THC: Accelerating Distributed Deep Learning Using Tensor Homomorphic Compression

Minghao Li, Ran Ben Basat, Shay Vargaftik, ChonLam Lao, Kevin Xu,
Michael Mitzenmacher, Minlan Yu

Deep Neural Networks (DNNs) are prevalent

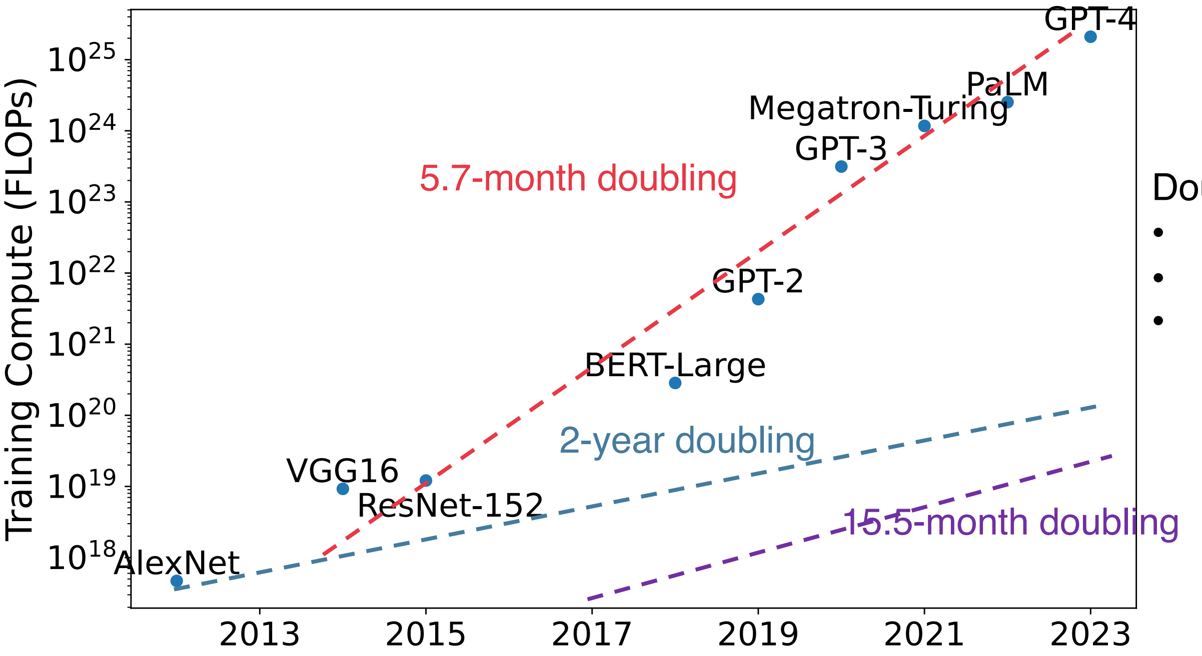
ChatGPT


GitHub Copilot

Midjourney

Midjourney image generator

DNN training job size grows fast



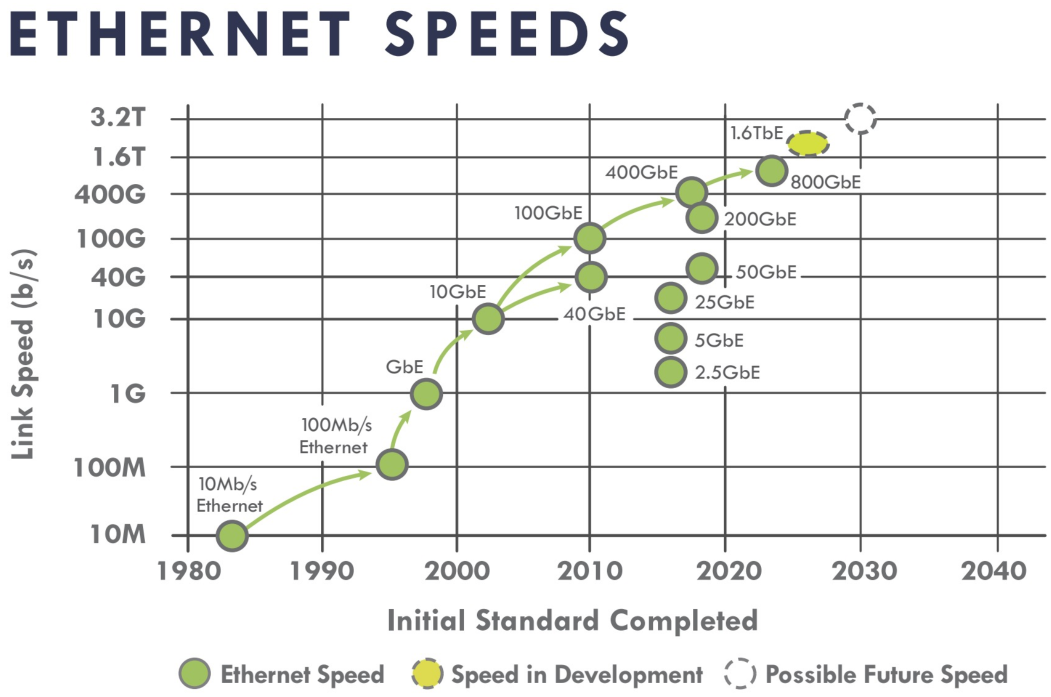
Doubling rate:

- Training compute: ~5.7 months
- GPU processing power: ~2 years
- GPU memory: ≥ 15.5 months

More GPU workers.
More data transferred.

Sources:
[1] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, & Pablo Villalobos. (2022). Compute Trends Across Three Eras of Machine Learning. 2022 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8)
[2] NVIDIA (https://resources.nvidia.com/l/en-us-gpu)

