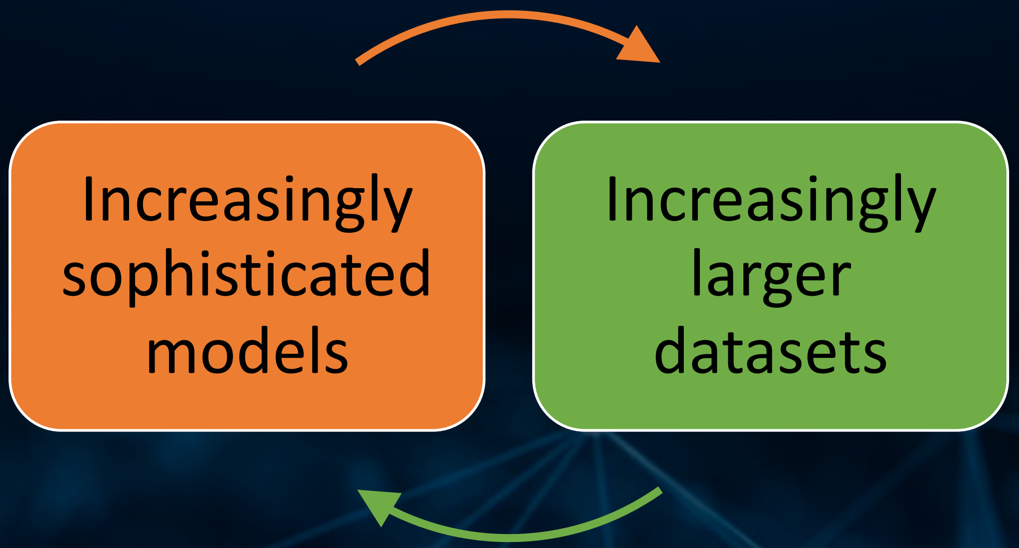
Accelerated Deep Learning via Efficient, Compressed and Managed Communication

Marco Canini

sands.kaust.edu.sa

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Deep Learning



Increasingly sophisticated models

Increasingly larger datasets

Innovation fueled by leaps in (costly) infrastructure:

Clusters with hundreds of machines,

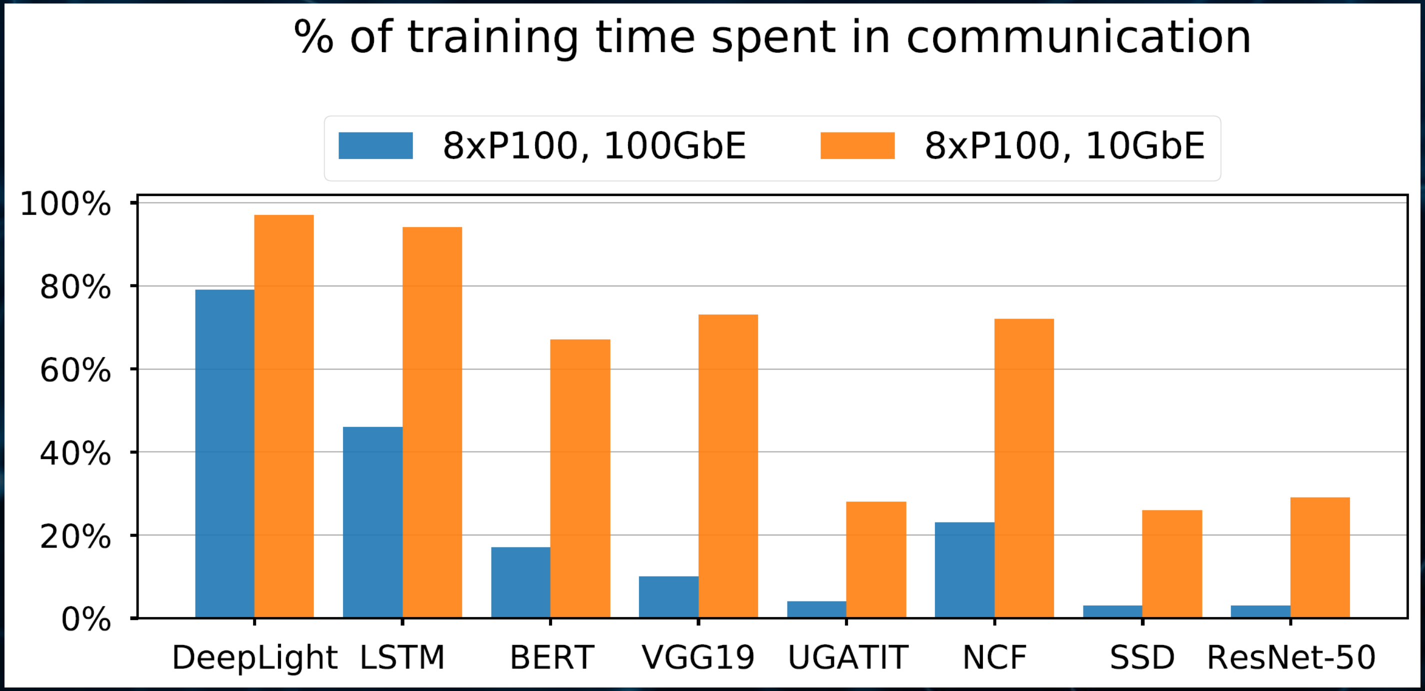
each with many HW accelerators (GPUs)

Compute requirements **doubling every 3 months!**

Training models is still **very time-consuming**: days or even weeks!

