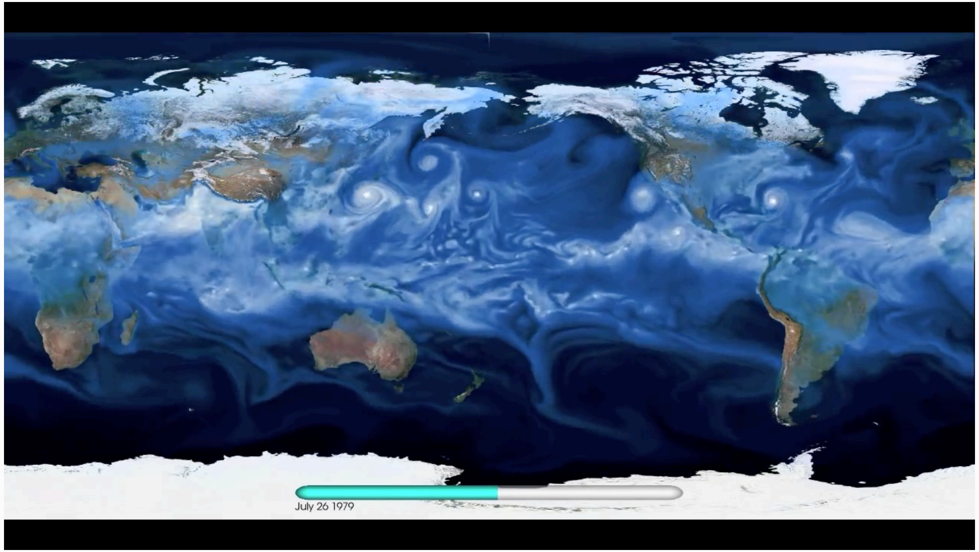


Impact Quantification of Extreme Weather Events

- automatically finding hurricanes and atmospheric rivers in climate model projections requires pixel-level segmentation

- enable extreme weather impact predictions to very high resolution

- gear up for future simulations with ~1 km^2 spatial resolution

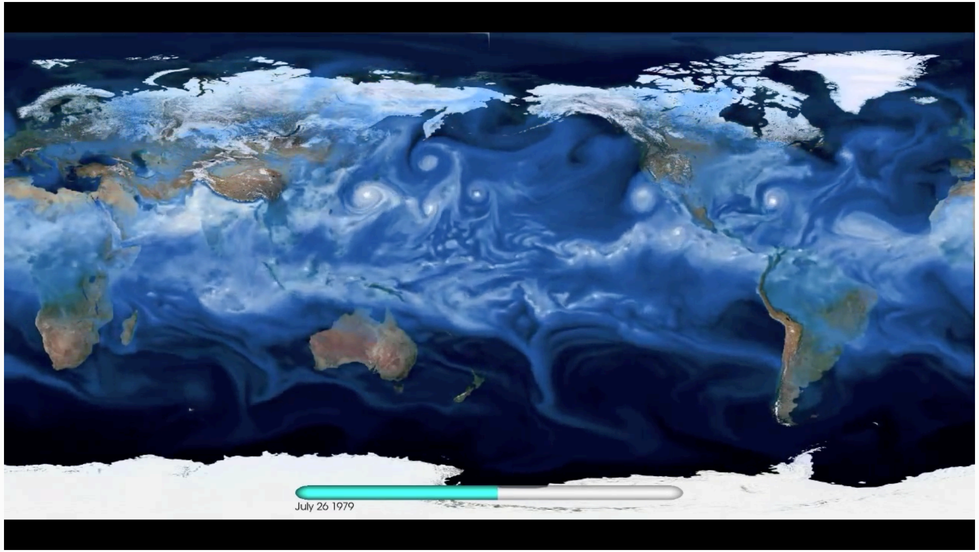


Impact Quantification of Extreme Weather Events

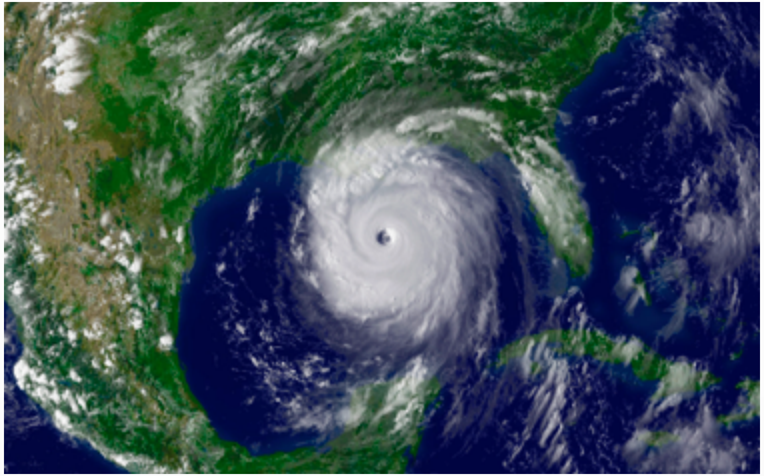
- automatically finding hurricanes and atmospheric rivers in climate model projections requires pixel-level segmentation

- enable extreme weather impact predictions to very high resolution

- gear up for future simulations with ~1 km^2 spatial resolution



Unique Challenges for Climate Analytics



- climate data is complex
    - many input channels
    - channels have very different properties
    - high resolution desired because each pixel occupies a large area of 25 km^2
    - no static *background*, highly variable in time and space
- interpret as segmentation problem
    - 3 classes - background (BG), Tropical Cyclones (TC), Atmospheric Rivers (AR)
    - high imbalance - most pixels are background (>95% on average)
    - high variance - shape of events change over time and in-between themselves

Deep Learning 101

- define neural network that computes predictions for arbitrary input: $\vec{y} = f(\vec{x}; \vec{w})$
- assemble a training set of sample inputs $\vec{x}_i$ and expected outputs $\vec{y}_i$
- define a loss function: $l(\vec{w}) = \frac{1}{N} \sum_{i=1}^{N} \left[\vec{y}_i - f\left(\vec{x}_i; \vec{w}\right)\right]^2$
- find the weights that minimize the loss: $\vec{w}^* = \operatorname{argmin} l(\vec{w})$
- typically solved using stochastic gradient descent (SGD):
  - $\tilde{g}^{(s)} = \frac{1}{B} \sum_{j=1}^{B} \nabla l_w \left(\vec{x}_{\pi_j}; \vec{w}^{(s)}\right)$
  - $\vec{w}^{(s+1)} = \vec{w}^{(s)} - \quad \cdot \tilde{g}^{(s)}$
  - iterate until converged

Unique Challenges for Deep Learning

- need labeled data for supervised approach
  - can leverage labels from existing heuristic-based approaches
- which neural network architecture to use?
  - balancing act between compute performance and model accuracy
  - employ high-productivity/flexible framework for rapid prototyping
  - performance optimization requires a holistic approach -- cannot focus on single set of kernels
- hyperparameter tuning (learning rate, regularization, etc.)
  - necessary for convergence and accuracy
  - finding hyperparameters which perform well at multiple concurrencies

