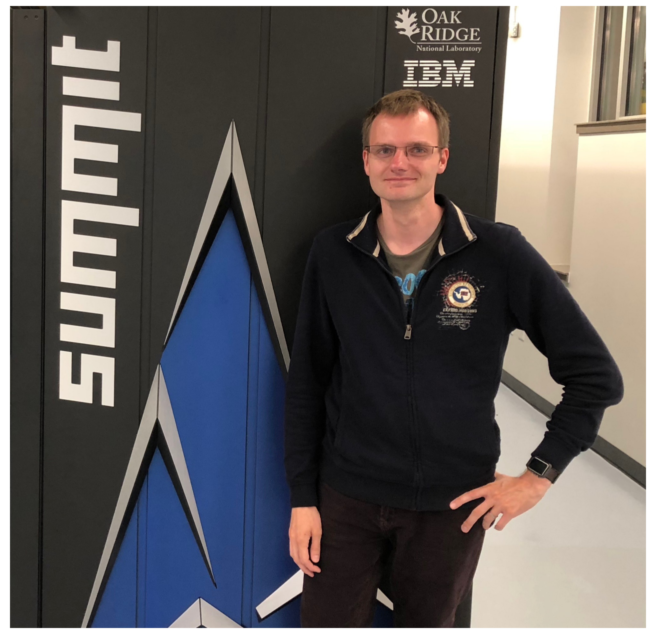
Exascale Deep Learning for Climate Analytics