Network bandwidths grow much slower

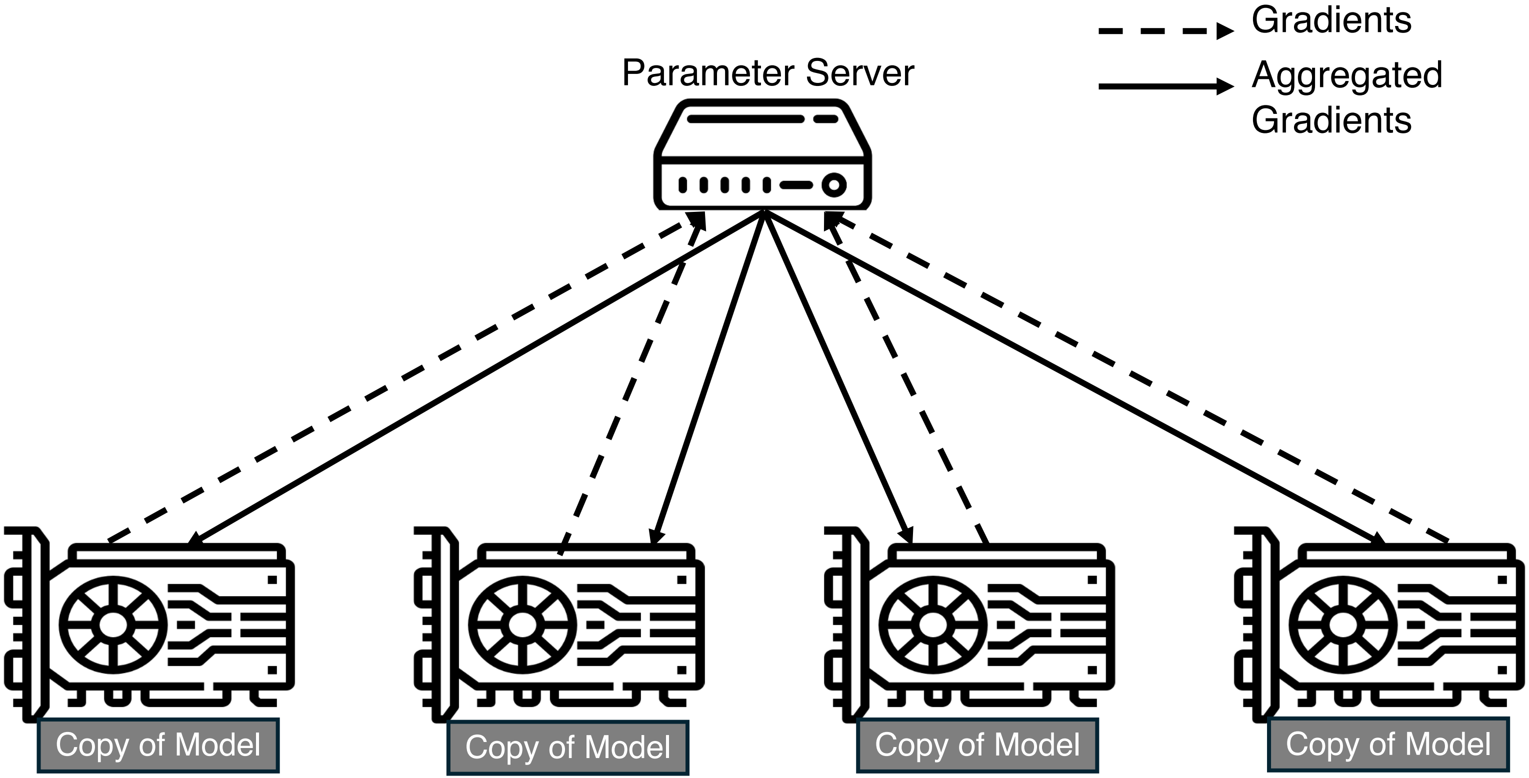


~3-year doubling rate

Network is a bottleneck in distributed training.

Sources:
[1] https://ethernetalliance.org/technology/ethernet-roadmap/

Synchronization in data parallel training



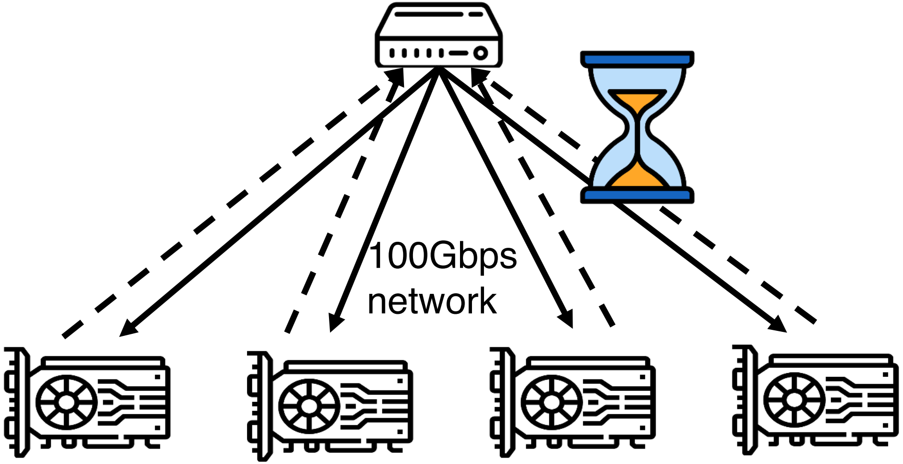
The synchronization cost is already high

BytePS, four A100 GPUs, weak scaling, Stanford Sentiment Treebank (SST2)

Increase the number of workers from one to four.
Ideal speed up: 4 ×

Actual speed up:

- With GPT-2: 2.58 ×
- With BERT-base: 2.27 ×



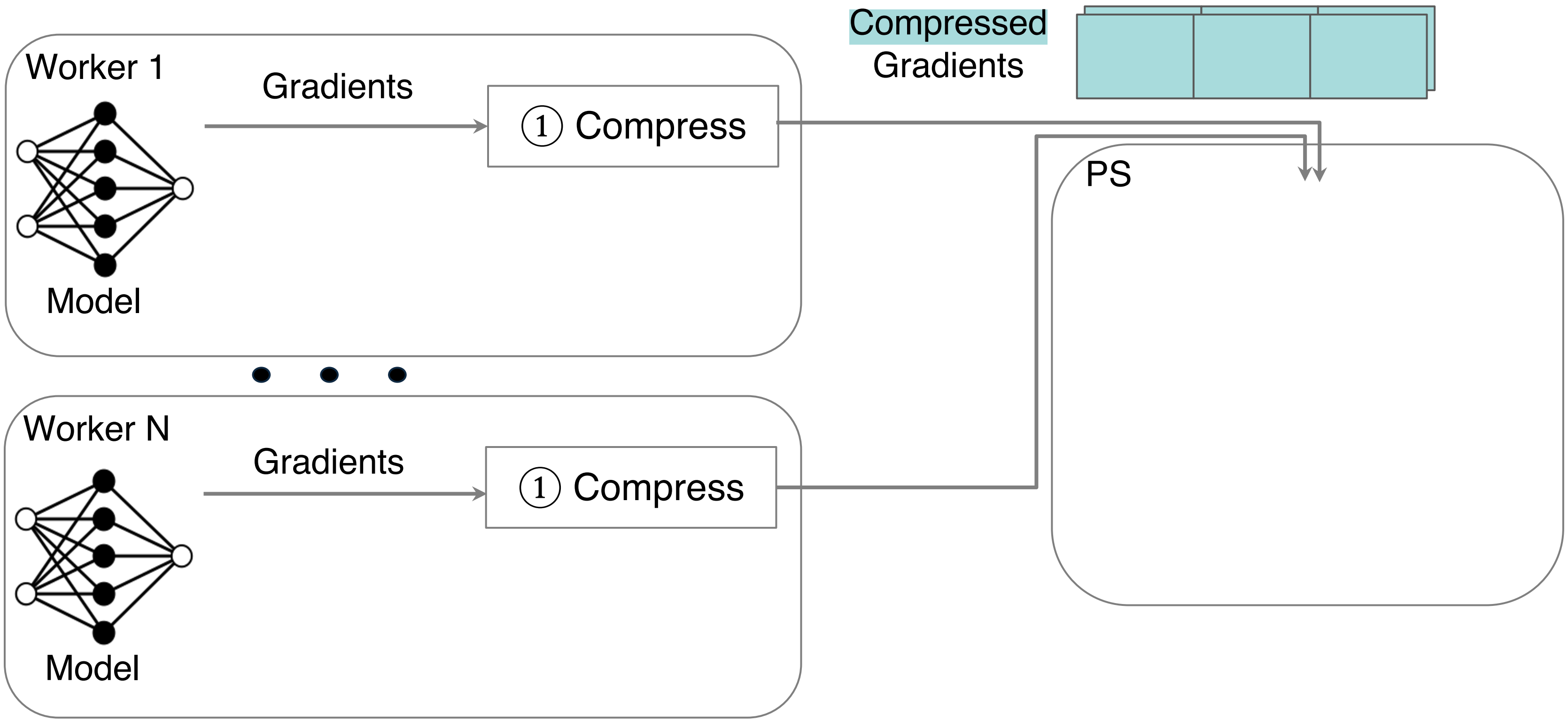
Potential solution: gradient compression

- Send compressed gradients to reduce the number of bits transmitted.
- Previous works integrated gradient compression into training systems.