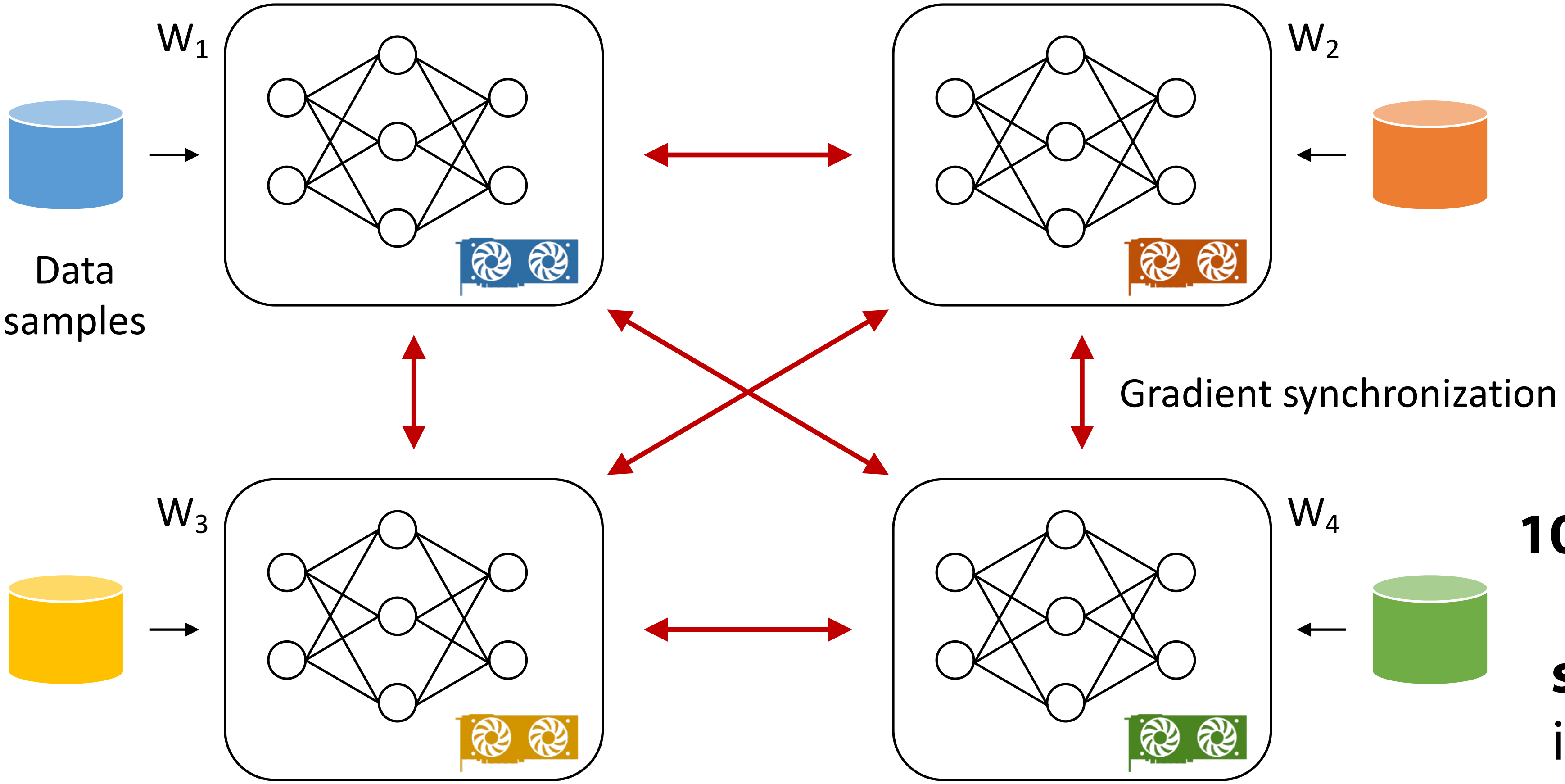
Scaling Machine Learning

Make efficient use of combined resources at multiple worker nodes while tackling significant synchronization overheads



Can the network be the ML accelerator?