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

GPUs for Science Day
07/02/2019, Berkeley, CA

The Team



Thorsten Kurth

Sean Treichler

Joshua Romero

Mayur Mudigonda

Nathan Luehr

Everett Phillips

Ankur Mahesh

Michael Matheson

Jack Deslippe

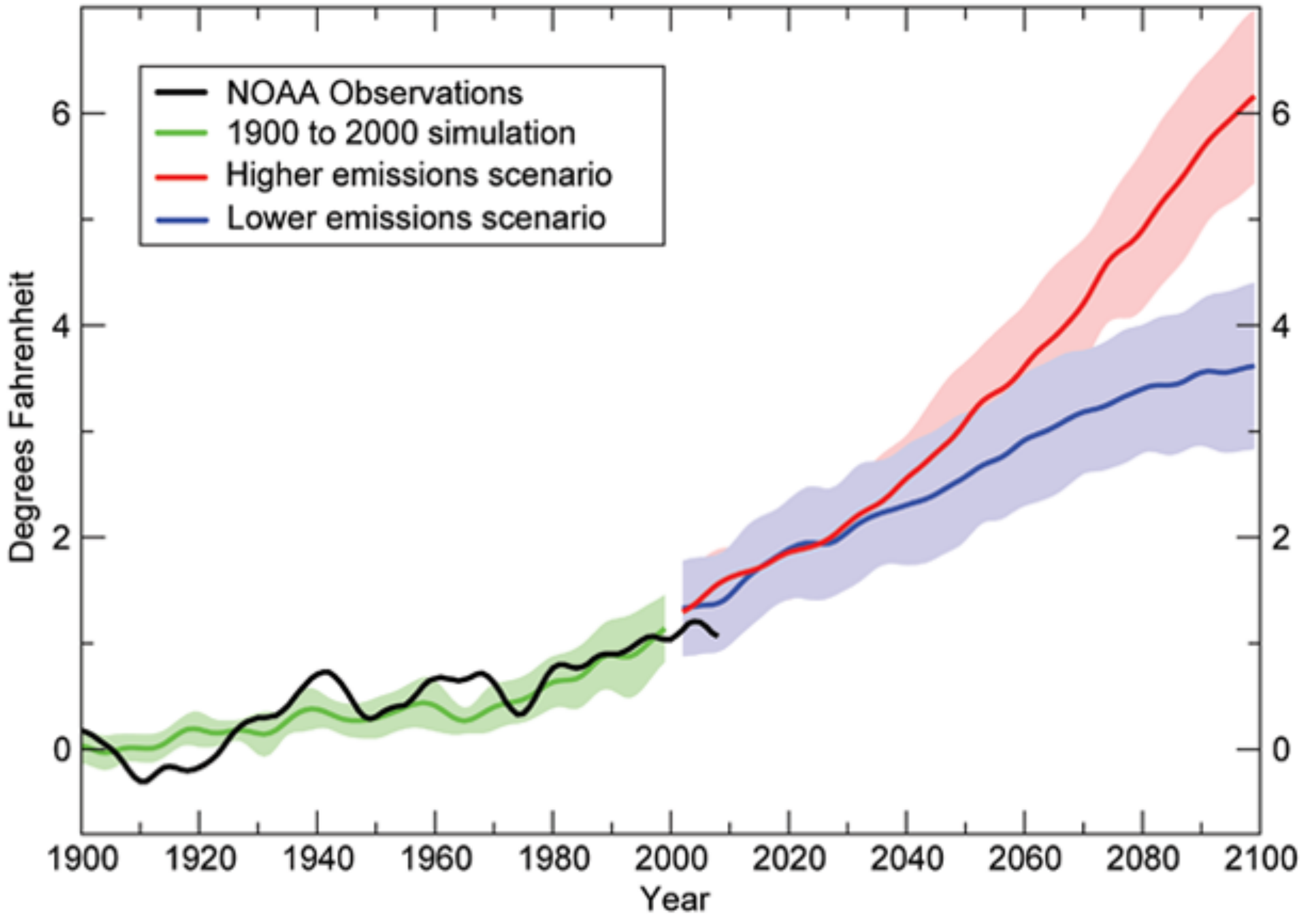
Massimiliano Fatica

Prabhat

Michael Houston

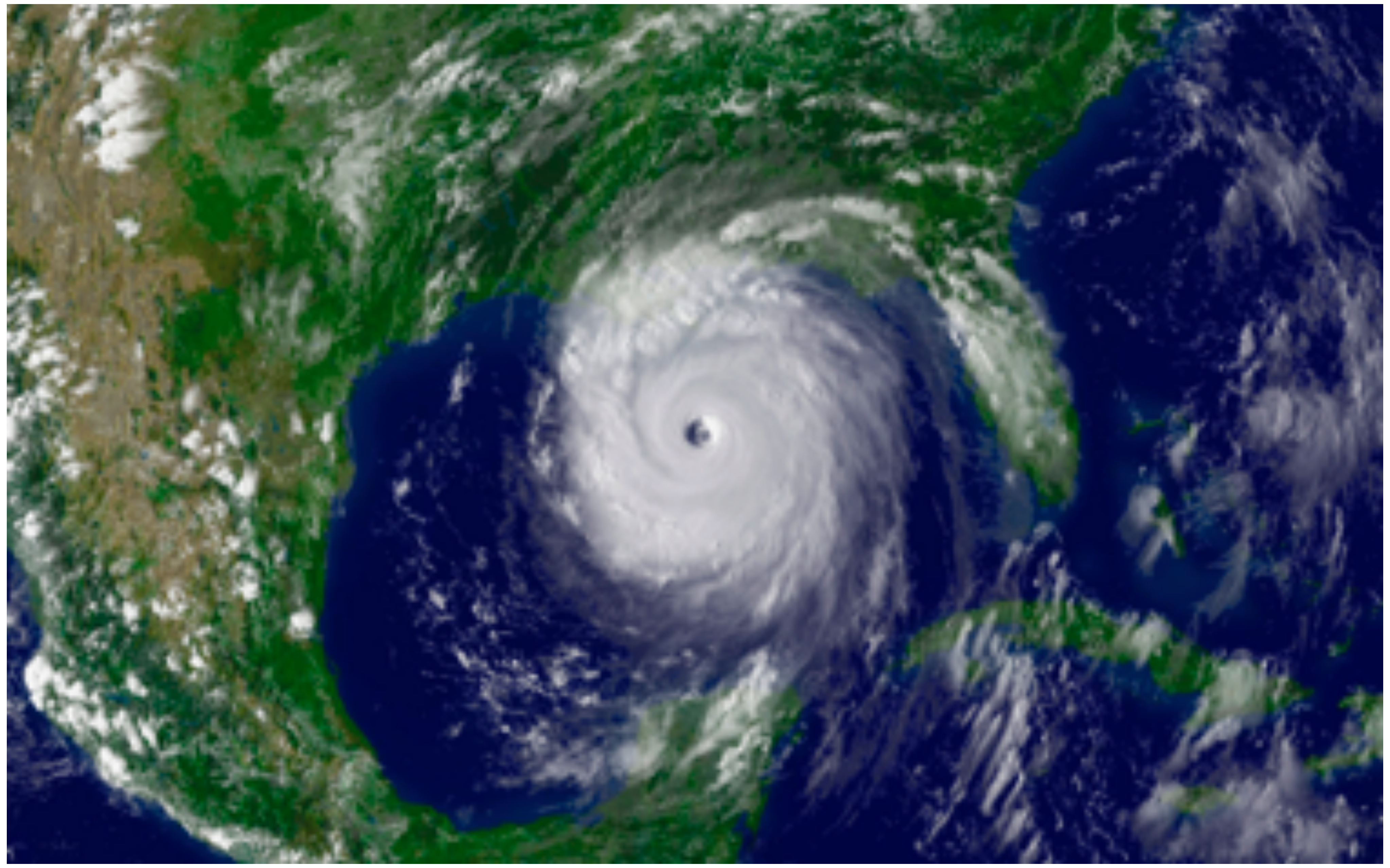
Understanding Climate Change

- How will the global weather develop by 2010?
  - 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 be more intense?
  - will they make landfall more often?
  - will atmospheric rivers carry more water?
  - can they help mitigate droughts
  - will they cause flooding and heavy precipitation?



Unique Challenges for Climate Analytics

- interpret as segmentation problem
  - 3 classes - background (BG), tropical cyclones (TC), atmospheric rivers (AR)
- climate data is complex
  - high imbalance - more than 95% of pixels are background
  - high variance - shape of events change
  - many input channels w/ different properties
  - high resolution required
  - no static *background*, highly variable in space and time
- Deep Learning has proven successful for these tasks



Unique Challenges for Deep Learning at Extreme Scale

- need labeled data (supervised approach): leverage from heuristic-based approaches
- define neural network architecture: good balance between compute and model performance, rapid prototyping capabilities essential
- data management: shuffling/loading/processing/feeding 20 TB dataset to keep GPUs busy
- multi-node synchronization: synchronous reduction of O(50)MB across 27360 GPUs after each iteration
- convergence and accuracy at scale
- hyper parameter tuning (HPO)