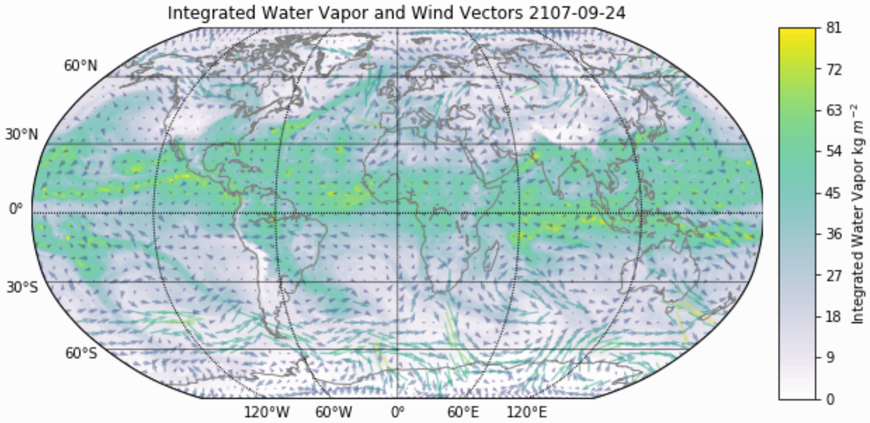
Unique Challenges for Extreme Scaling

- data management
  - shuffling/loading/preprocessing/feeding 20 TB dataset
  - feed data fast enough to keep GPUs busy

- multi-node synchronization
  - synchronous reductions of O(50) MB across 27,360 GPUs after every iteration

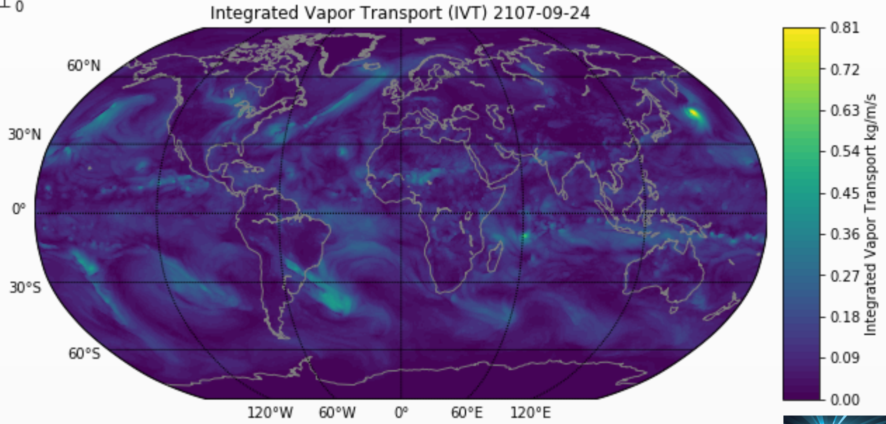
- convergence and accuracy at scale
  - mitigate typical batch-parallel training *generalization-gap* at large effective batch sizes

Atmospheric River Label Creation

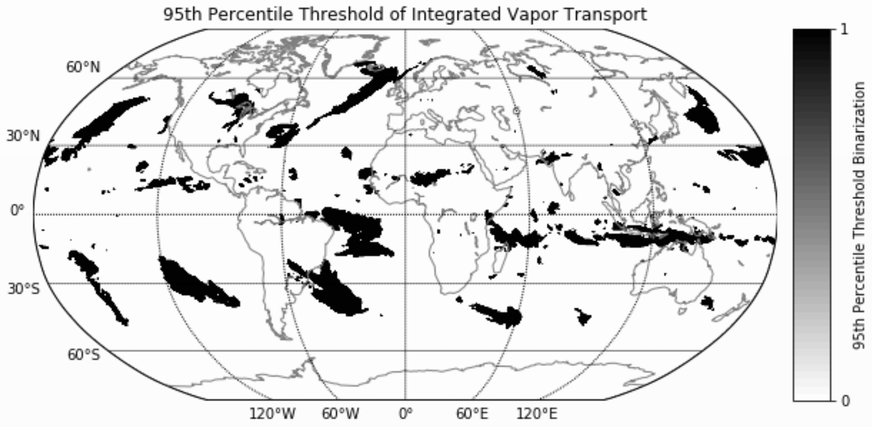


1. The climate model predicts levels of water vapor, wind, and specific humidity.

2. These fields are used to approximate the *transport* of water vapor

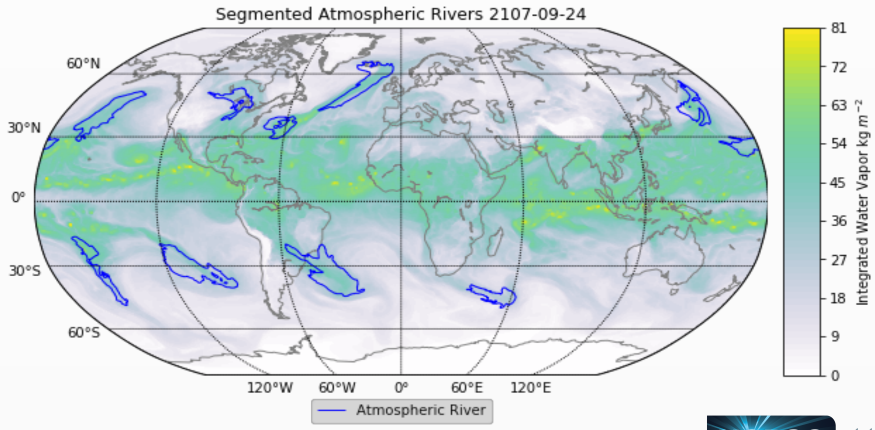


Atmospheric River Label Creation



3. Integrated Vapor Transport (IVT) is binarized at the 95th percentile.

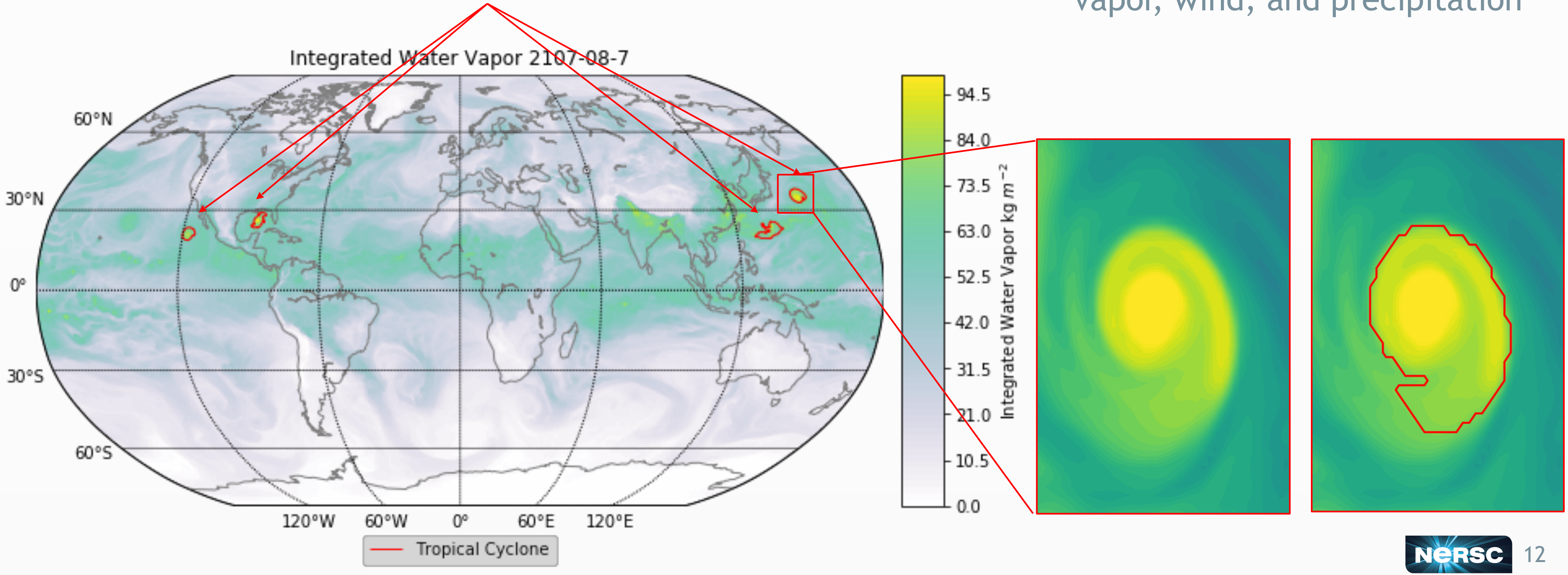
4. A flood fill algorithm is used to identify the atmospheric rivers: long, narrow regions of high IVT in the mid-latitudes