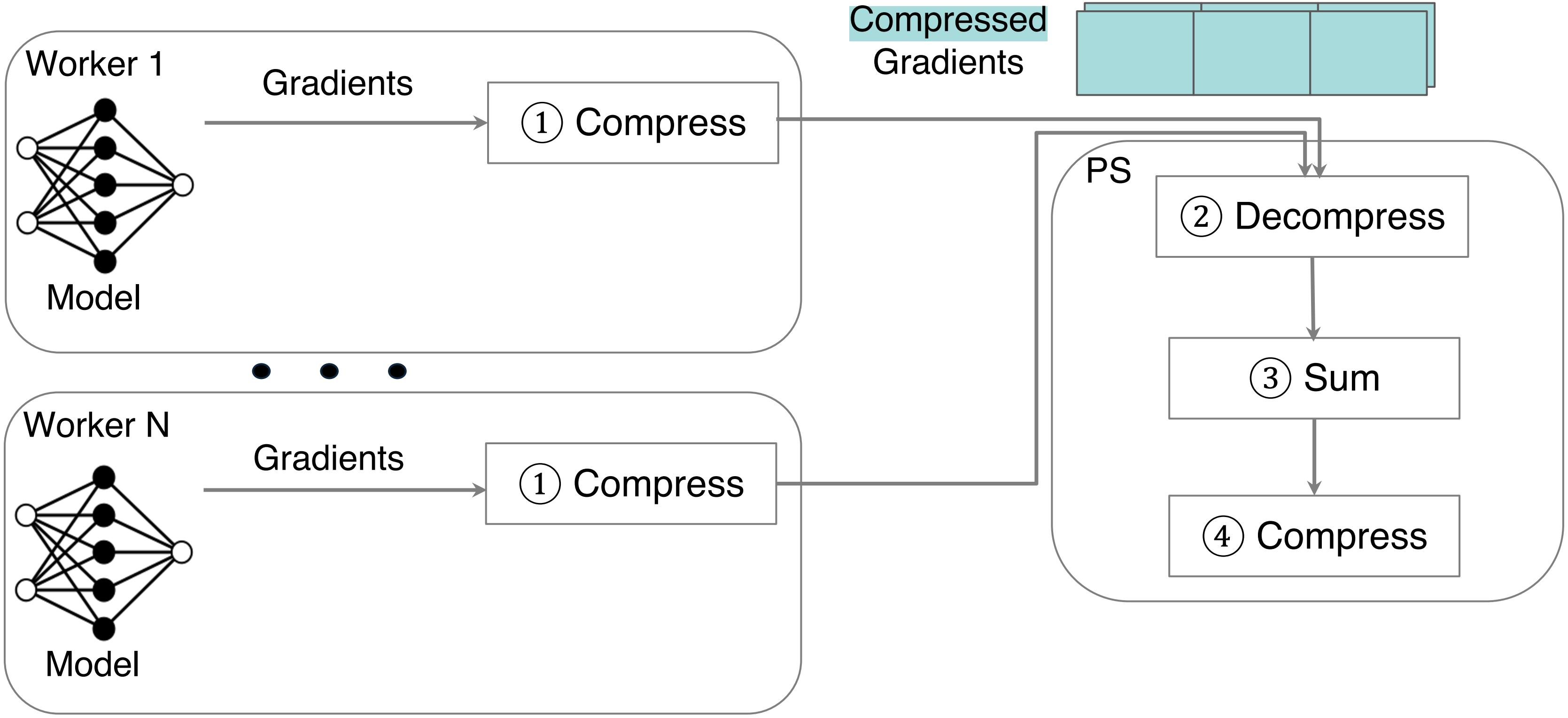
BytePS-Compress	HiPress [SOSP' 21]	Espresso [EuroSys' 23]

Require decompression and re-compression at every synchronization step.

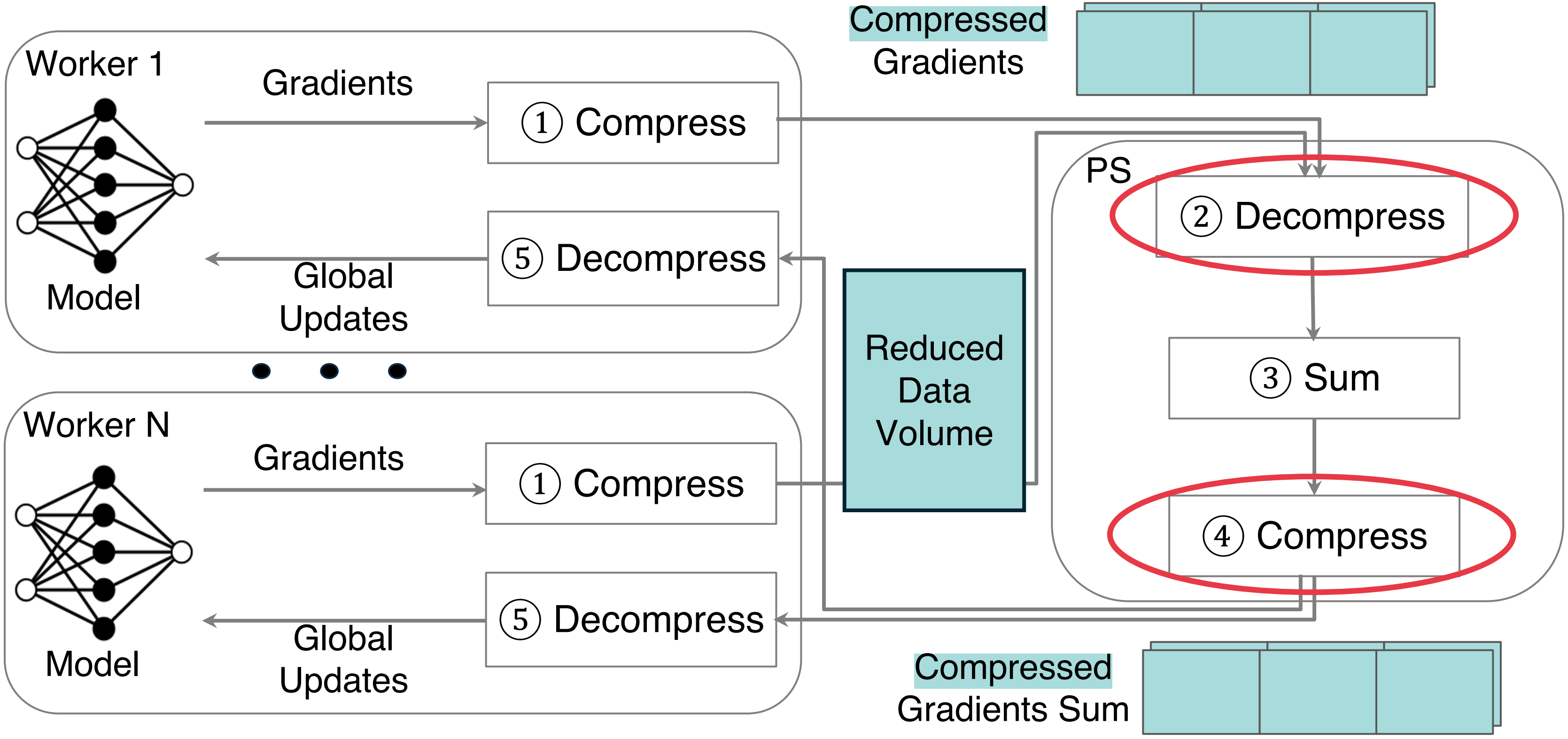
Issue: de- and re-compression at every synchronization step