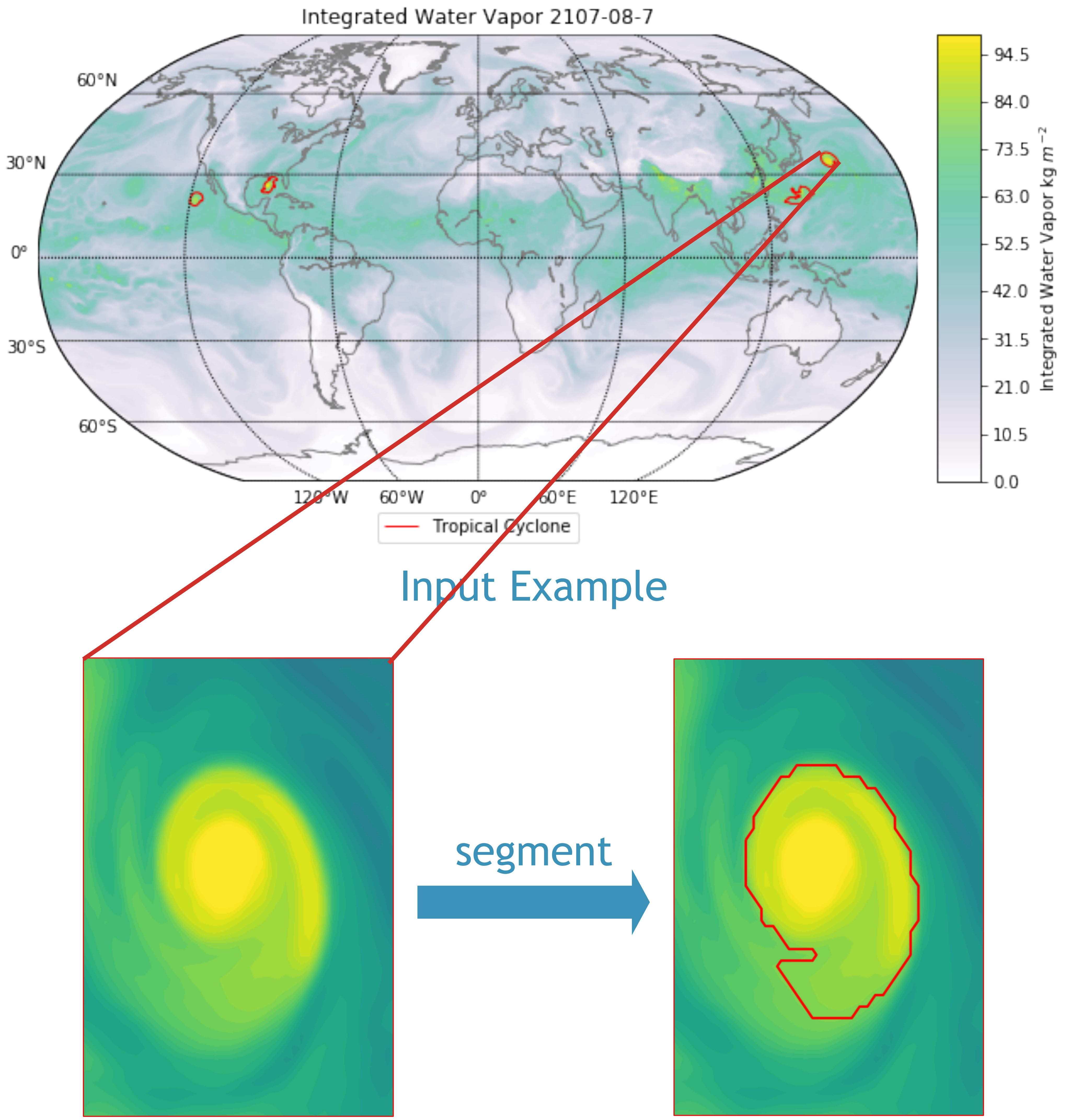
Software: TensorFlow and Horovod

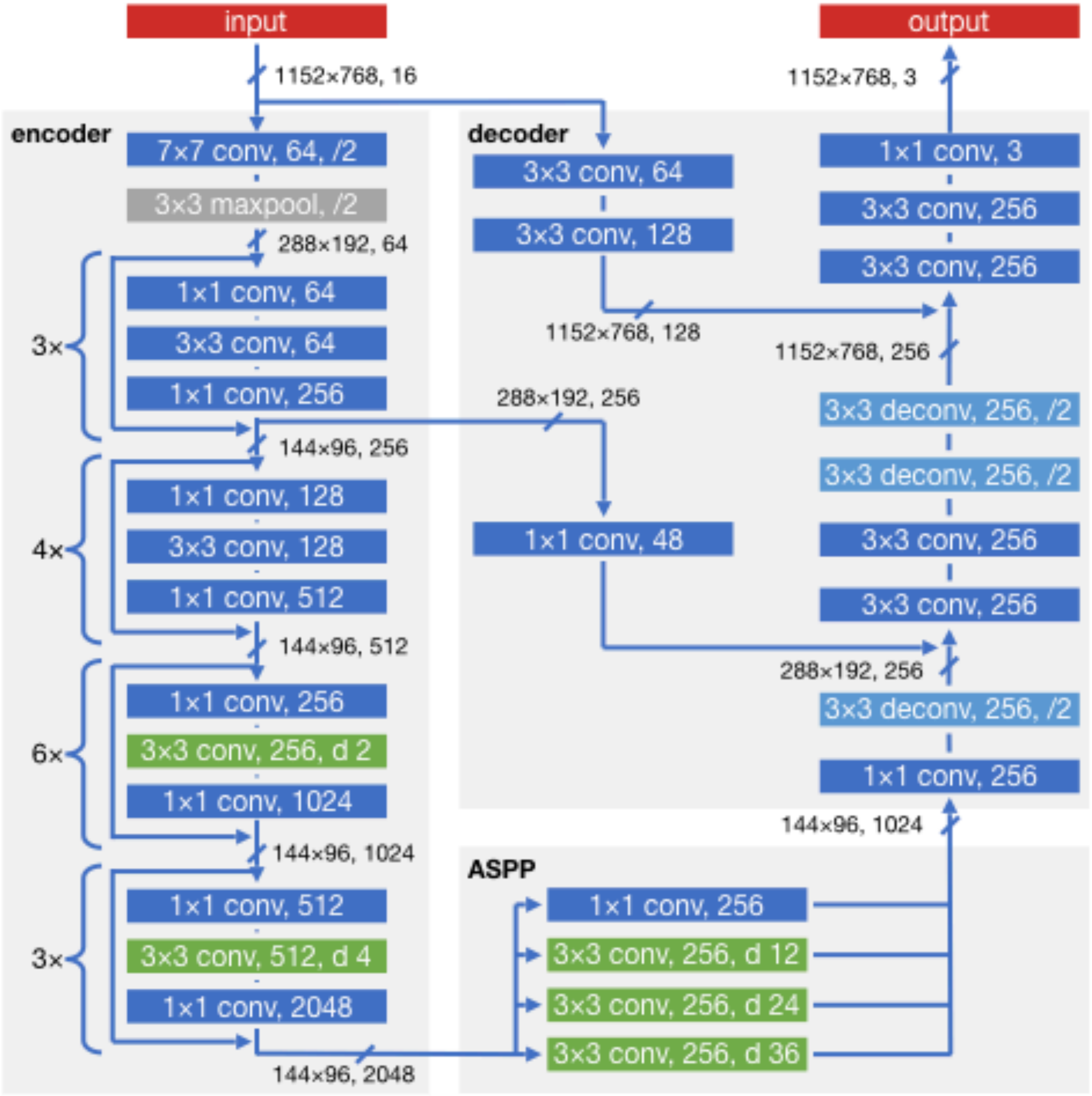
- TensorFlow
  - high-productivity deep learning framework in Python with C++-backend, developed by Google
  - makes use of optimized cuDNN library for performance sensitive kernels (e.g. convolutions)
  - dataflow-style programming and asynchronous graph execution
  - provides features for building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