Tropical Cyclone Label Creation

1. Extract cyclone center and radius using thresholds for pressure, temperature, and vorticity

2. Binarize patch around cyclone center using thresholds for water vapor, wind, and precipitation



Software: TensorFlow and Horovod

- TensorFlow
  - high-productivity deep learning framework in Python with C++ backend, developed by Google
  - dataflow-style programming and asynchronous graph execution
  - makes use of optimized cuDNN library for performance sensitive kernels (e.g. convolutions)
  - provides features for building I/O input pipeline
  - can be combined with other Python modules to provide good flexibility

- Horovod
  - distributed-training enabling framework developed by Uber
  - provides MPI callback functions and convenience wrappers for TensorFlow
  - operates asynchronously with the TensorFlow dataflow scheduler

System 1: Piz Daint

- Cray XC50 HPC system at CSCS, Switzerland, ranked 5th in Top500 (Nov 2018)
- 5320 nodes with Intel Xeon E5-2695v3 and 1 Nvidia P100 GPU
- Cray Aries interconnect in diameter 5 dragonfly topology
- 54.4 PetaFlop/s peak performance (FP32)

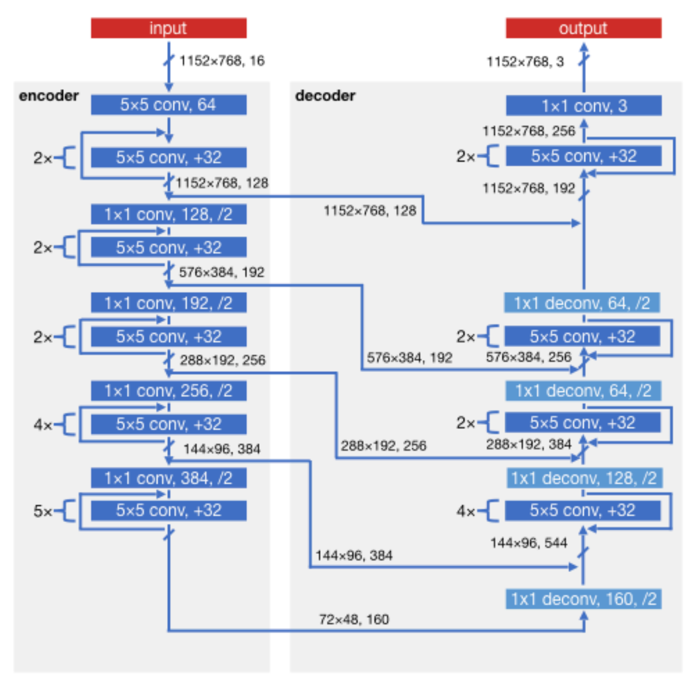
included to ensure and verify portability of our approach to other computing systems

System 2: Summit

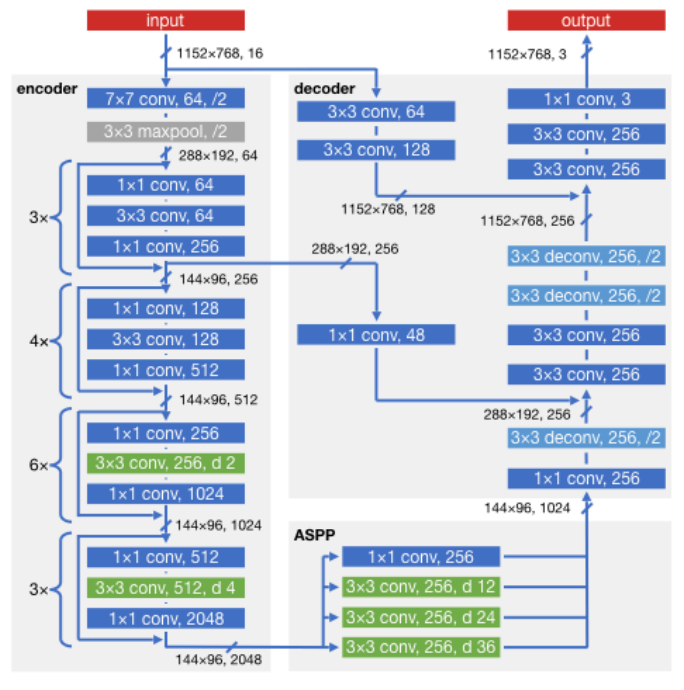
- leadership class HPC system at OLCF, ranked first on Top500 (Nov 2018)
- 4608 nodes with 2 IBM Power 9 CPU and 6 Nvidia Volta GPU with Tensor Cores
- 300 GB/s NVLink connection btw. 3 GPUs in a group
- 800 GB available NVMe storage/node
- dual-rail EDR Infiniband in fat-tree topology
- ~3.45 ExaFlop theoretical peak (FP16)

Our code stresses all above components of the system

Deep Learning Models for Extreme Weather Segmentation



Tiramisu, 35 layers,
7.8M parameters, 4.2 TF/sample

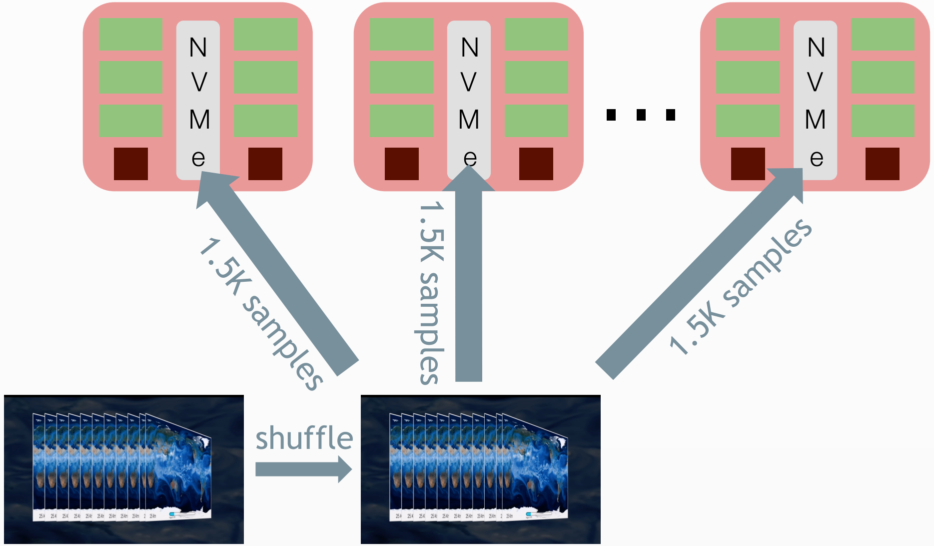


DeepLabv3+, 66 layers,
43.7M parameters, 14.4 TF/sample

NERSC

Data Staging

Dataset Size	Required BW (27K GPUs)	GPFS/LUSTRE	BurstBuffer	NVM/e or DRAM
20 TB (~63K samples)	3.8 TB/s	~400 GB/s	~2 TB/s	~26 TB/s



