

When Does Offloading FSDP's Reduce-Scatter to a DPU Actually Pay Off?

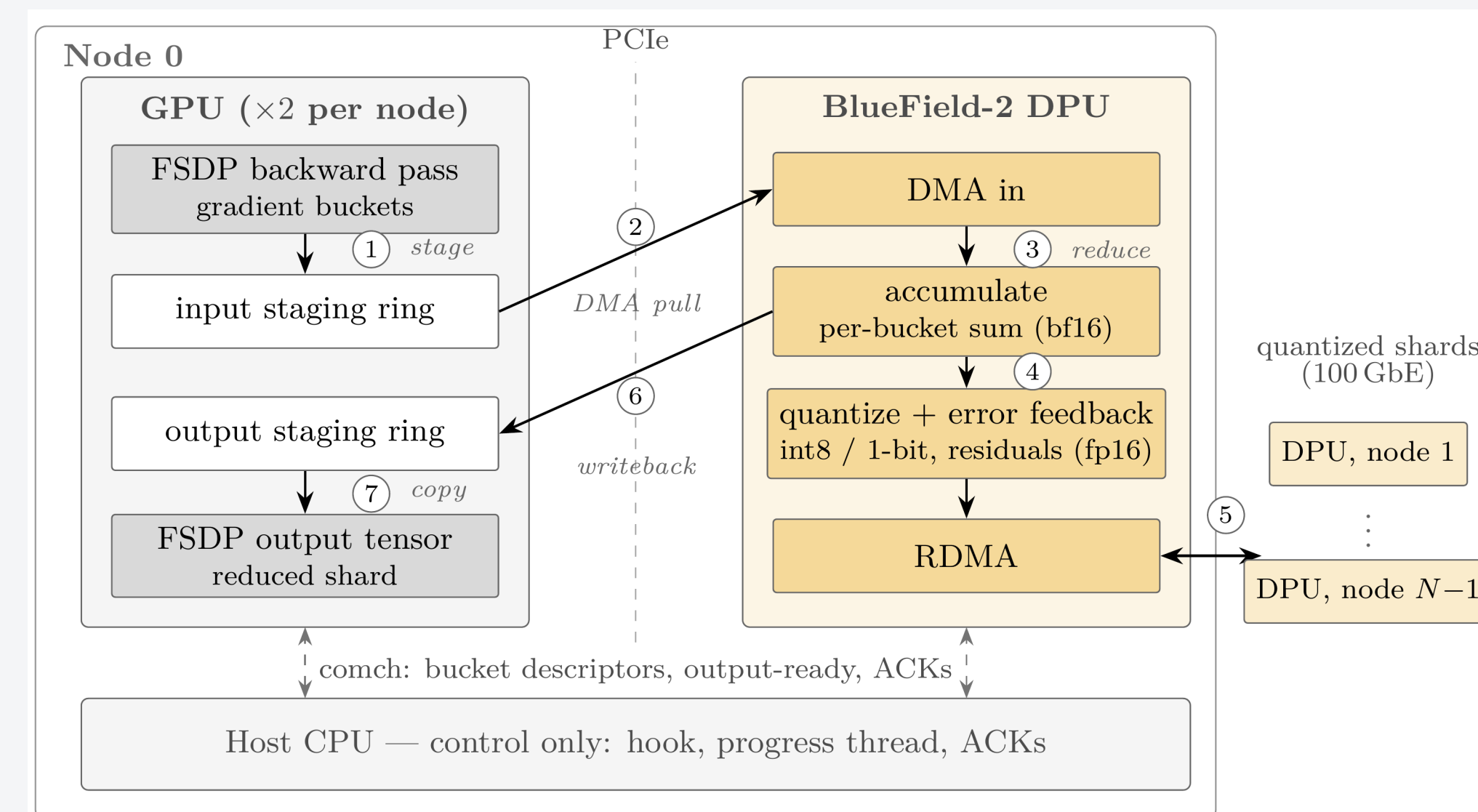
Venkatesh Bellale, Ryan Stutsman, Vivek Srikumar, Robert Ricci
University of Utah

Motivation

FSDP allows large models to train across multiple GPUs, but gradient exchange over the network can slow the training. We offload gradient reduction and compression to DPUs and measure when it improves model training speed without degrading model quality.

Architecture

1. A PyTorch hook sends gradient buckets to the DPU.
2. The DPU reduces and quantizes with error feedback, then exchanges shards over RDMA.
3. Reduced shards return to the GPU.



all-gathers remain unchanged.