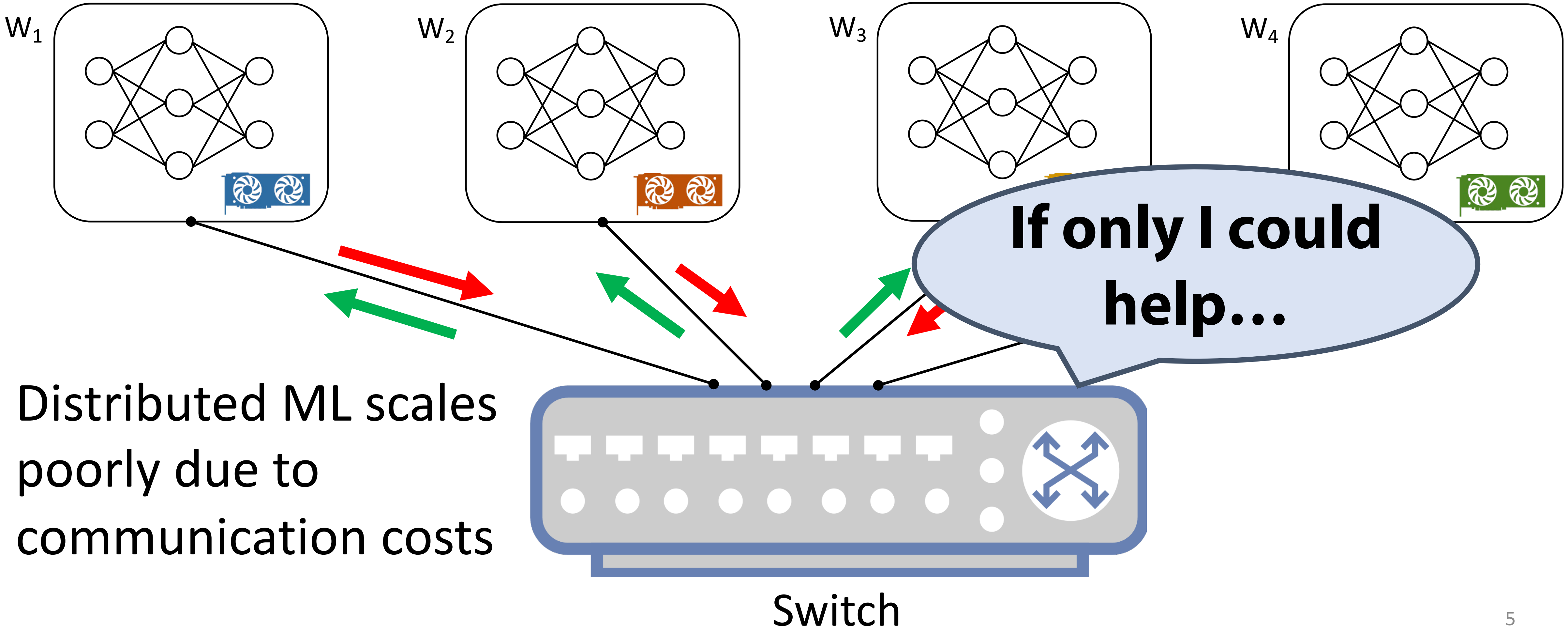
Data-parallel distributed DNN training



All-to-all intensive communication pattern

100s of MBs to GBs gradient synchronization in **each iteration**

A closer look at model synchronization



Distributed ML scales poorly due to communication costs

SwitchML: Co-design ML and networking

[NSDI’21] github.com/p4lang/p4app-switchML

Challenges

Limited computation

Limited storage

No floating points

Packet loss

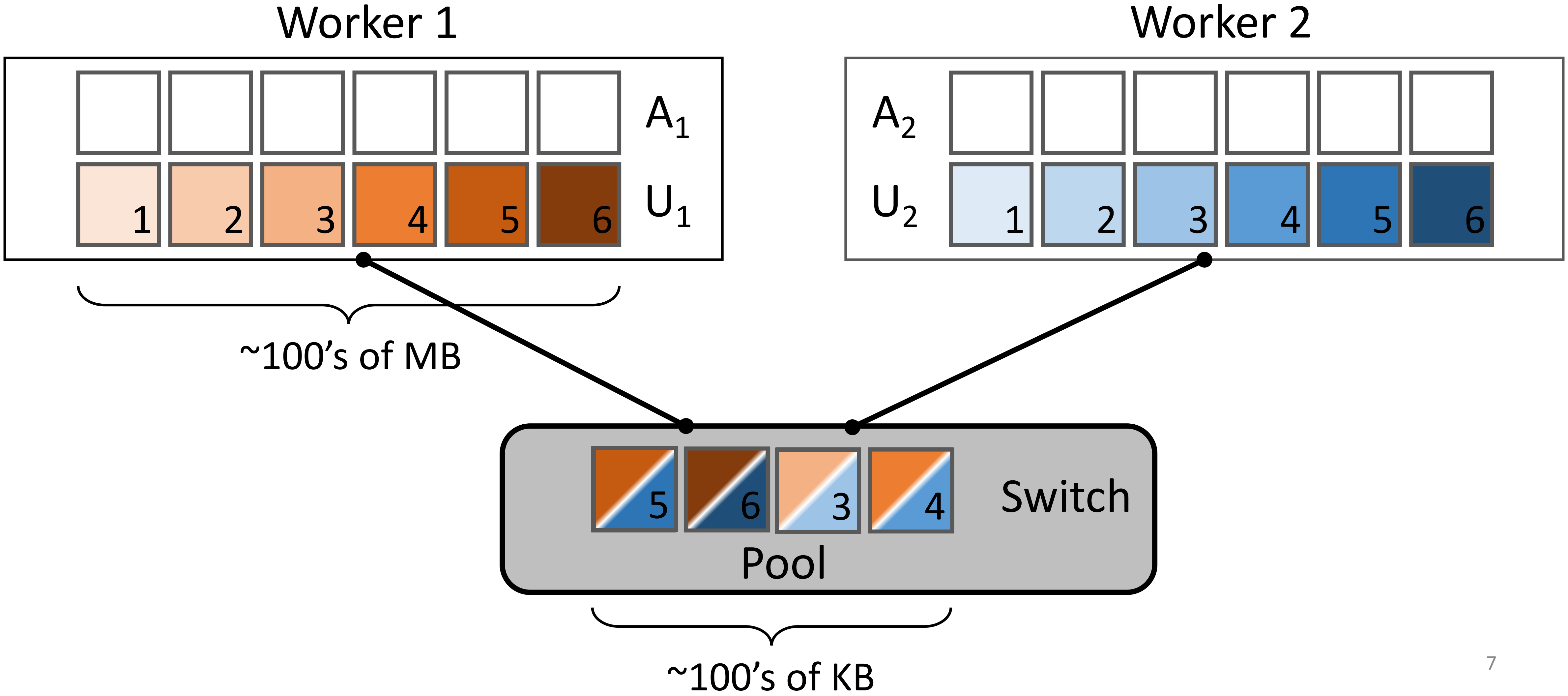
6.5 Tbps
programmable
data plane

Design

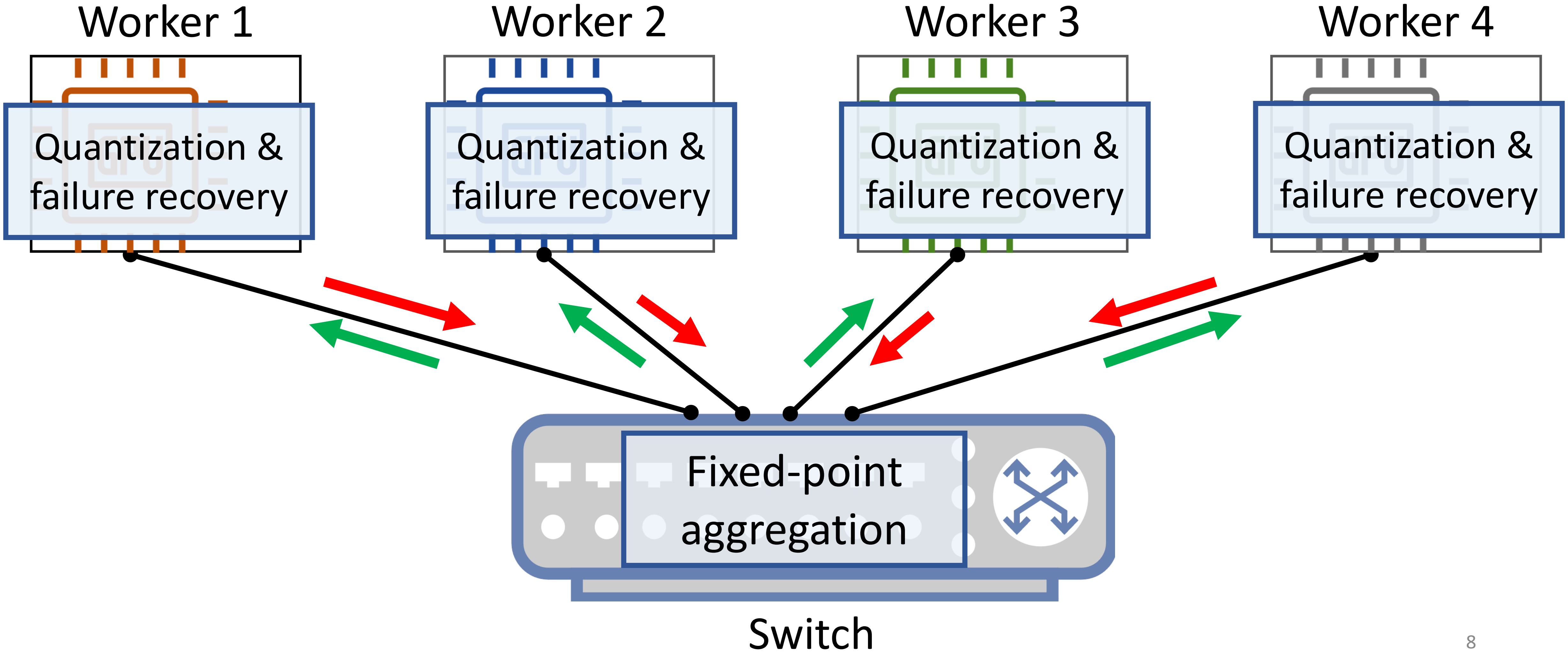
- Combined switch-host architecture
- Pool-based streaming aggregation
- Quantized integer operations
- Failure-recovery protocol
- In-switch RDMA implementation

In collaboration with: Microsoft BAREFOOT NETWORKS | an Intel company W UNIVERSITY of WASHINGTON

Streaming aggregation



Combined switch-host architecture


Worker 1
Worker 2
Worker 3
Worker 4
Quantization & failure recovery
Quantization & failure recovery
Quantization & failure recovery
Quantization & failure recovery
Fixed-point aggregation
Switch

Combined switch-host architecture

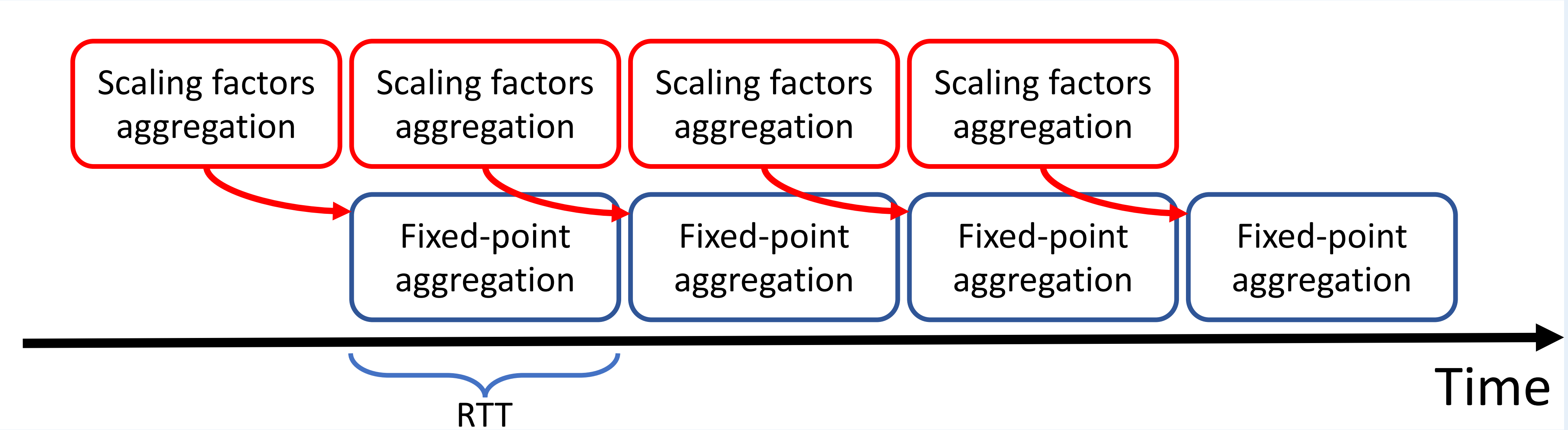
Worker 1

Worker 2

Worker 3

Worker 4

Block quantization


Scaling factors aggregation
Scaling factors aggregation
Scaling factors aggregation
Scaling factors aggregation
Fixed-point aggregation
Fixed-point aggregation
Fixed-point aggregation
Fixed-point aggregation
RTT
Time