- 250 training samples/GPU (~15 GB), sample w/ replacement
- each file will be read at most once from FS
- files shared between nodes via MPI (`mpi4py`)

On-Node I/O Pipeline

- files are in HDF5 with single sample + label/file
- list of filenames passed to TensorFlow Dataset API (`tf.data`)
- HDF5 serialization bottleneck addressed with `multiprocessing + h5py`
- extract and batch using `tf.data` input pipeline

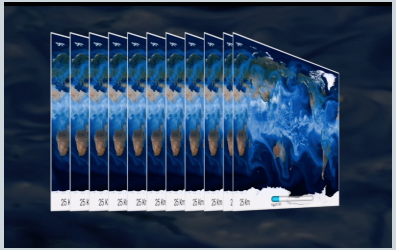
CPU

```
...
data-2107-12-26-02-4.h5
data-2107-12-26-03-1.h5
data-2107-12-26-03-4.h5
data-2107-12-26-04-1.h5
data-2107-12-26-04-4.h5
data-2107-12-26-05-1.h5
data-2107-12-26-05-4.h5
data-2107-12-26-06-1.h5
data-2107-12-26-06-4.h5
data-2107-12-26-07-1.h5
...
```

shuffle

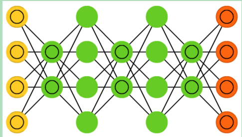
```
...
data-2107-03-03-06-1.h5
data-2107-05-24-00-4.h5
data-2107-08-30-03-4.h5
data-2107-10-29-01-4.h5
data-2107-12-11-07-1.h5
data-2107-08-14-03-4.h5
data-2107-01-08-01-4.h5
data-2107-09-08-04-1.h5
data-2107-09-22-00-1.h5
data-2107-07-16-03-4.h5
...
```

4-way parallel
read + preprocess



batch

GPU



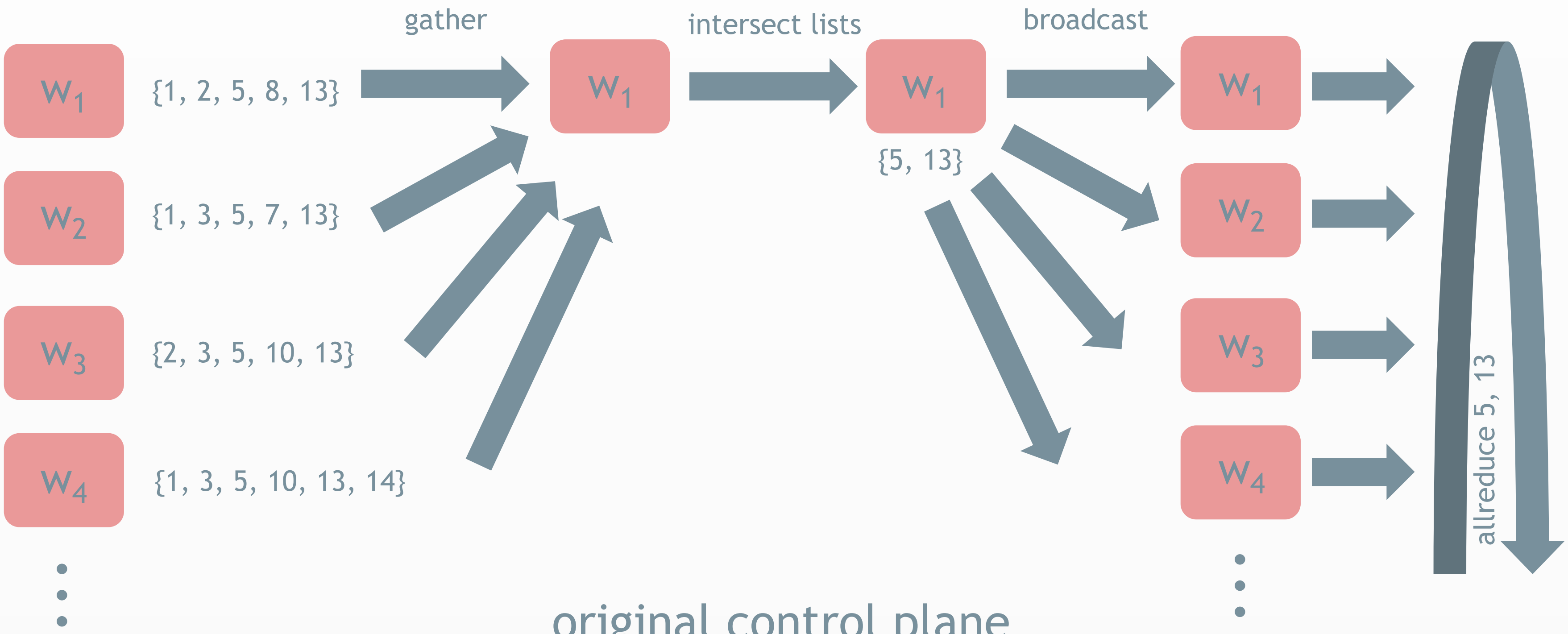
asimovinstitute.org/neural-network-zoo

Single Node Performance

- GPU execution profiled with CUDA profiler, kernels grouped by category
- convolution kernels: use latest cuDNN, favor higher computational intensity
- pay attention to memory layout to reduce transposes and copies
- tuning of input pipeline on CPU to keep off critical path

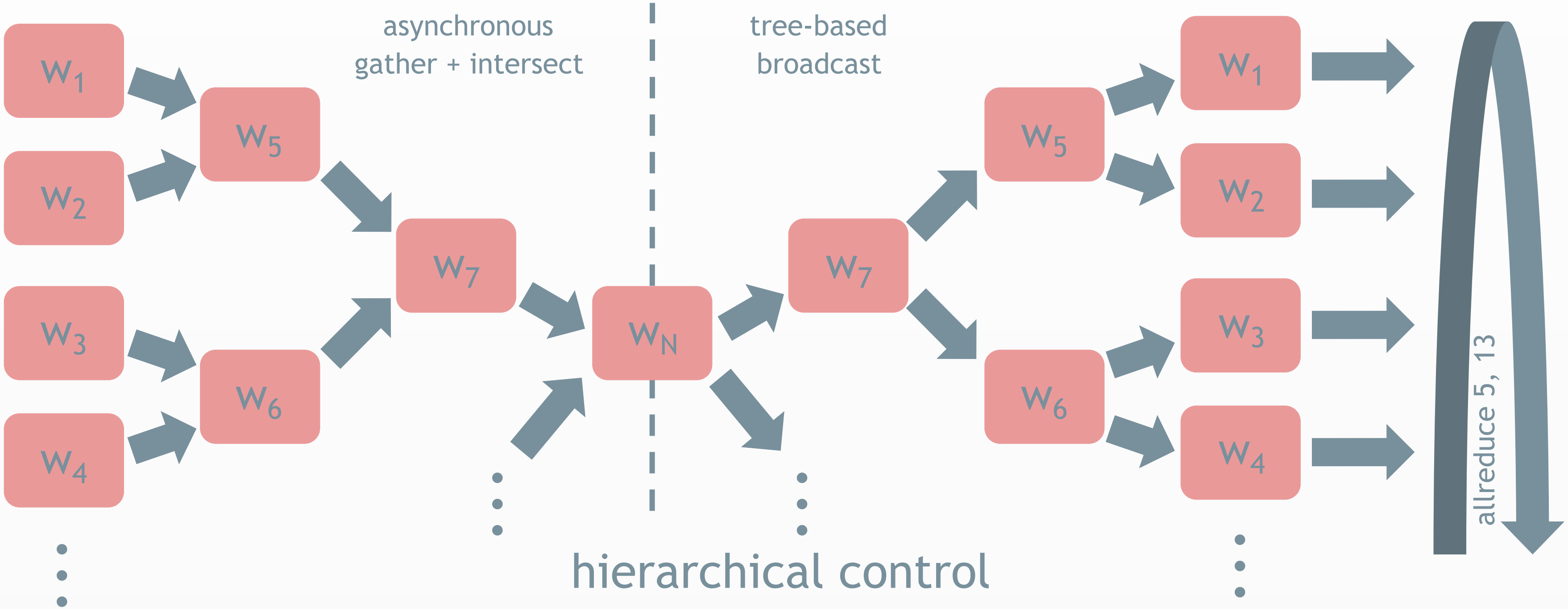
Category		DeepLabv3+ FP16 Training						
		# Kern	Time (ms)	Math (TF)	Mem (GB)	% Time	% Math	% Mem
Forward	Convolutions	158	147.9	9.61	27.6	18.1	52.0	20.7
	Point-wise	829	52.3	< 0.1	24.3	6.4		51.6
Backward	Convolutions	195	300.2	19.21	50.5	36.7	51.2	18.7
	Point-wise	157	25.6	< 0.1	6.3	3.1		27.3
Optimizer		1219	3.9	< 0.1	1.1	0.5		31.3
Copies / Transposes		708	213.2	-	92.6	26.1		48.3
Allreduce (NCCL)		30	58.7	< 0.1	0.6	7.2		1.1
Type Conversions		201	1.3	-	0.6	0.2		51.3
GPU Idle			14.2			1.7		
Total		3497	817.3	28.82	203.6		28.2	27.7