Summit

- leadership class HPC system at OLCF, 1st on top500
- 4609 nodes with 2 IBM P9 CPU and 6 NVIDIA V100 GPU
- 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/s theoretical peak performance (FP16)

Deep Learning Model for Extreme Weather Segmentation

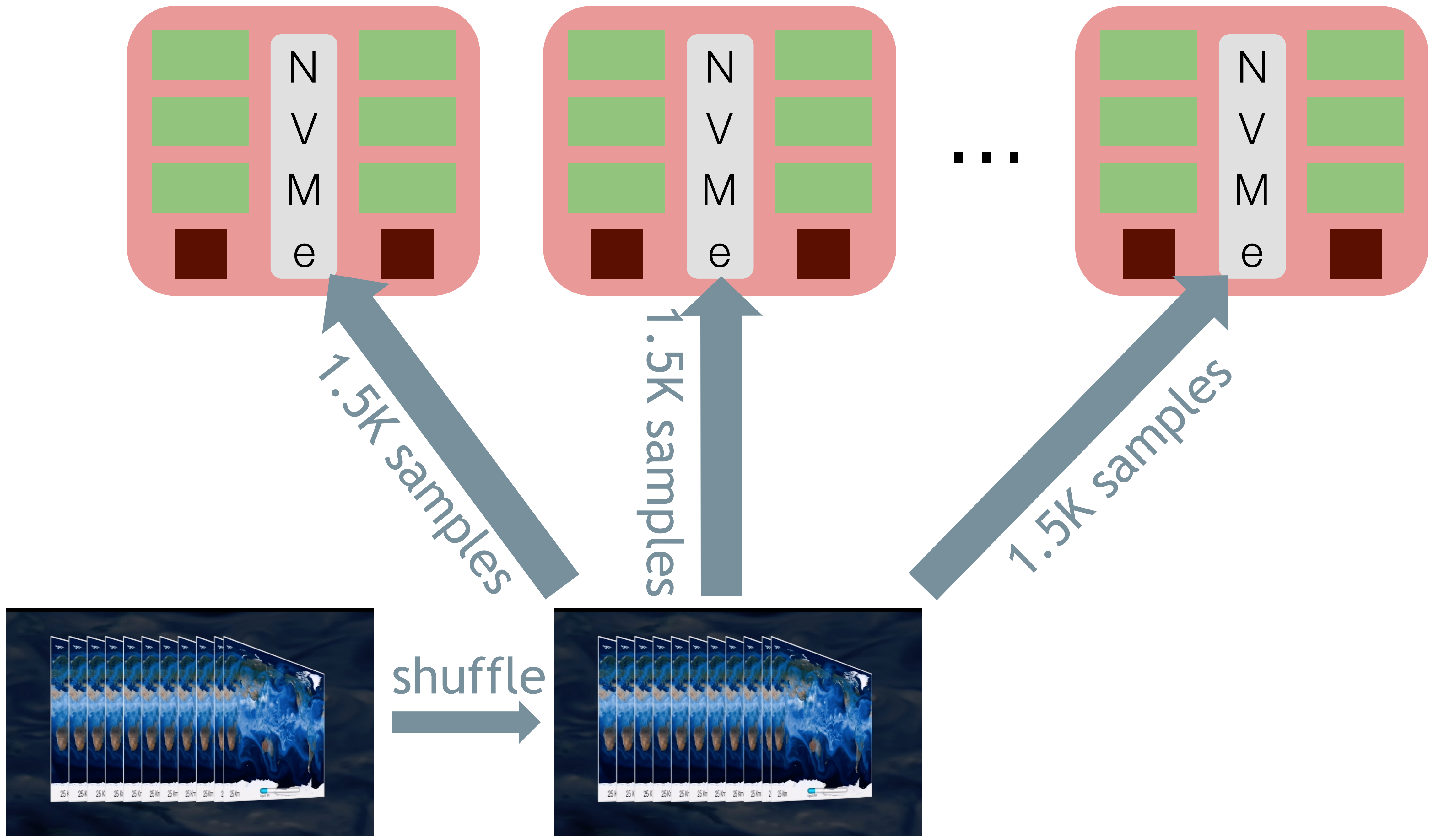




DeepLabv3+, 66 layers,
43.7M parameters, 14.4 TF/sample

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



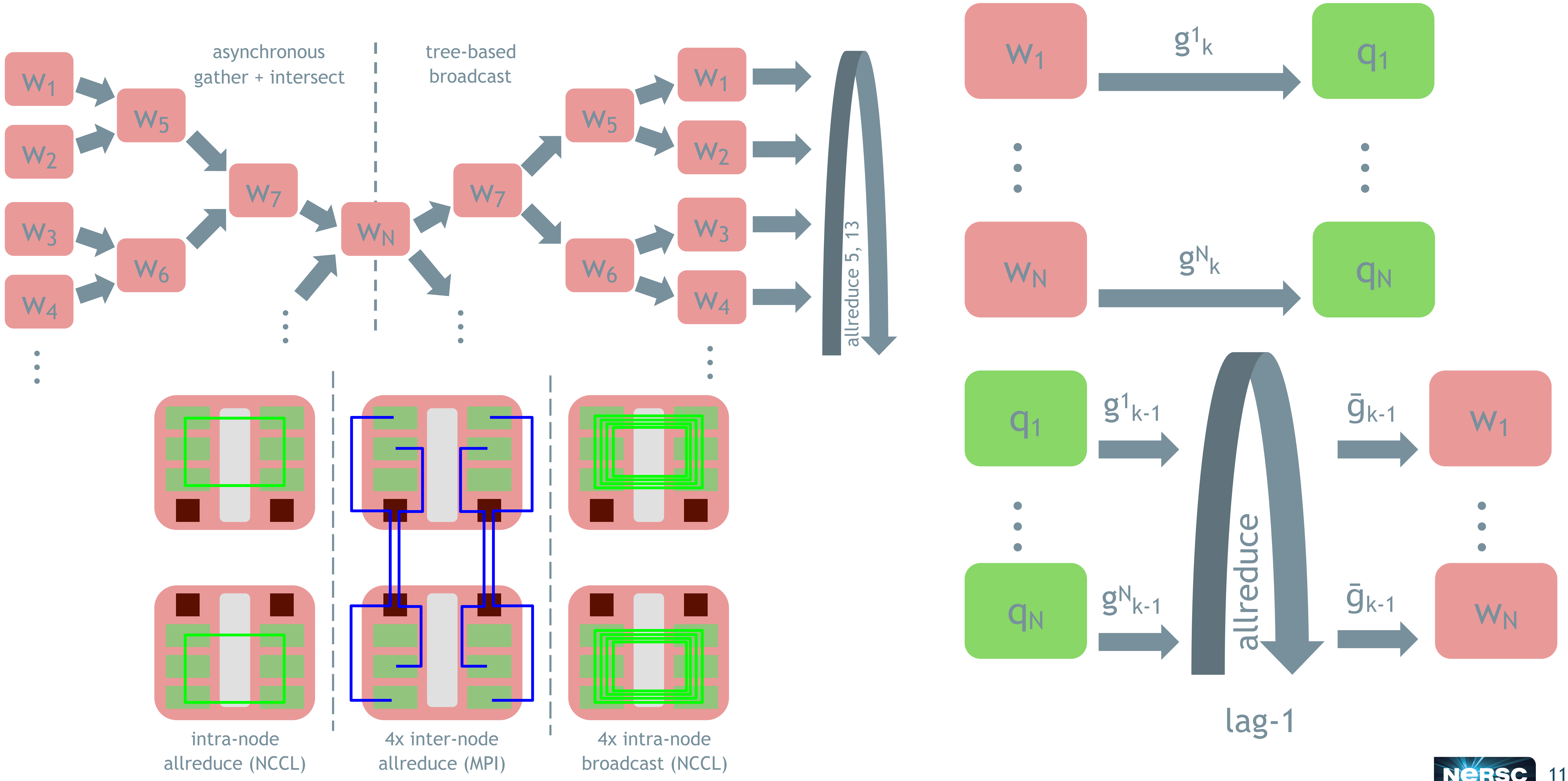
- 250 training samples/GPU (15 GB), sample w/ replacement
- each file will be read at most once from FS
- files shared between nodes via MPI (mpi4py)
- preprocess and feed data to GPU asynchronously using tf.data and python multiprocess