Switch

Combined switch-host architecture

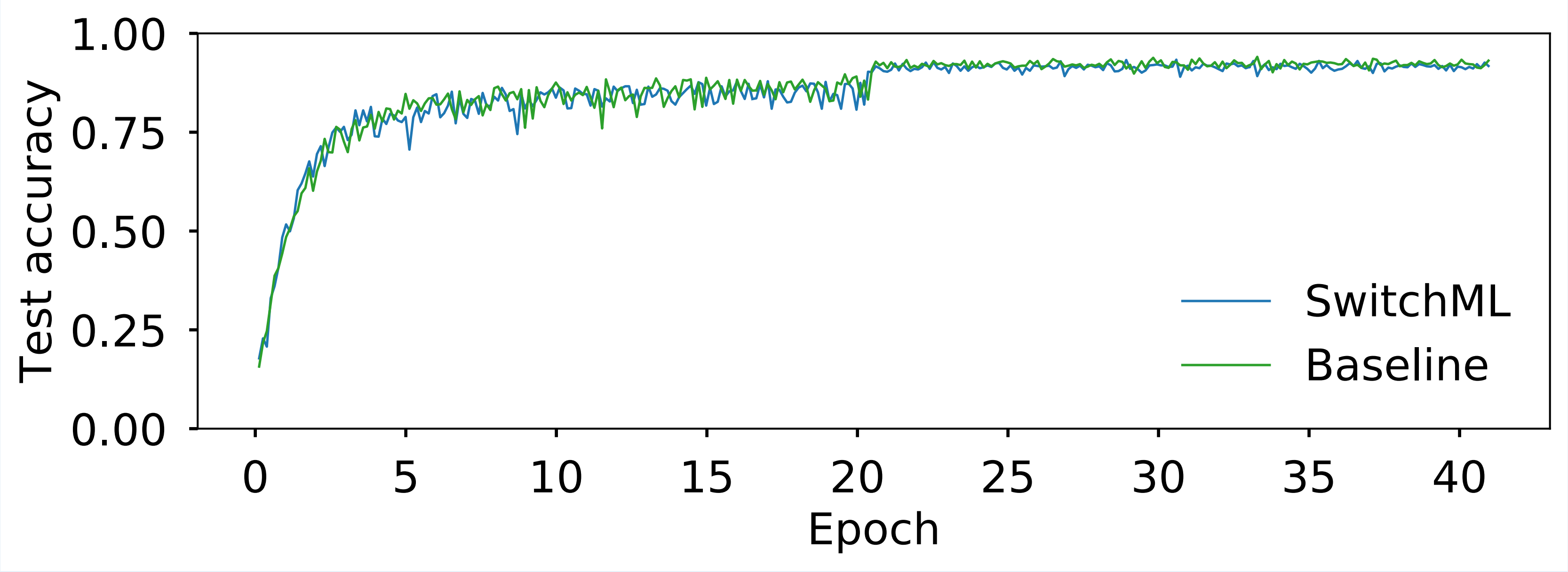
Worker 1

Worker 2

Worker 3

Worker 4

Quantization allows training to similar accuracy in a similar number of iterations as an unquantized network

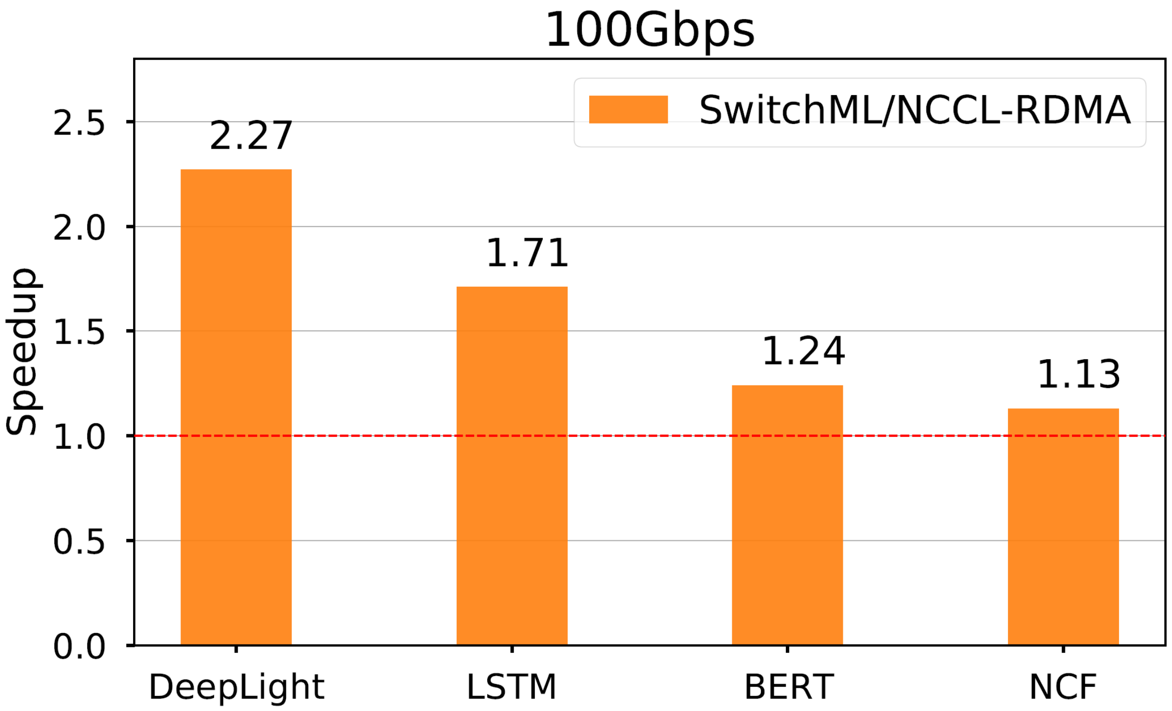


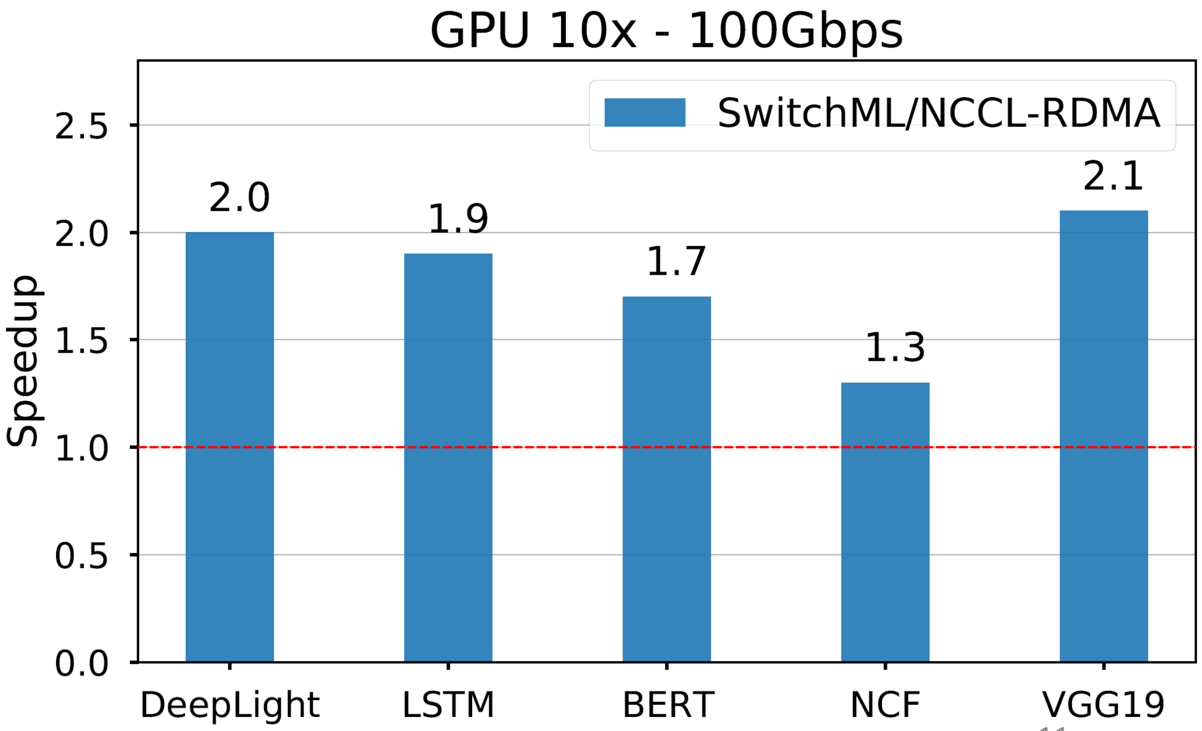
Switch

How much faster is SwitchML?