Horovod Control Plane Optimizations



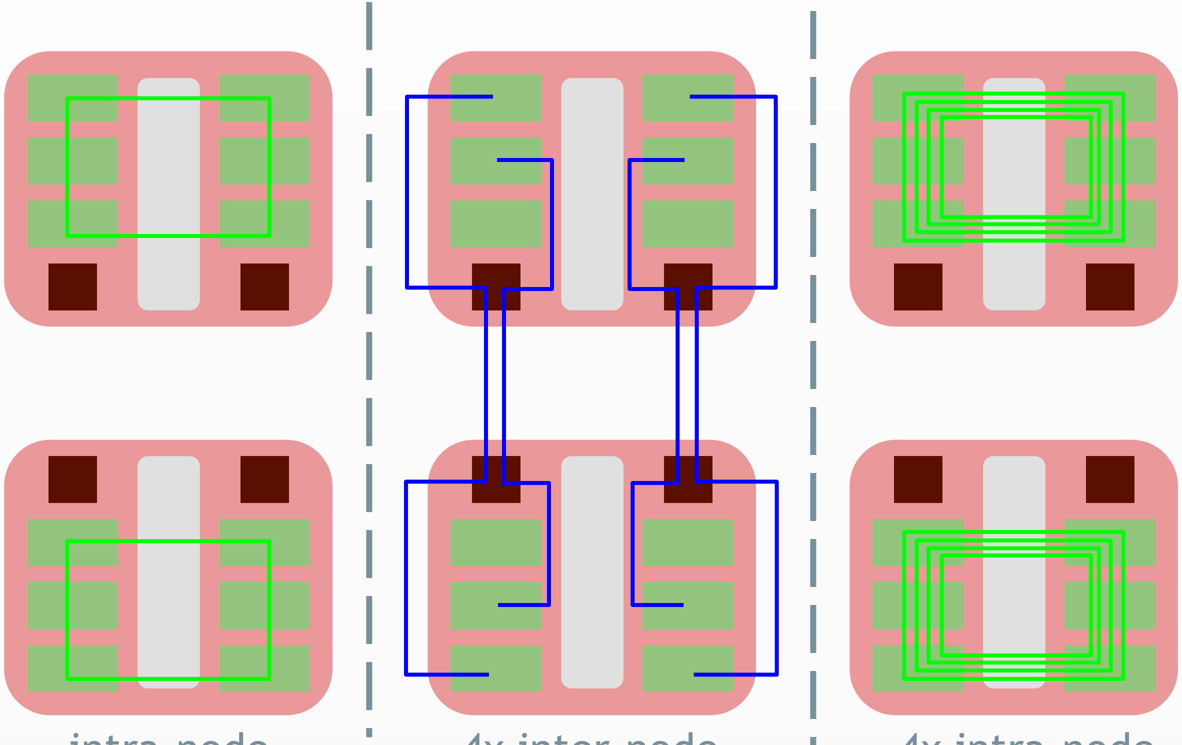
original control plane

Horovod Control Plane Optimizations



hierarchical control plane

Hybrid All-Reduce



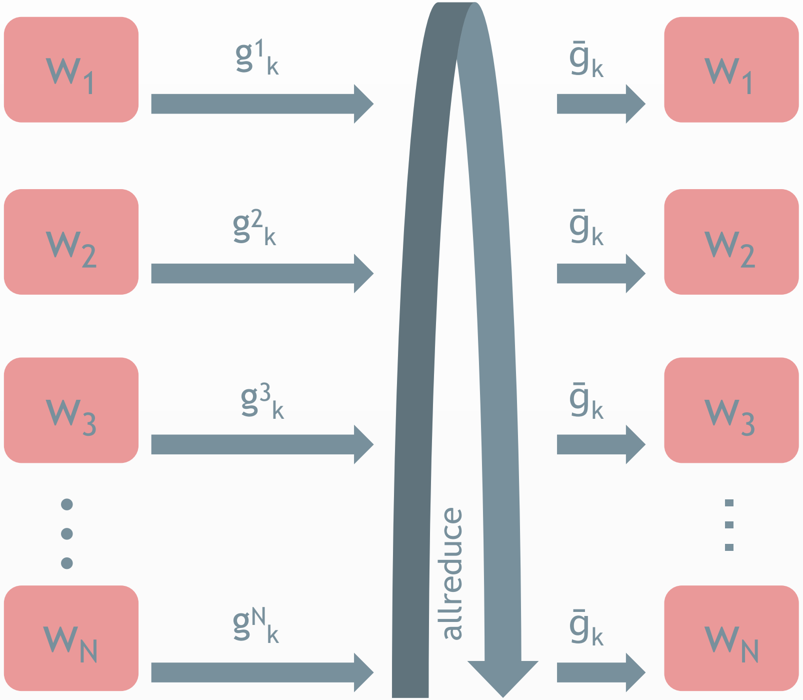
intra-node allreduce (NCCL)

4x inter-node allreduce (MPI)

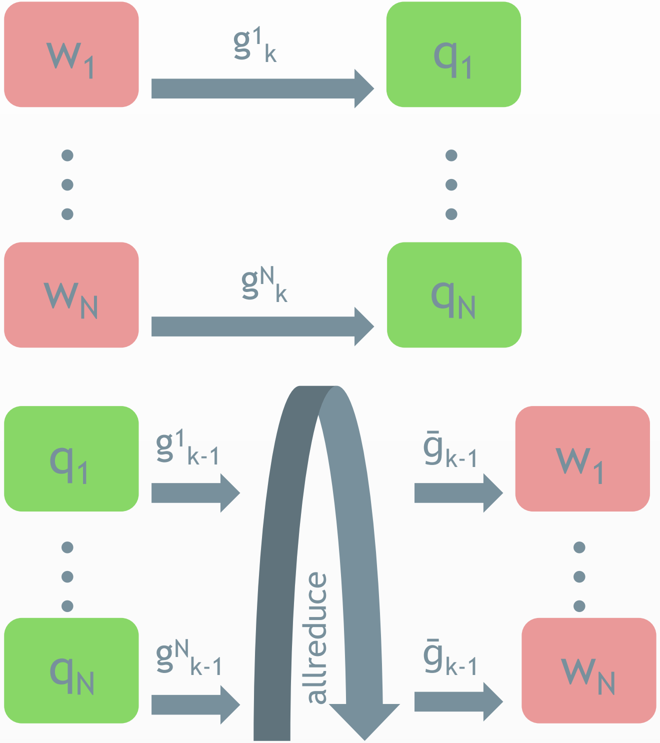
4x intra-node broadcast (NCCL)