Issue: de- and re-compression at every synchronization step



Issue: de- and re-compression at every synchronization step

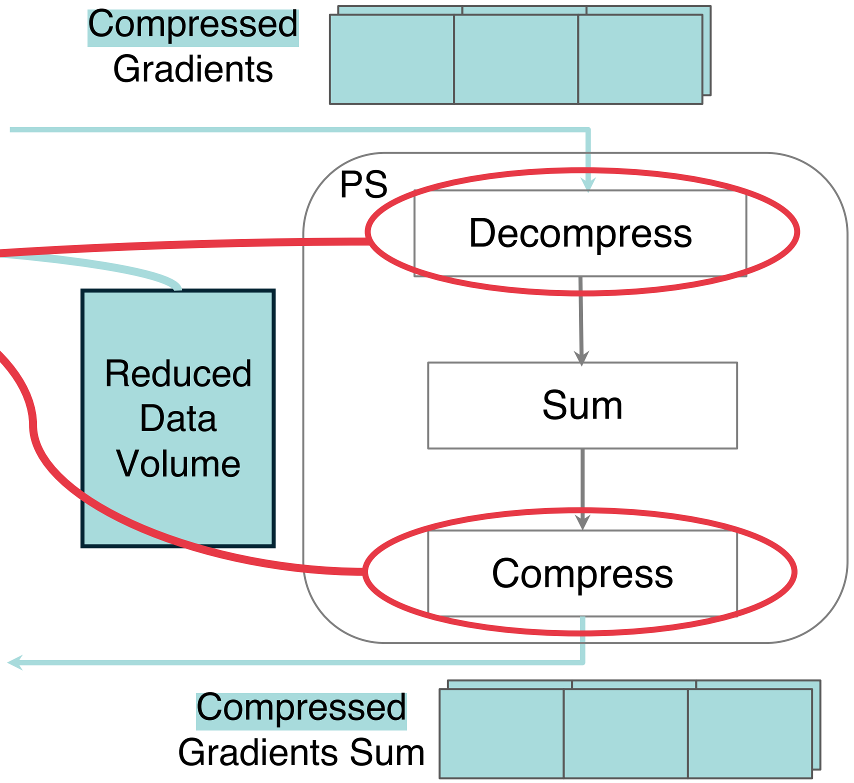


Compression saving diluted by high computational cost

Microbenchmark:

A vector of 1M coordinates,
BytePSCompress-TopK10%

ΔNetwork Time (%)	ΔPS Time (%)	ΔTotal Time (%)
-52.4%	+23.5%	-28.9%



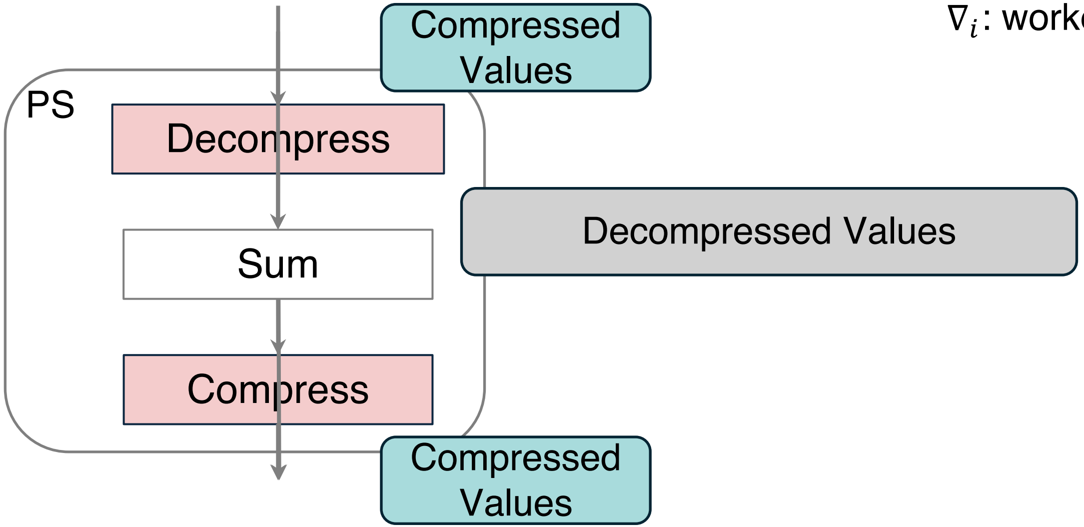
Key idea: Tensor Homomorphic Compression (THC)

Removing decompression on PS requires homomorphic compression:

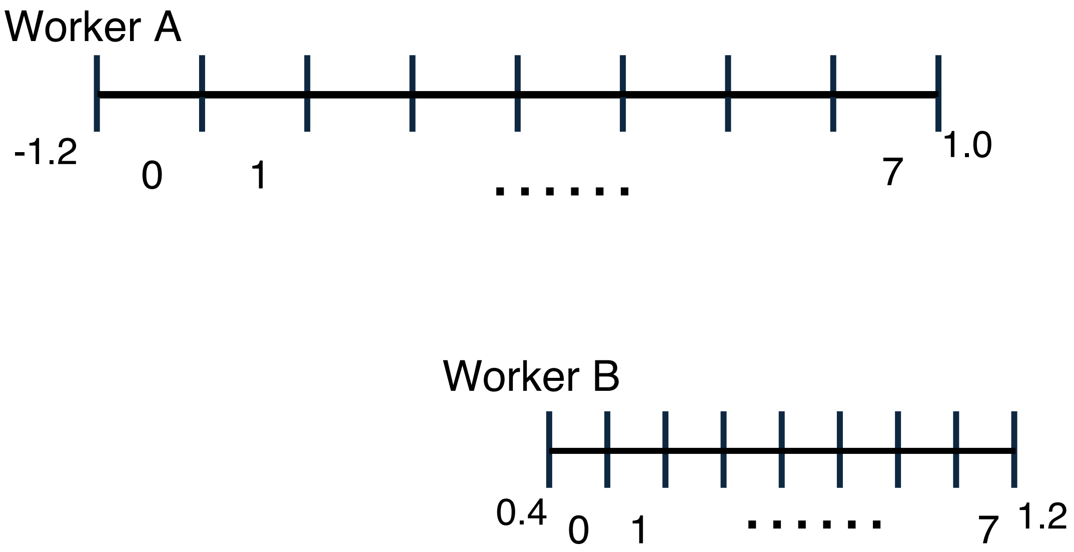
$$\frac{1}{n} \cdot \sum_i \text{Decompress}\,(\text{Compress}(\nabla_i)) = \text{Decompress}\left(\frac{1}{n} \cdot \sum_i \text{Compress}\,(\nabla_i)\right)$$

n: number of workers

∇_i: worker i gradient



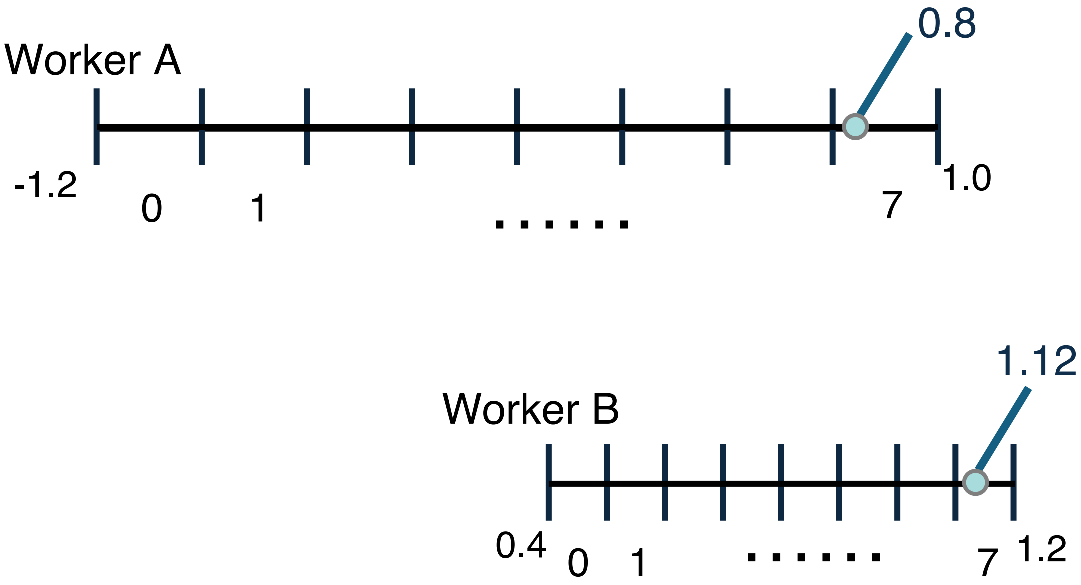
Designing a homomorphic quantization scheme



Quantization schemes:

Convert floats into quantized values taking fewer bits.