Single Node Performance

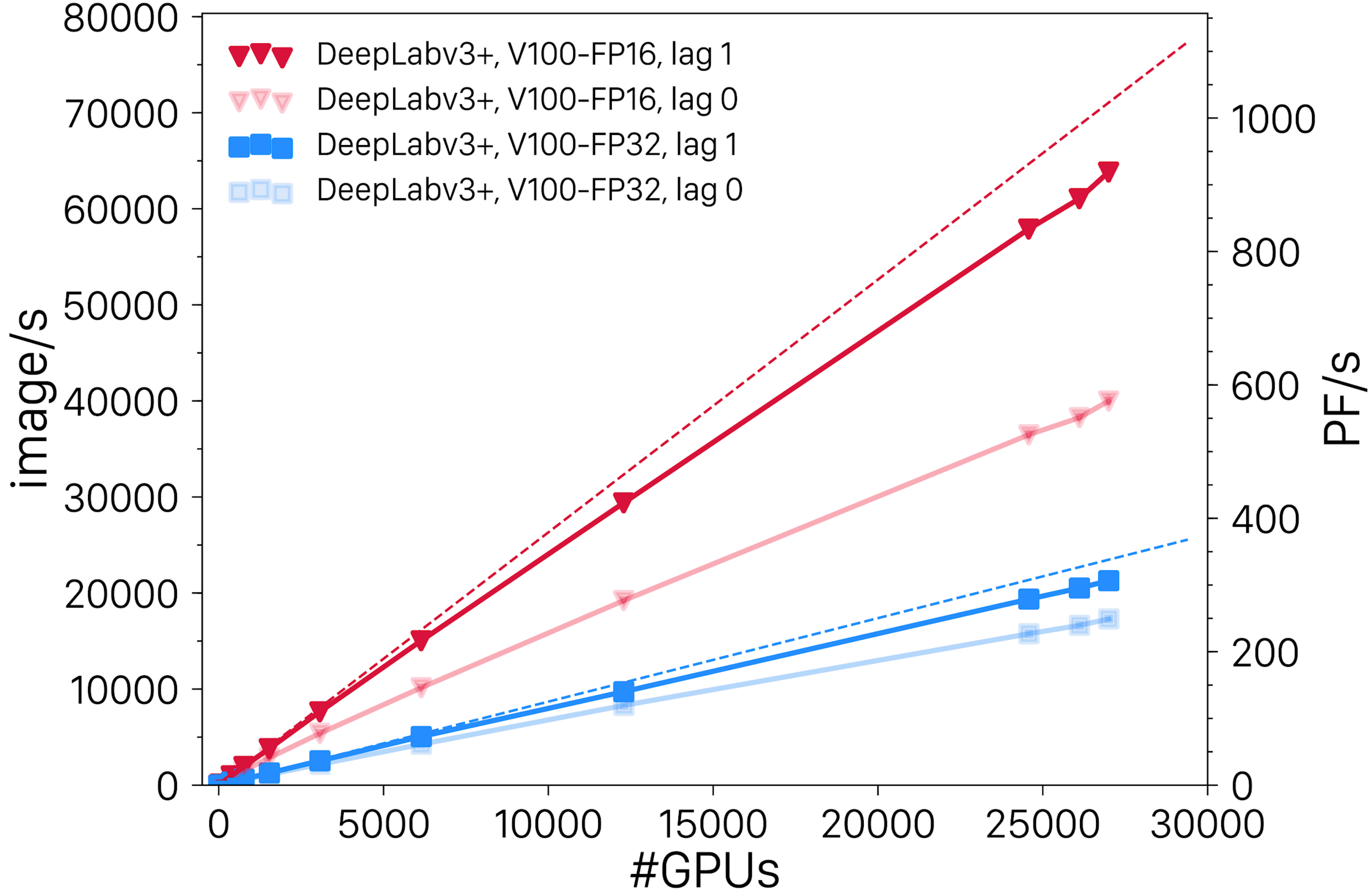
- GPU execution profiled with CUDA profiler, kernels grouped by category
- convolution kernels: use latest cuDNN, favor higher computational intensity
- pay attention to memory layout to reduce transposes and copies
- tuning input pipeline on CPU to keep off critical path

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

Communication Optimizations



Scaling DeepLabv3+



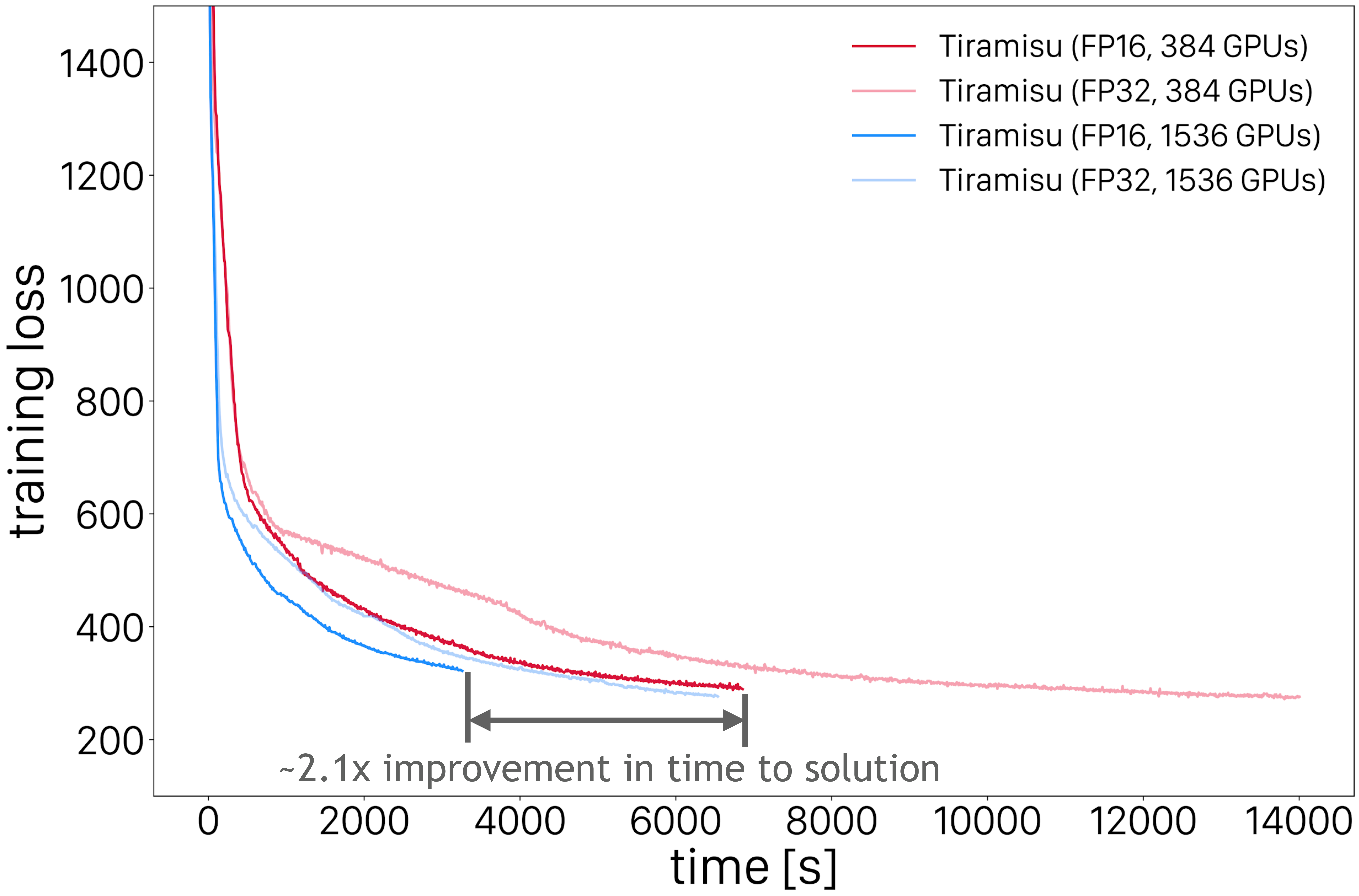
- FP16-model sensitive to communication
- FP16-model BW-bound (only 2.5x faster than FP32)
- excellent scaling for both precisions on Summit when gradient lag is used