- NCCL uses NVLink for high throughput, but ring-based algorithms latency-limited at scale
- hybrid NCCL/MPI strategy uses strengths of both
- one inter-node allreduce per virtual NIC
- MPI work overlaps well with GPU computation

Gradient Pipelining (Lag)

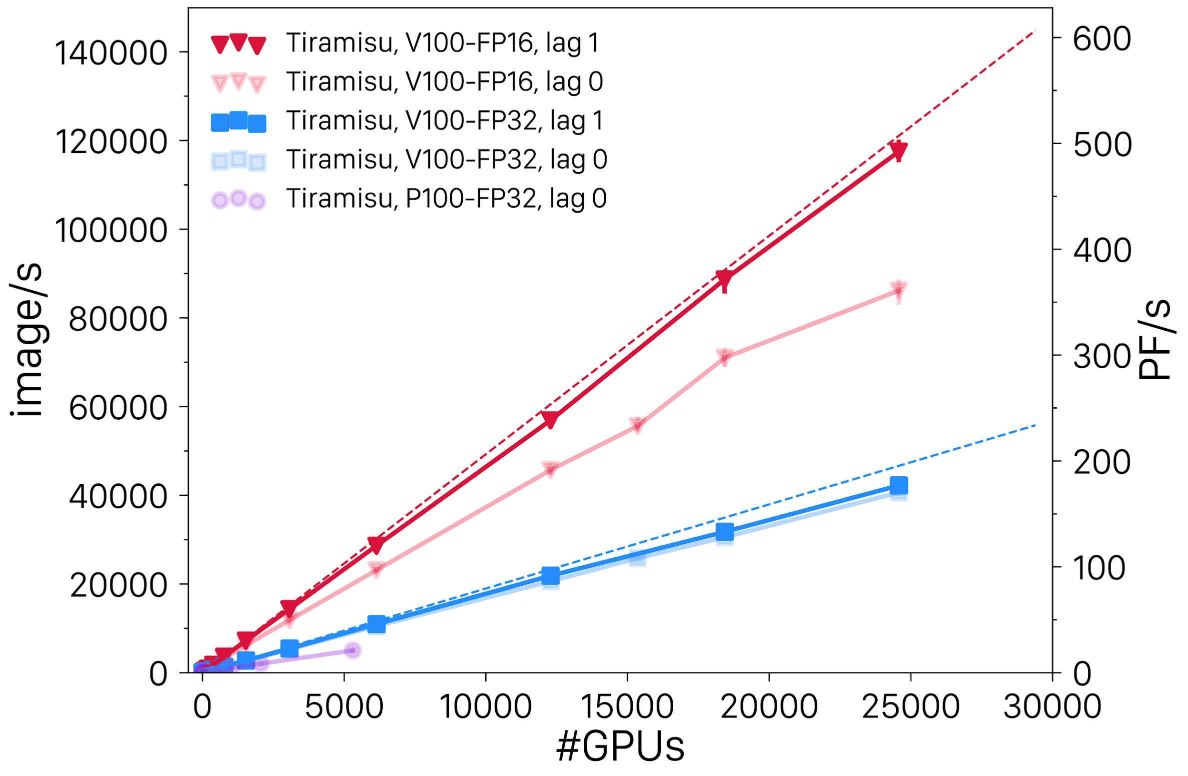


lag-0 (fully synchronous)



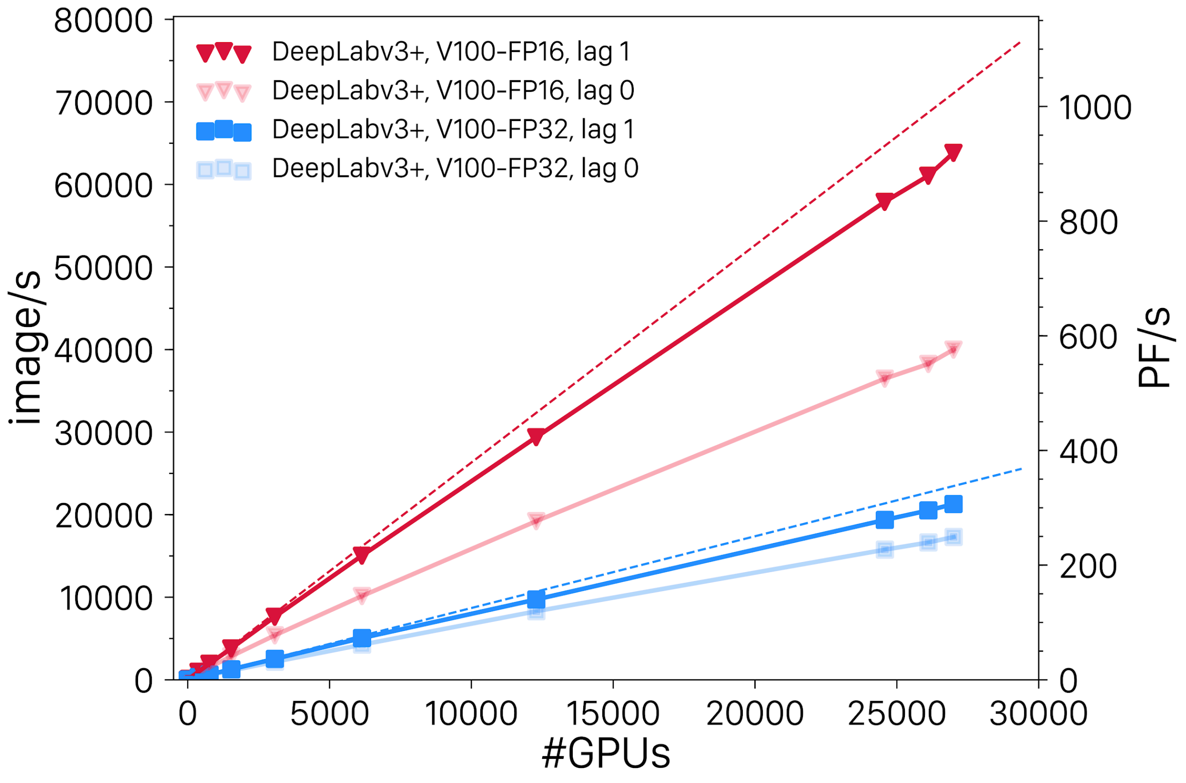
lag-1

Scaling Tiramisu



- FP16-model sensitive to communication
- FP16-model BW-bound (only 2.5x faster than FP32)
- almost ideal scaling for both precisions on Summit when gradient lag is used

Scaling DeepLabv3+



- FP16-model sensitive to communication
- FP16-model BW-bound (only 2.5x faster than FP32)
- excellent scaling for both precisions on Summit when gradient lag is used