Worker-specific quantization is not homomorphic



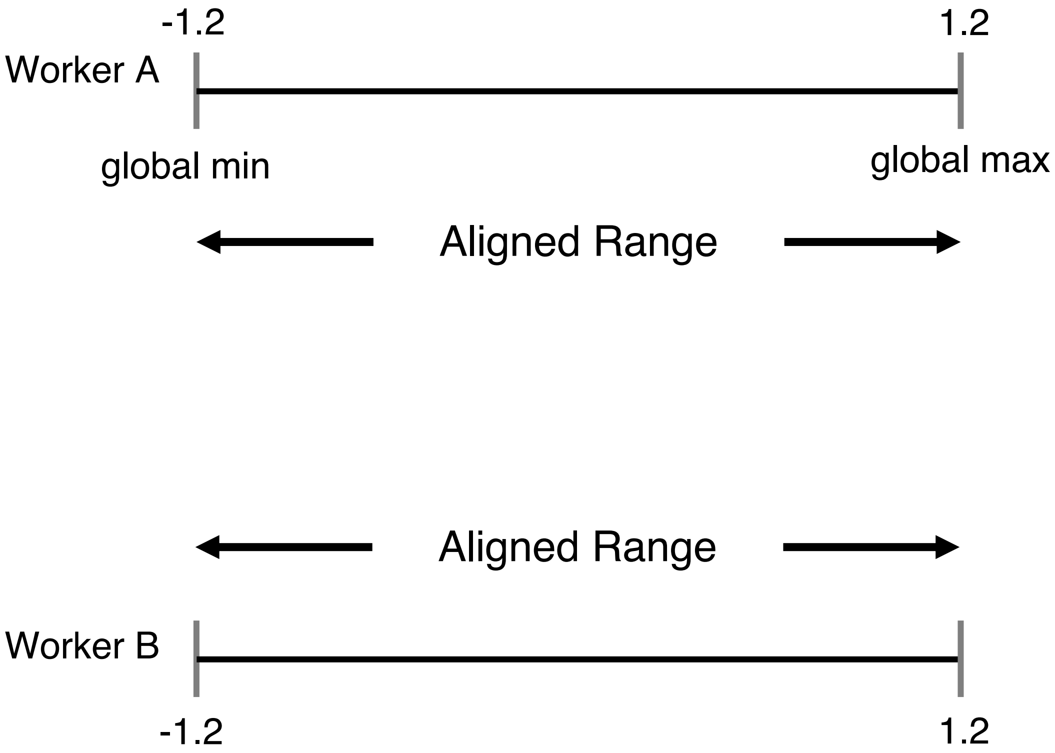
Quantization schemes:

Same quantized value might cover different ranges.

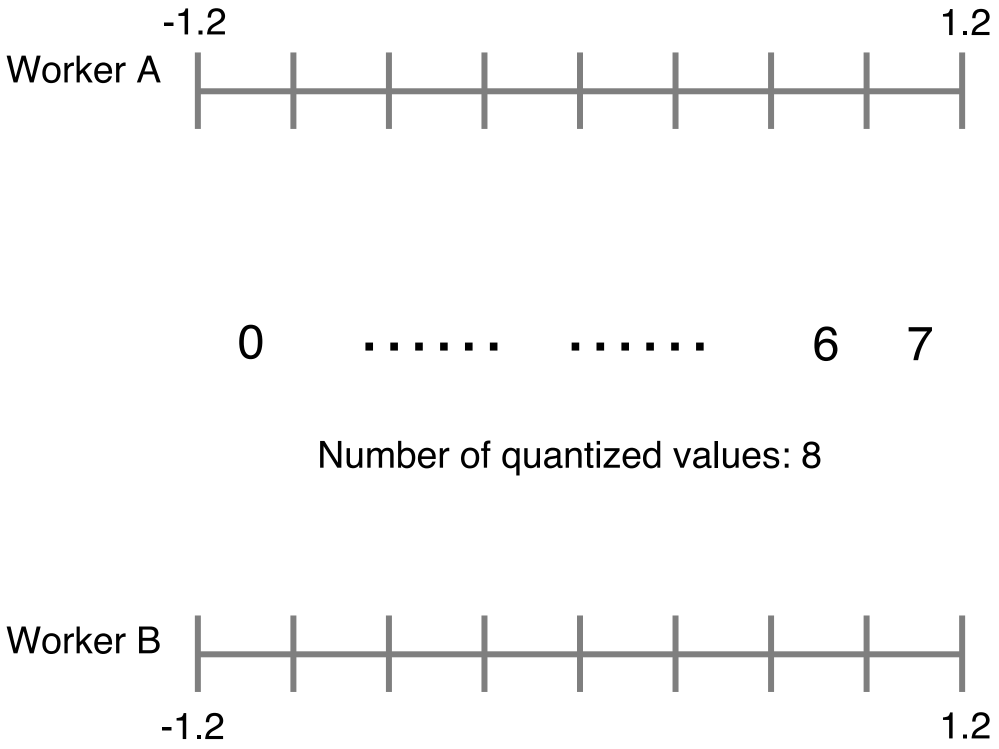
We can't sum up quantized values directly.

Requires PS to scale the quantized values based on the per-worker range.

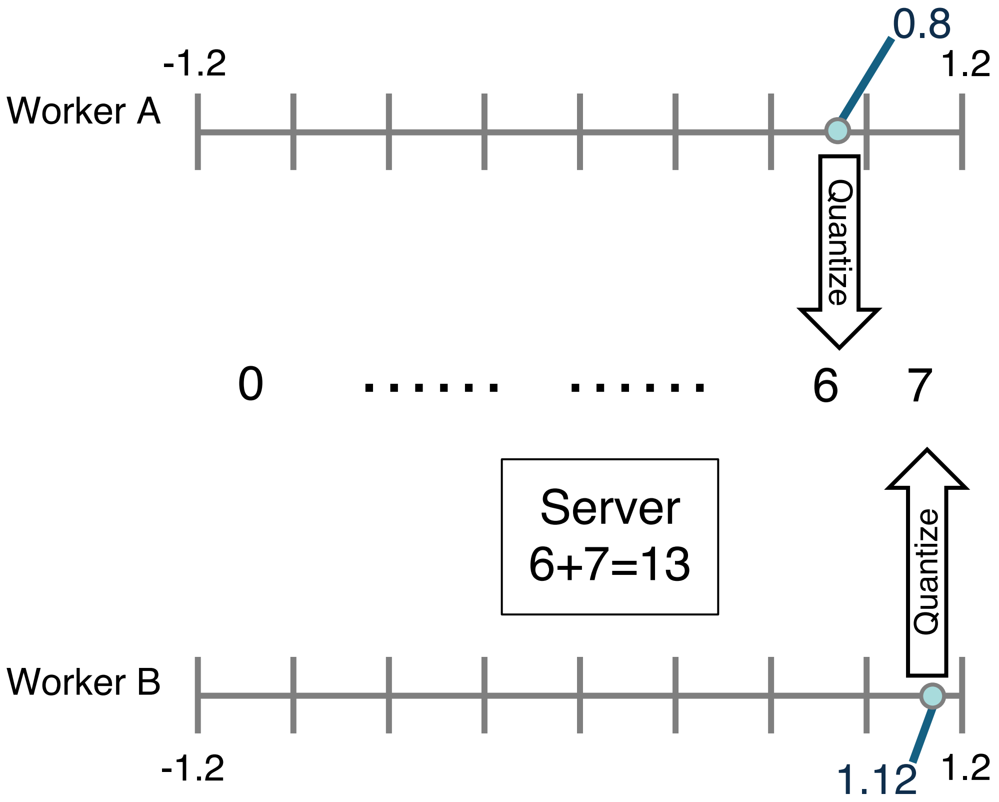
Achieve homomorphism by aligning worker ranges