999 PetaFlop/s (FP16) sustained

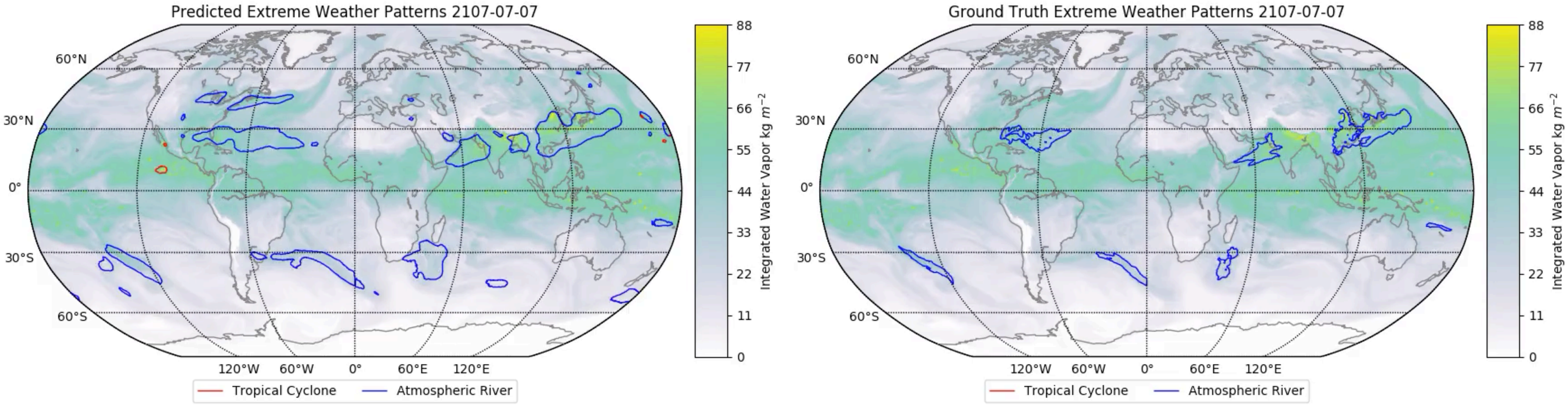
1.13 ExaFlop/s (FP16) peak

DeepLabv3+, 4560 nodes (27360 GPU)