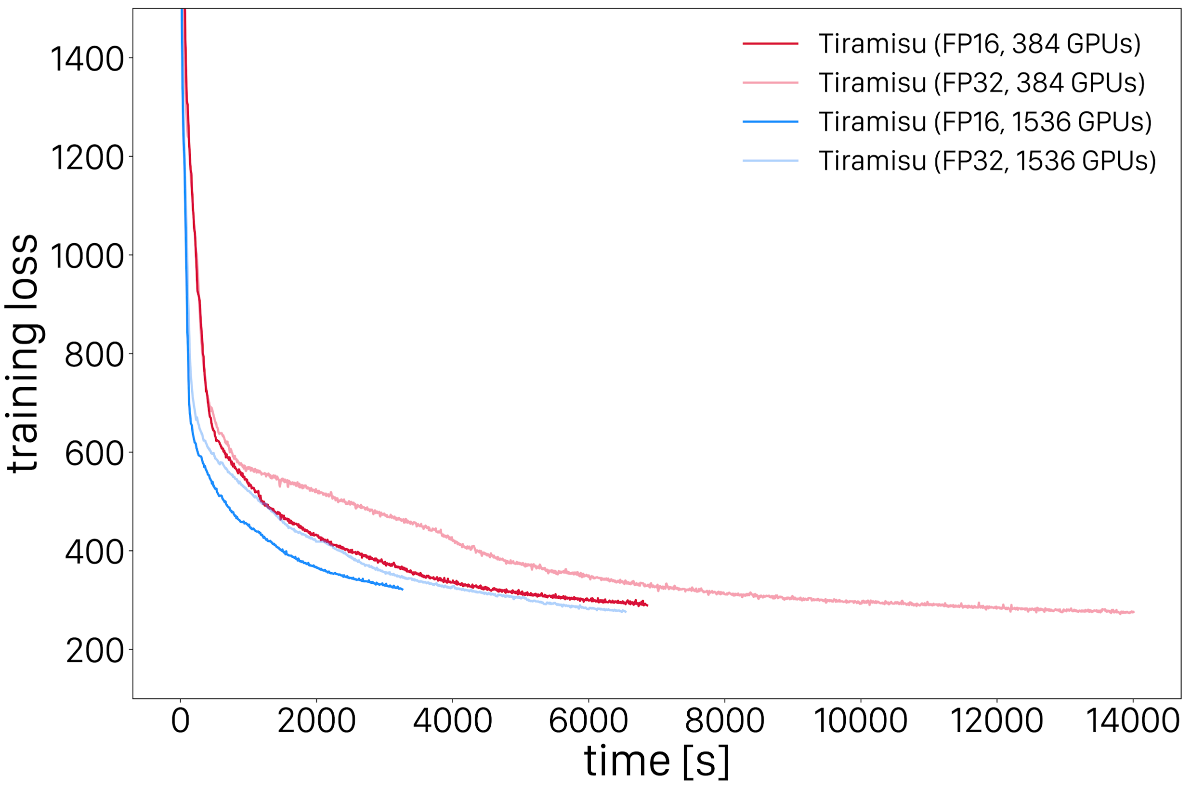
999 PetaFlop/s (FP16) sustained

DeepLabv3+, 4560 nodes (27360 GPU)

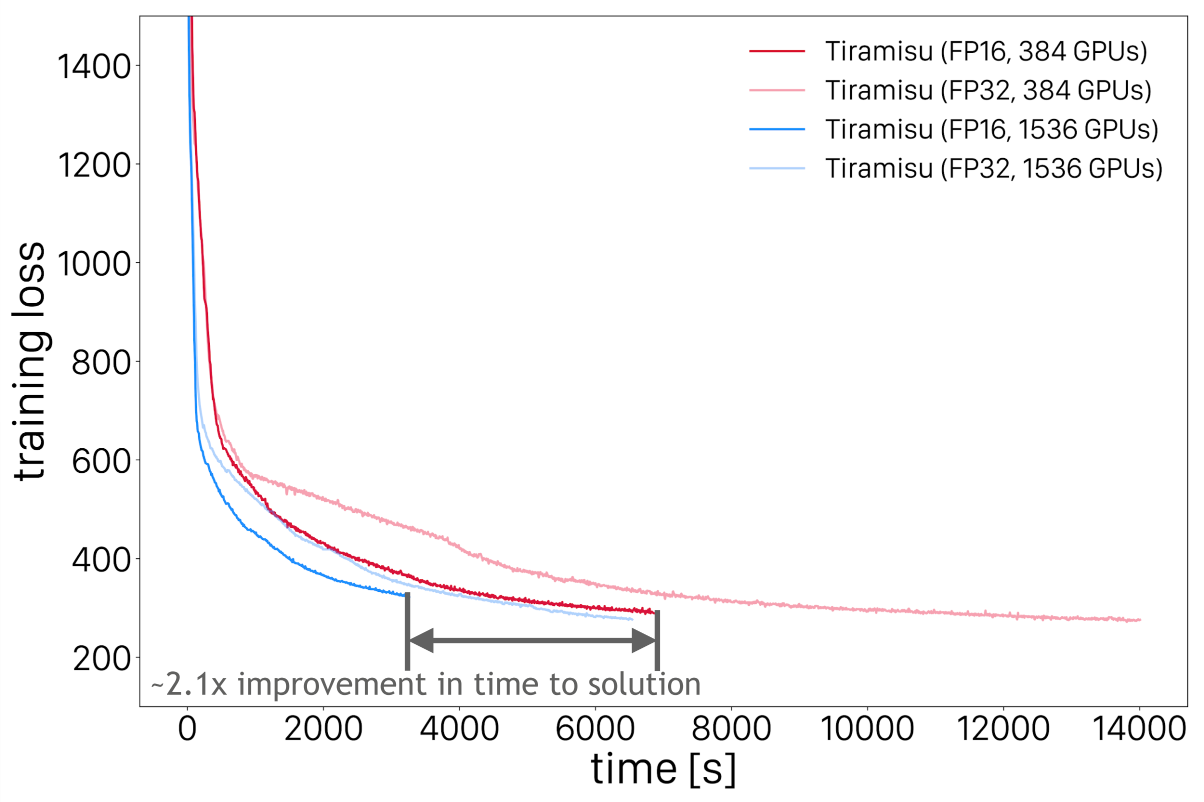
1.13 ExaFlop/s (FP16) peak

DeepLabv3+, 4560 nodes (27360 GPU)