Quantization with global range is homomorphic

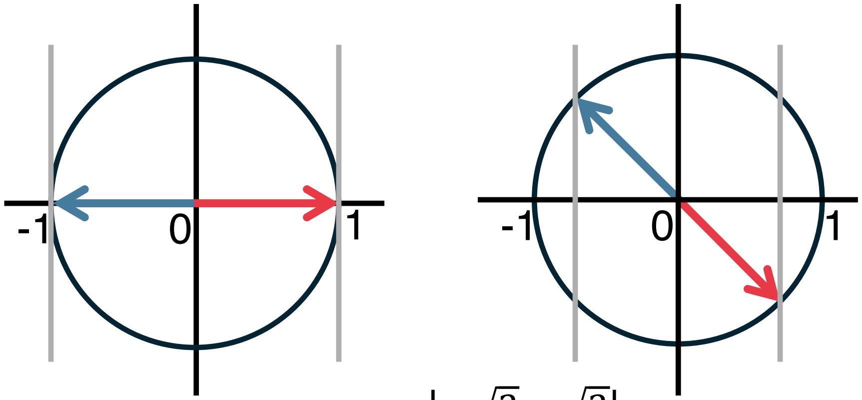


Quantization with global range is homomorphic



Optimizations for accuracy improvement

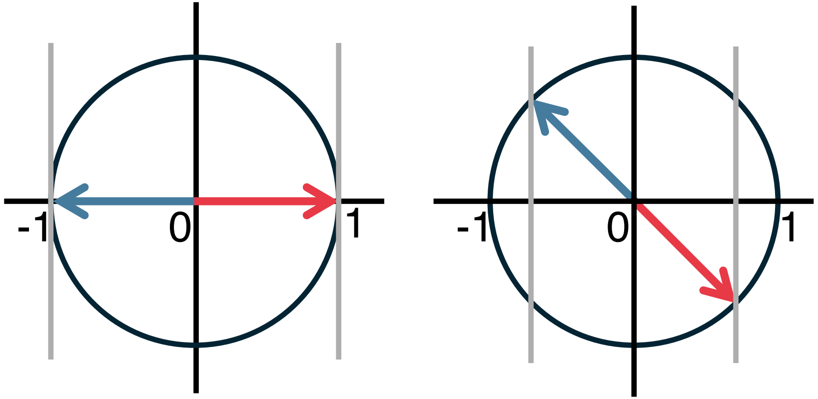
- Shrink the quantization range through Randomized Hadamard Transform (RHT)
- Intuition: “squeeze” values together before quantization to reduce the difference between min and max values.



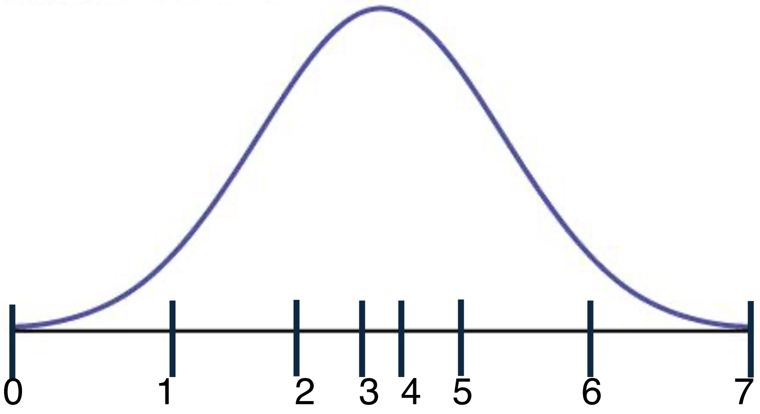
$$|-1-1| > \left|-\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}\right|$$

Optimizations for accuracy improvement

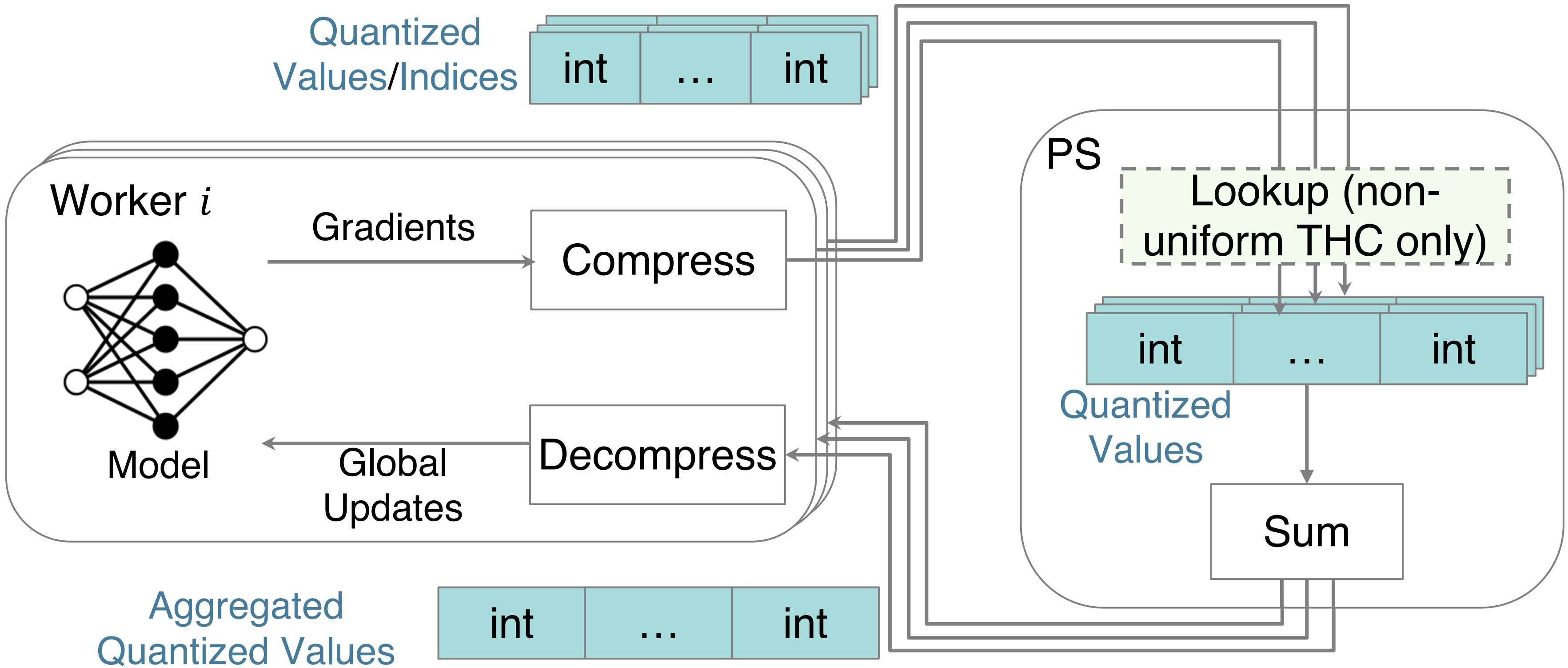
- Shrink the quantization range through RHT
  - Makes coordinates approach a normal distribution
  - Happens in parallel with global range alignment