SwitchML provides a speedup in training throughput up to 2.27× on 100Gbps networks

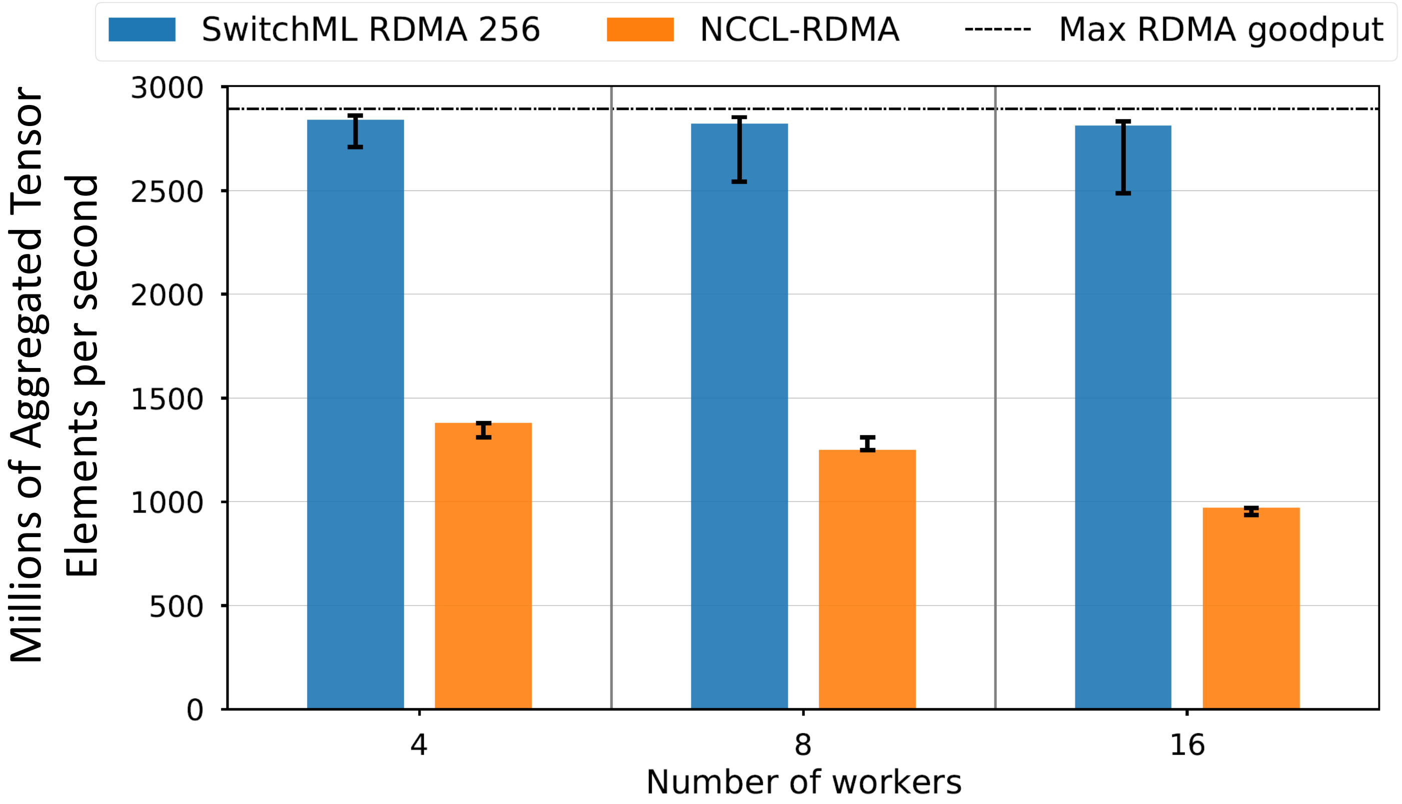
Speedup is higher with faster GPUs that reduce the computation/communication ratio





How does SwitchML scale with # of workers?

SwitchML performance does not depend on the number of workers



FPISA [NSDI’22]

- How to perform floating point ops on programmable switches?
- Proposed mechanisms to enable native floating point support in commodity PISA switches (w/ a few, small HW modifications)

Sparse Collective Communication

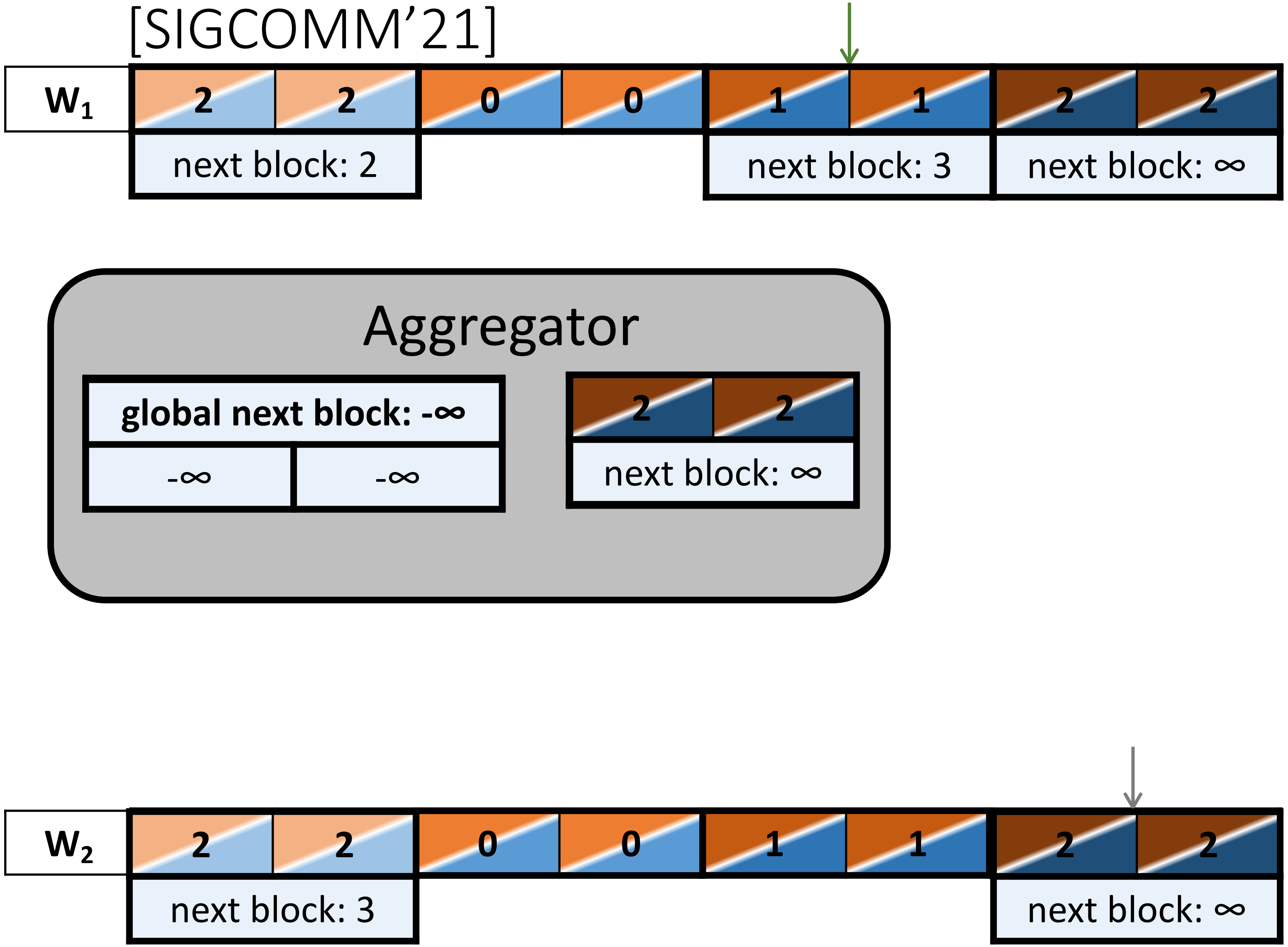
Many gradients in huge models are highly sparse