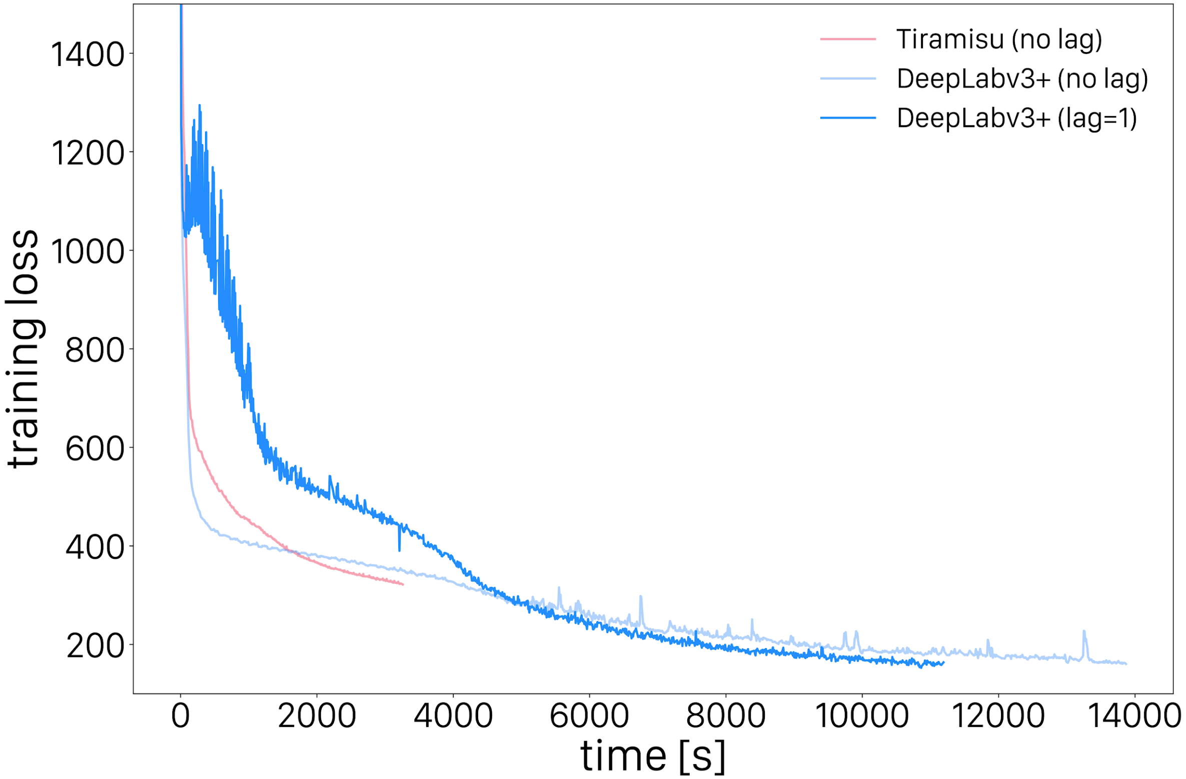
Concurrency/Precision and Convergence



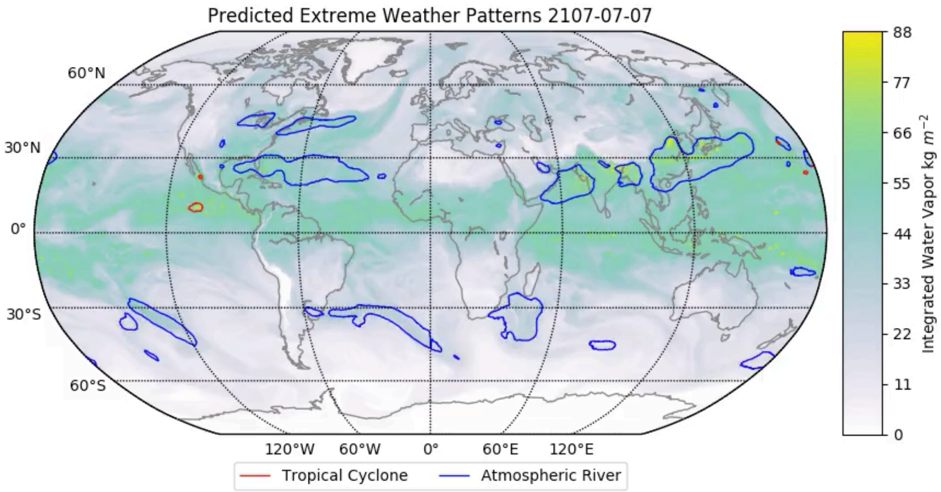
Concurrency/Precision and Convergence

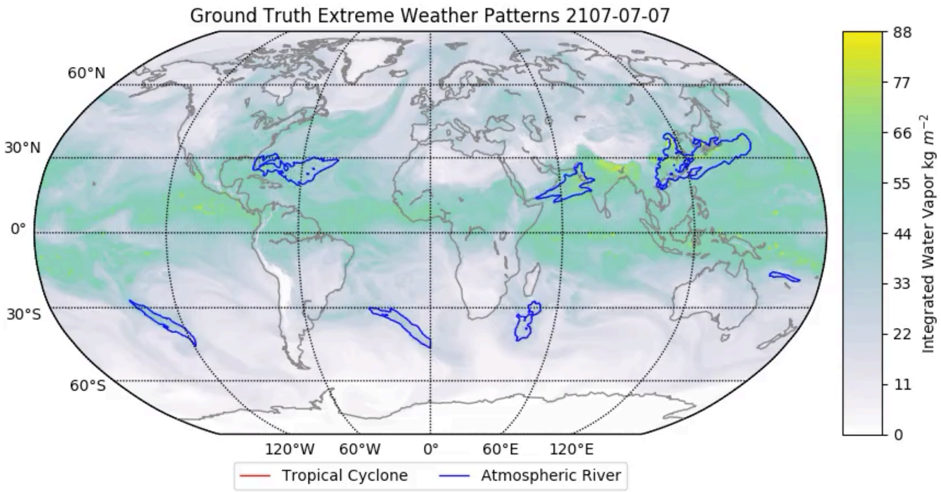


Model/Lag and Convergence



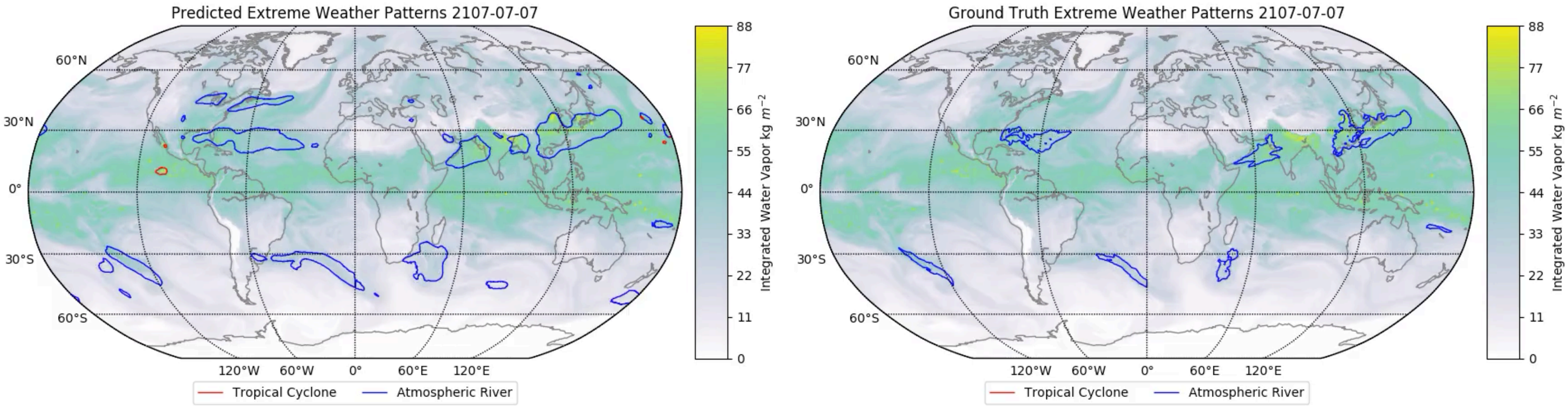
Segmentation Animation





- best result for intersection-over-union (IoU) obtained: ~73%
- result at large scale (batch-size > 1500): IoU~55%

Segmentation Animation



- best result for intersection-over-union (IoU) obtained: ~73%
- result at large scale (batch-size > 1500): IoU~55%

Conclusions

- deep learning and HPC converge, achieving *exascale* performance
- demonstrated that compute capabilities at LCF facilities can be utilized to tackle difficult scientific deep learning problems
- software enhancements benefit deep learning community, in- and outside DOE
- deep learning-powered techniques usher in a new era of precision analytics for various science areas

Acknowledgements

- National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE- AC02-05CH11231
- Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory, supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725
- Swiss National Supercomputing Center (CSCS), project g107
- we further thank OLCF and CSCS staff, especially Nicholas Cardo, Veronica Melesse Vergara, Don Maxwell, Matthew Ezell for technical as well as Arjun Shankar, Ashley Barker, Tjerk Straatsma and Jack Wells for programmatic support

Thank You