- Non-uniform quantization
  - Enables more fine-grained quantized values
  - Convert non-uniform indices to uniform quantized values with a lookup table built offline

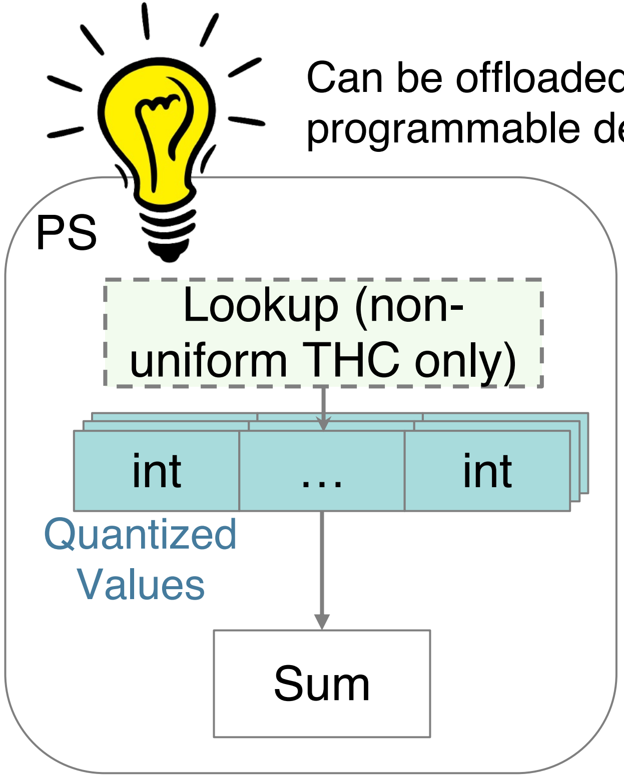


THC workflow



Easy In-Network Aggregation (INA) integration

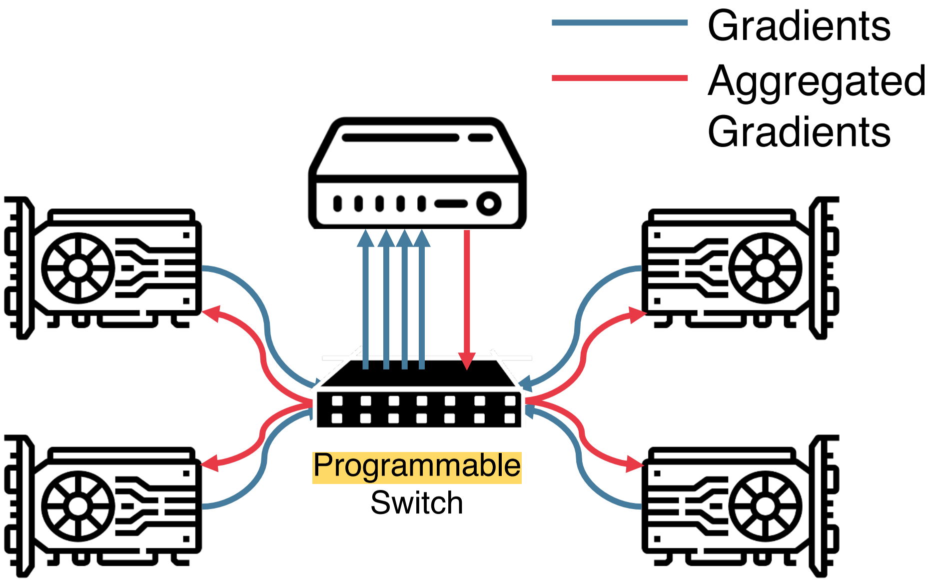
Can be offloaded to programmable devices



No complex (de)compress operations.

Integer operations only.

THC prototype uses INA with Programmable Switches



Remove traffic between the switch and PS.

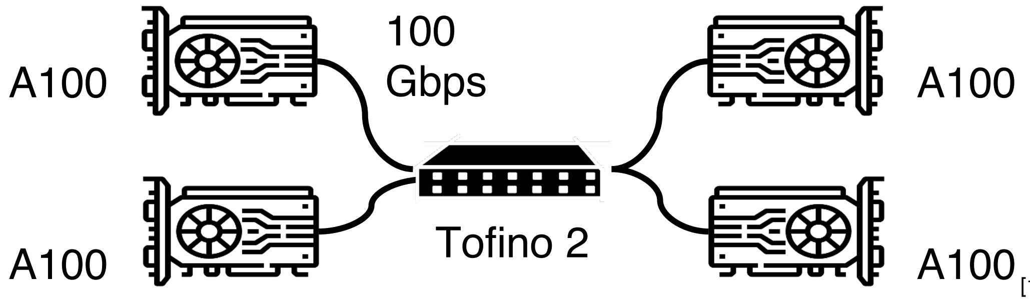
Evaluation setup

Models: **VGG16**; RoBERTa-base, GPT-2

Baselines: Horovod without compression, DGC10%, TopK10%. All using RDMA.

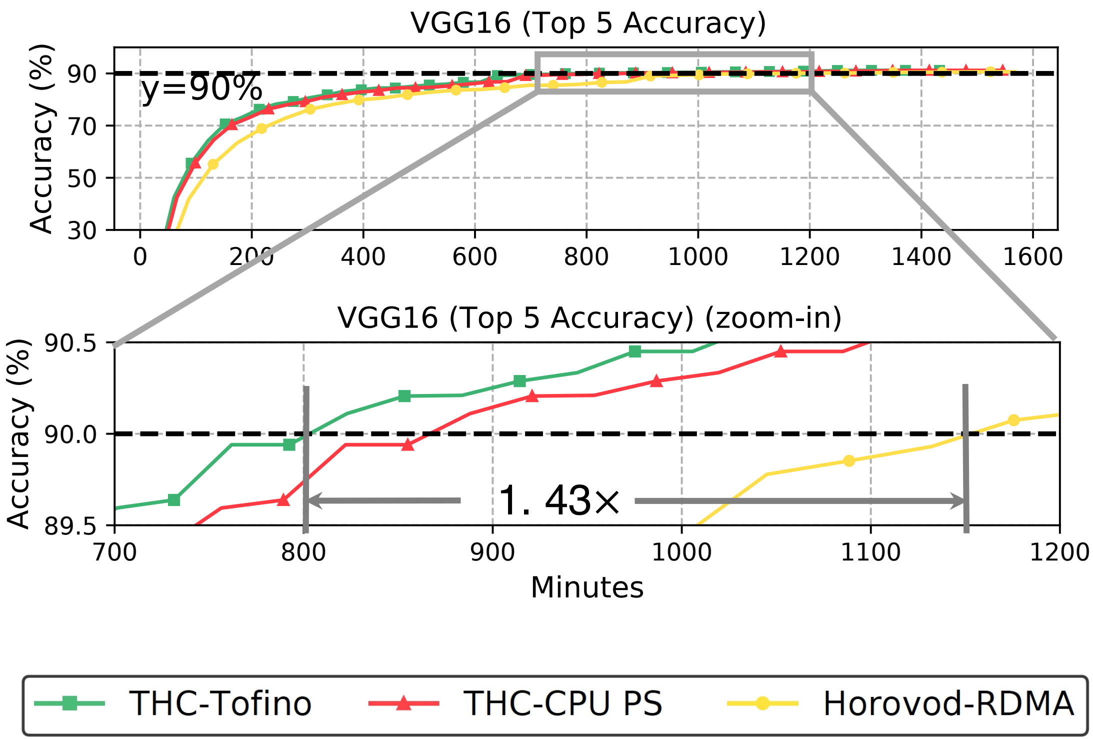
- Horovod: SOTA AllReduce (bandwidth optimal in homogenous settings [1]) framework.
- DGC10% and TopK10%: communicate top 10% of coordinates by magnitude.