Model	Task	Model size	Sparsity
DeepLight	CTR prediction	2.3 GB	**99%**
LSTM	Language modeling	1.5 GB	**94%**
BERT	Qs answering	1.3 GB	9%
NCF	Recommendation	680 MB	**84%**
VGG19	Image classification	548 MB	32%
ResNet152	Image classification	230 MB	21%

How to efficiently aggregate sparse gradients?

OmniReduce: sparse streaming aggregation

[SIGCOMM'21]



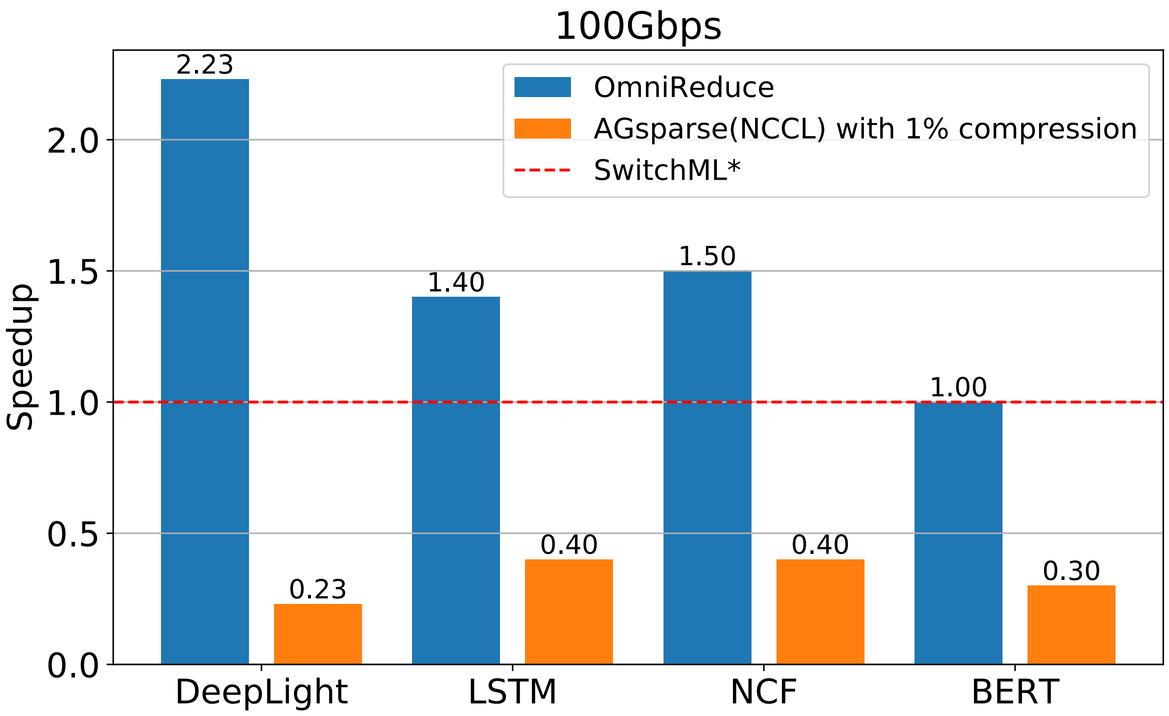
- Split data into blocks
- Stream non-zero blocks to aggregator
- Keep global view of next block

High performance through fine-grained parallelization (*pool of aggregation slots*) and pipelining to saturate network bandwidth

Does OmniReduce speed up training?

OmniReduce is up to 2.23× faster than SwitchML* on 100Gbps networks

Models with higher sparsity gain more from efficient sparse collective communication



- SwitchML* is a software-based implementation of SwitchML

  (fair comparison with software aggregator)
- AGsparse is allgather-based sparse allreduce method

  (compression overheads are not considered)

OmniReduce is in trial deployment at 美团 Meituan

Compressing Gradients

Quantization ▬ → ■
Reduces the bitwidth of each element
(e.g., float32 → float16)

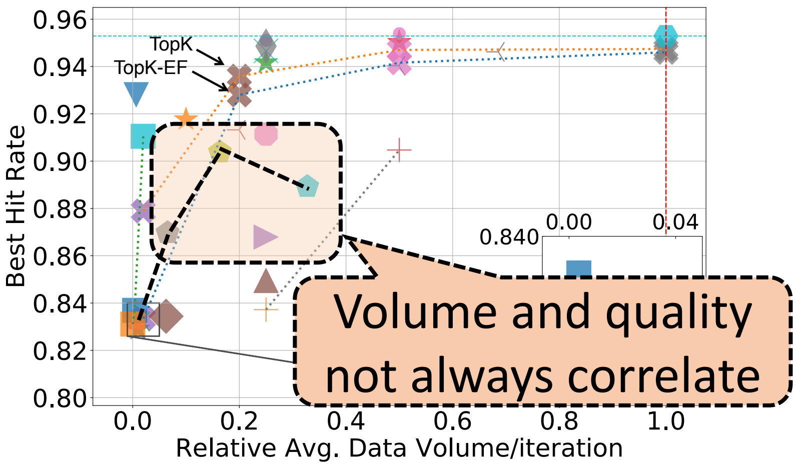
Sparsification
Samples only a few elements
(e.g., top-k values by magnitude)

Decrease communication overhead by reducing data volume via **lossy compression**

Raises interesting trade-offs:
accuracy vs training throughput
vs (de)compression efficiency

GRACE [ICDCS’21]

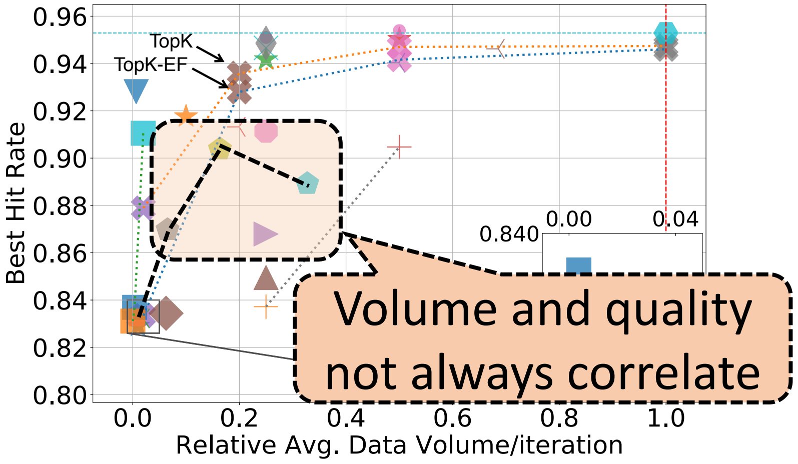
- Unified framework, survey and quantitative evaluation of 16 compressors on 7 benchmarks
  - **No one-size-fits-all, compression has overheads**