Concurrency/Precision and Convergence



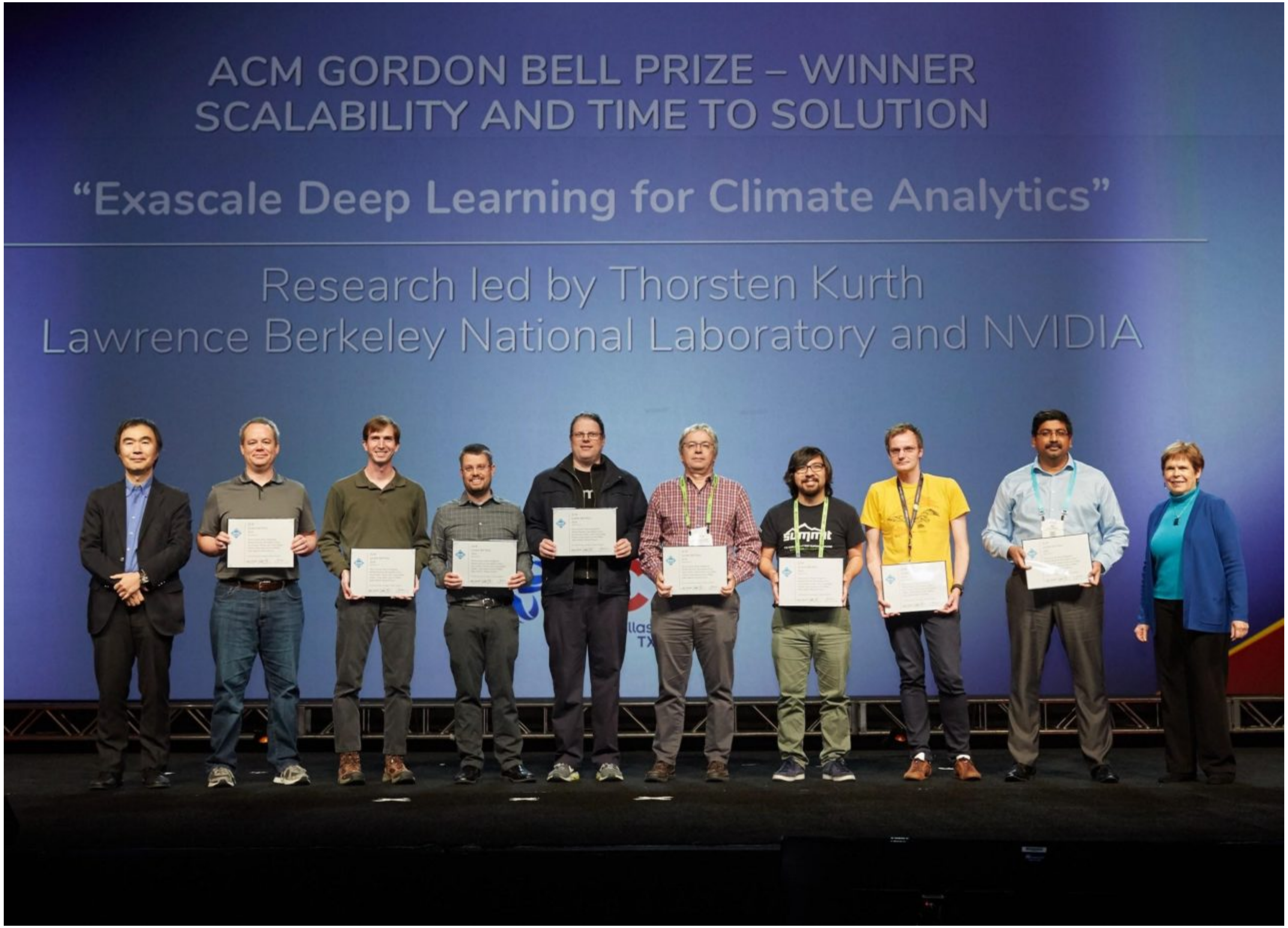
Segmentation Animation



- 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
- compute capabilities at leading HPC facilities can be utilized to tackle difficult scientific deep learning problems
- software enhancements benefit deep learning community as a whole
- HPO and convergence at scale still an open problem
- deep learning-powered techniques usher in a new era of precision analytics for various science areas

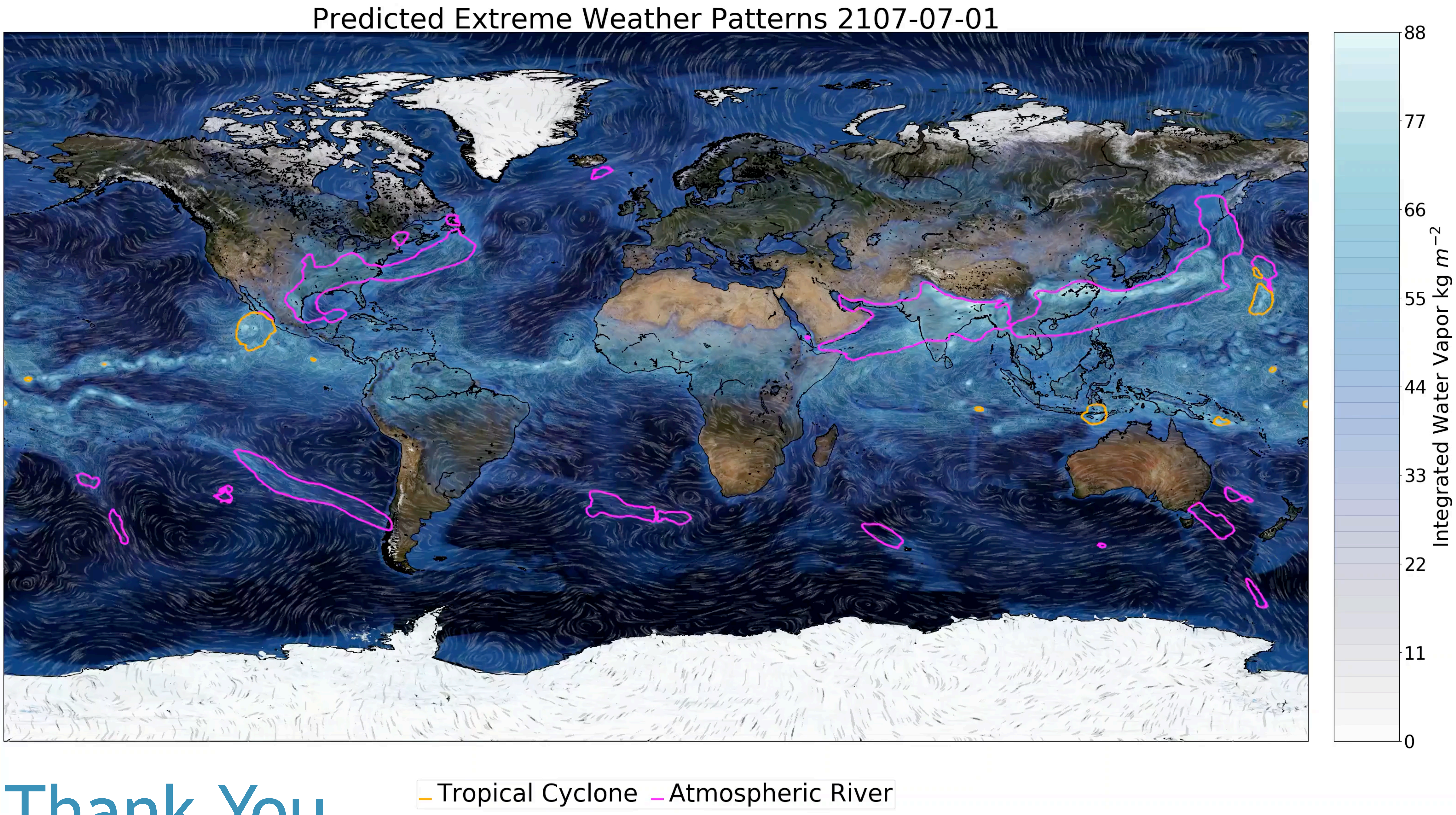

ACM GORDON BELL PRIZE – WINNER
SCALABILITY AND TIME TO SOLUTION
"Exascale Deep Learning for Climate Analytics"
Research led by Thorsten Kurth
Lawrence Berkeley National Laboratory and NVIDIA

https://arxiv.org/abs/1810.01993

Paper Link

https://youtu.be/p45kQklIsd4

TensorFlow Dev Summit 2019 Trailer


Predicted Extreme Weather Patterns 2107-07-01
88
77
66
55
44
33
22
11
0
Integrated Water Vapor kg m^{-2}
Tropical Cyclone
Atmospheric River

Thank You