Testbed Setup:



[1] Pitch Patarasuk and Xin Yuan. Bandwidth Optimal All-reduce Algorithms for Clusters of Workstations. Journal of Parallel and Distributed Computing, 2009.

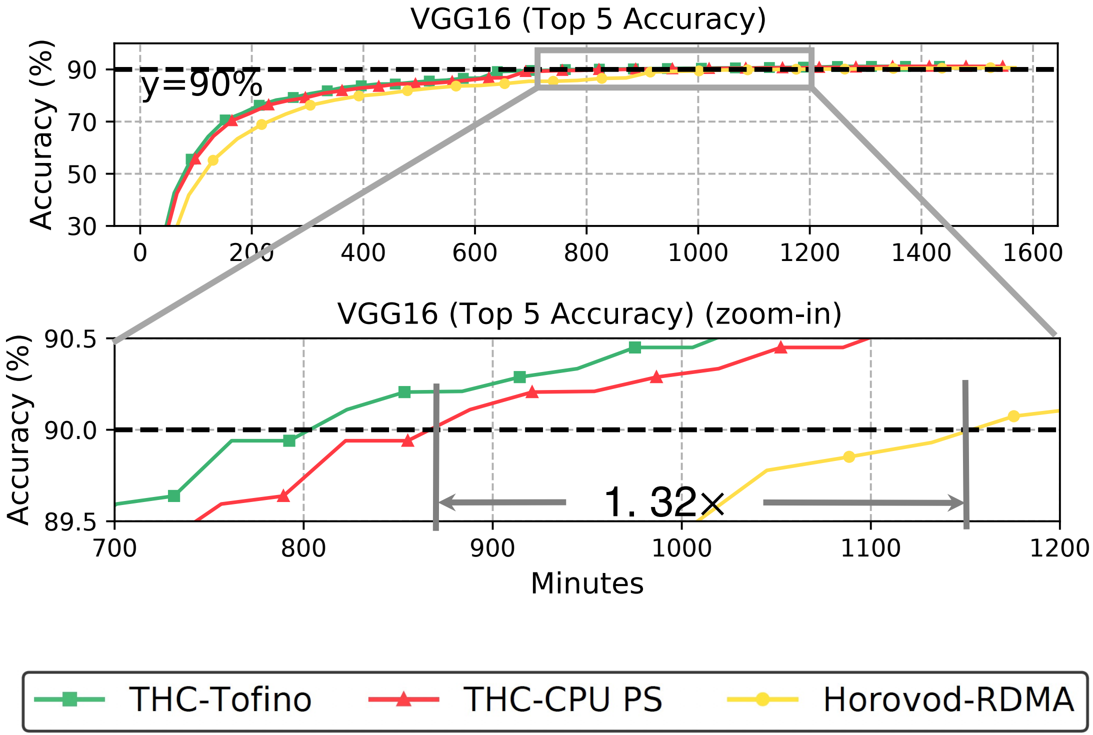
Evaluation: Time-to-Accuracy



THC-Tofino reaches the target accuracy 1.43× faster than the Horovod-RDMA baseline through

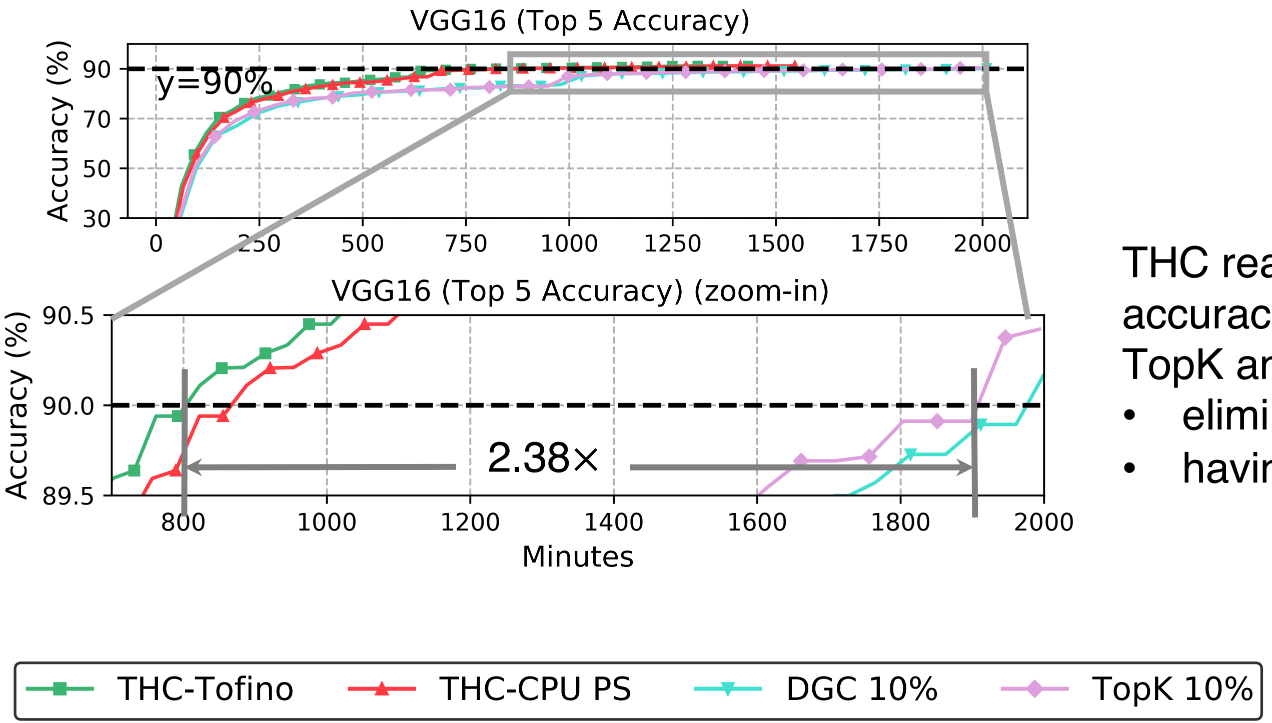
- 4 to 8× compression ratio
- INA speedup

Evaluation: Time-to-Accuracy



THC-CPU PS reaches the target accuracy 1.32× faster than the Horovod-RDMA baseline.

Evaluation: Time-to-Accuracy



THC reaches the target accuracy 2.38× faster than TopK and DGC by

- eliminating PS overhead
- having a lower error.

Additional results in paper

- THC Scalability
  - Large scale experiments with 64 GPUs on AWS (up to 1.16× better than no-compression baseline)
  - Simulations for up to 64 workers and comparisons with QSGD
- Other models
  - Vision models: VGG models, ResNet models
  - Language models: RoBERTa, BERT, Bart, GPT-2
- Other system opportunities
  - Stragglers handling
  - Packet loss

Conclusion

- Networks take an increasingly large portion of distributed training time.
- Tensor Homomorphic Compression (THC) is a novel scheme that enables direct aggregation on compressed data.
- THC offers up to 1.47x time-to-accuracy speedup, is scalable, and supports in-network aggregation.
- THC is integrated into BytePS and accessible at https://github.com/SophiaLi06/BytePS_THC.git

Thank you for listening!
Q&A