Volume and quality not always correlate

GRACE [ICDCS’21]

- Unified framework, survey and quantitative evaluation of 16 compressors on 7 benchmarks
  - **No one-size-fits-all, compression has overheads**



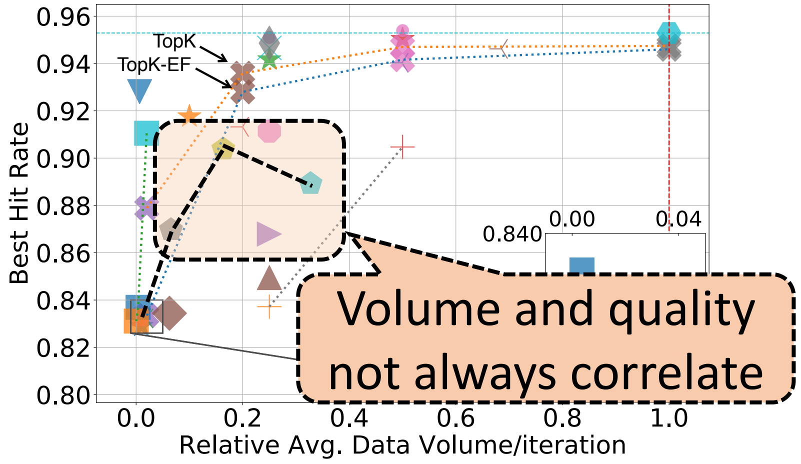
Volume and quality not always correlate

SIDCo [MLSys’21]

- Threshold sparsification: O(n) low overhead but estimation is hard
- Multi-stage estimation + sparsity-inducing distributions (gain 41×)

GRACE [ICDCS’21]

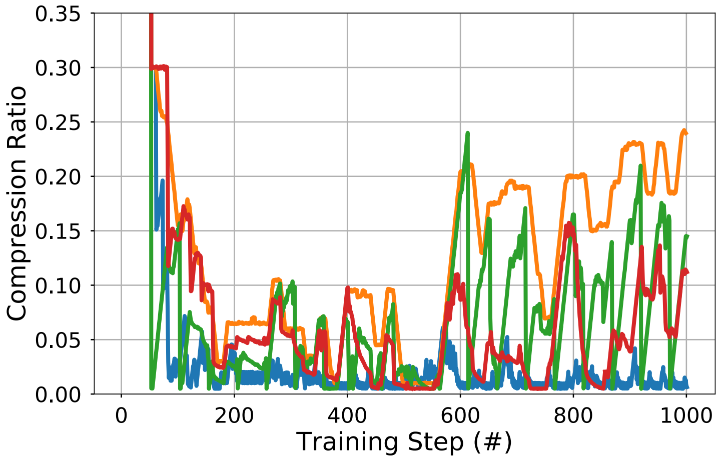
- Unified framework, survey and quantitative evaluation of 16 compressors on 7 benchmarks
  - **No one-size-fits-all, compression has overheads**



Volume and quality not always correlate

SIDCo [MLSys’21]

- Threshold sparsification: O(n) low overhead but estimation is hard
- Multi-stage estimation + sparsity-inducing distributions (gain 41×)

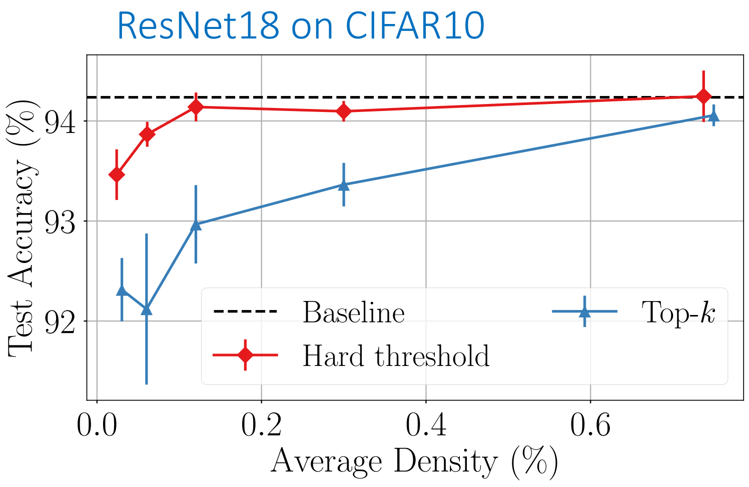


DC2 [INFOCOM’21]

- Fixed compression ineffective in dynamic nets
- Delay-aware adaptive compression couples compression with avail. bandwidth (gain 5.3×)

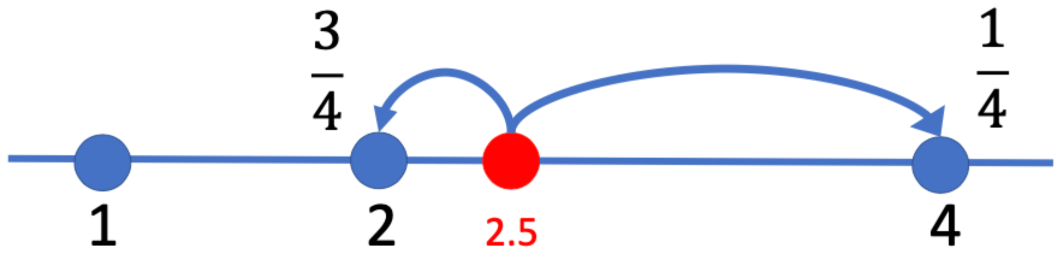
Gradient sparsification as total error minimization [NeurIPS’21]

- Prior work restricted to a fixed comm. budget per iteration, not opt. comm. savings vs. accuracy
- W/ total error perspective (variable comm. budget) we show hard threshold sparsifier is comm. opt. in this model

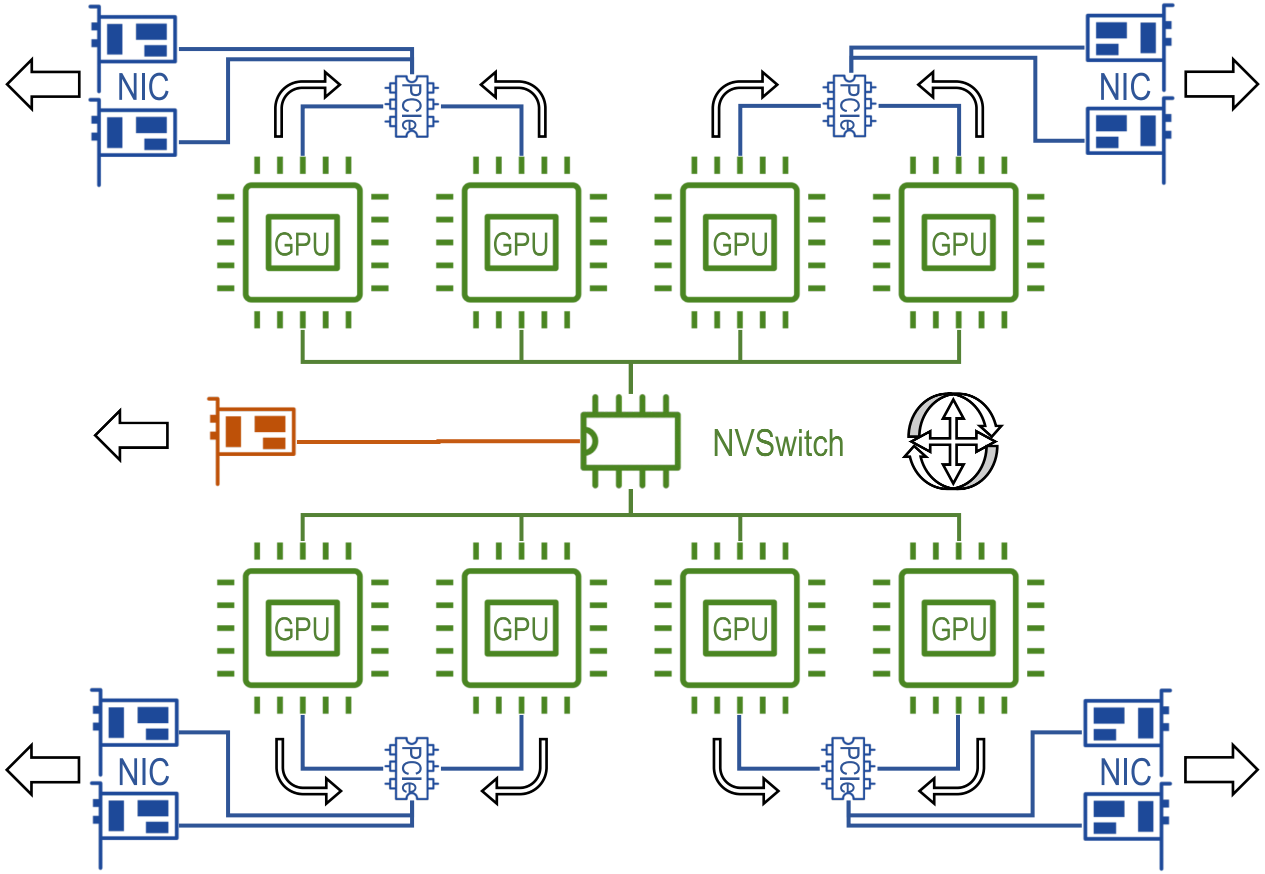


Natural Compression [MSML’22]

- Quantization scheme: randomized rounding to nearest power of 2
- Thanks to IEEE float format, allows to drop the mantissa and send 9 out 32 bits



Something still brewing



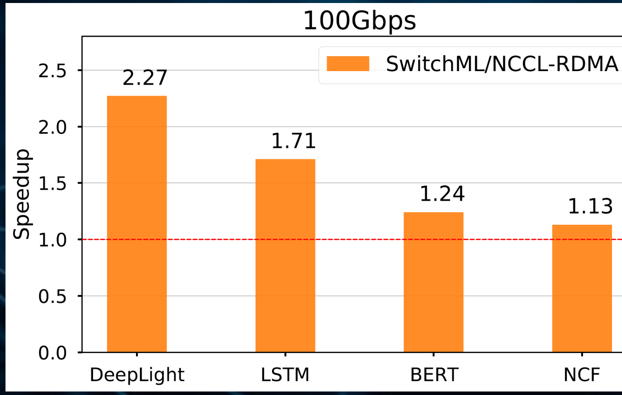
- Actual DC “unit”: Multi-GPU servers
- How to order compression relative to fast intra-node communication?
  - compress first, then intra, then inter
  - intra first, then compress, then inter
- Where to compress?
  - GPU: overheads and contention
  - NIC? emerging DPUs or FPGAs
- But then why send uncompressed data on slow PCI? Add NIC on interconnect?

Summary

Distributed DL increasingly a **communication-bound** workload

Our work seeks to accelerate training with:

- **Efficient** in-network streaming aggregation
- **Compressed** communication at low overhead
- **Managed** adaptation to network dynamics



We achieve **significant speed ups** over existing solutions

Our systems are **open source**

sands.kaust.edu.sa

Get in touch: marco@kaust.edu.sa

ACKs

KAUST

Ahmed M. Abdelmoniem
Omar Alama
M.-Slim Alouini
E. H. Bergou
Muhammad Bilal
Aritra Dutta
Ahmed Elzanaty
Suhaib Fahmy
Jiawei Fei
Chen-Yu Ho
Panos Kalnis
Konstantinos Karatsenidis
Pantelis Papageorgiou
Peter Richtarik
Atal N. Sahu
Amedeo Sapio
Hang Xu

now here

MSR

Dan R. K. Ports
Jacob Nelson

UW

Arvind Krishnamurthy

UIUC

Nam Sung Kim
Yifan Yuan

Barefoot (Intel)

Changhoon Kim
Masoud Moshref

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

This is interesting? Join us!

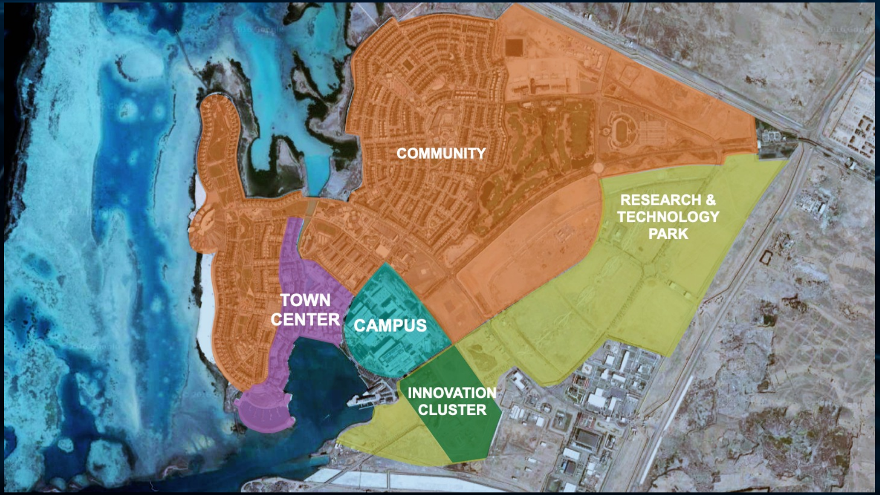
Post-doctoral and MS/PhD student positions

sands.kaust.edu.sa

Our **Aspiration**

Destination

KAUST aspires to be a destination for scientific and technological education and research. By inspiring discoveries to address global challenges, we strive to serve as a beacon of knowledge that bridges people and cultures for the betterment of humanity.

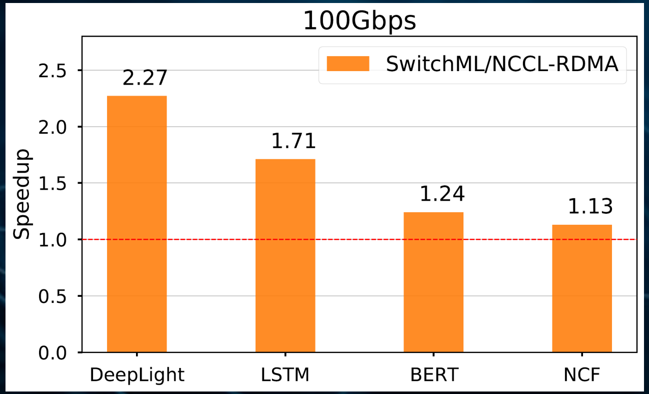


Summary

Distributed DL increasingly a **communication-bound** workload

Our work seeks to accelerate training with:

- **Efficient** in-network streaming aggregation
- **Compressed** communication at low overhead
- **Managed** adaptation to network dynamics



We achieve **significant speed ups** over existing solutions

Our systems are **open source**

sands.kaust.edu.sa

Get in touch: marco@kaust.edu.sa