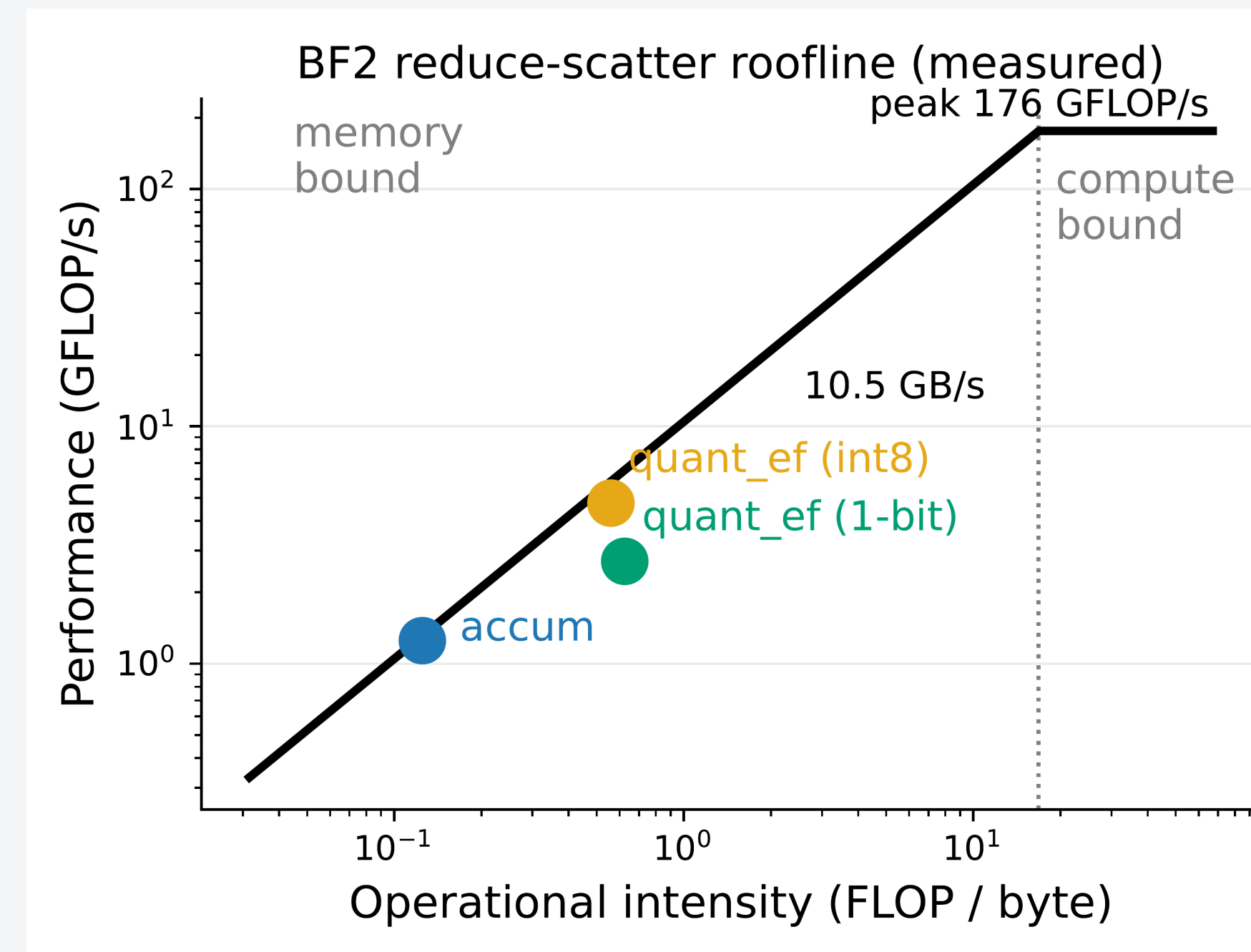
Does offload preserve model quality?

Model	Steps	NCCL Val Loss	DPU Val Loss
1.5B	5000	5.731	5.700
2.6B	5000	5.953	5.815

int8 + error feedback, lower validation loss is better.

No observed validation quality degradation in these runs.

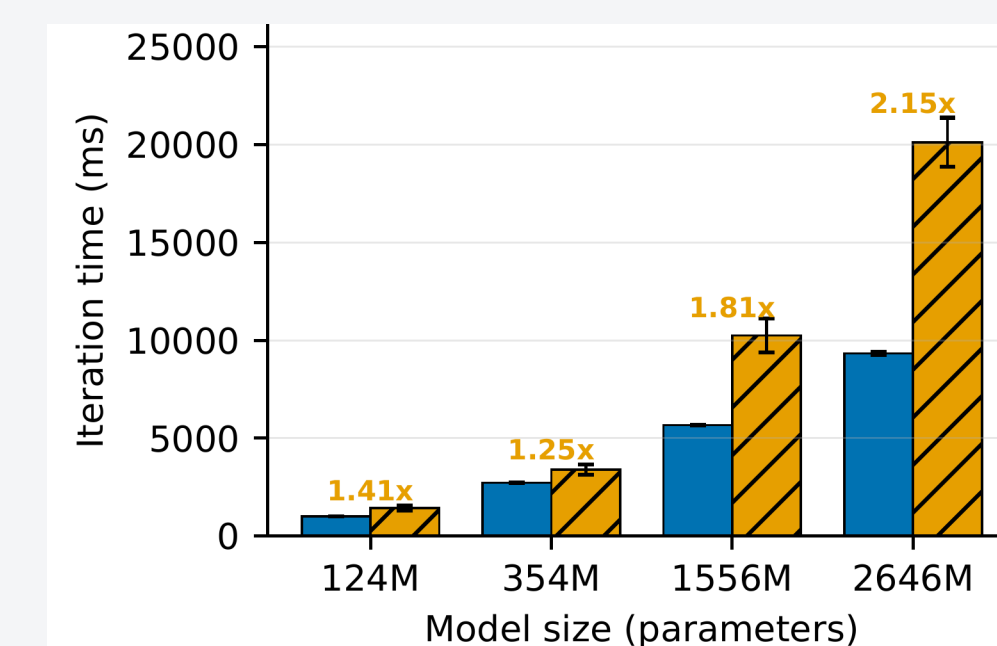
Why is offload slower?



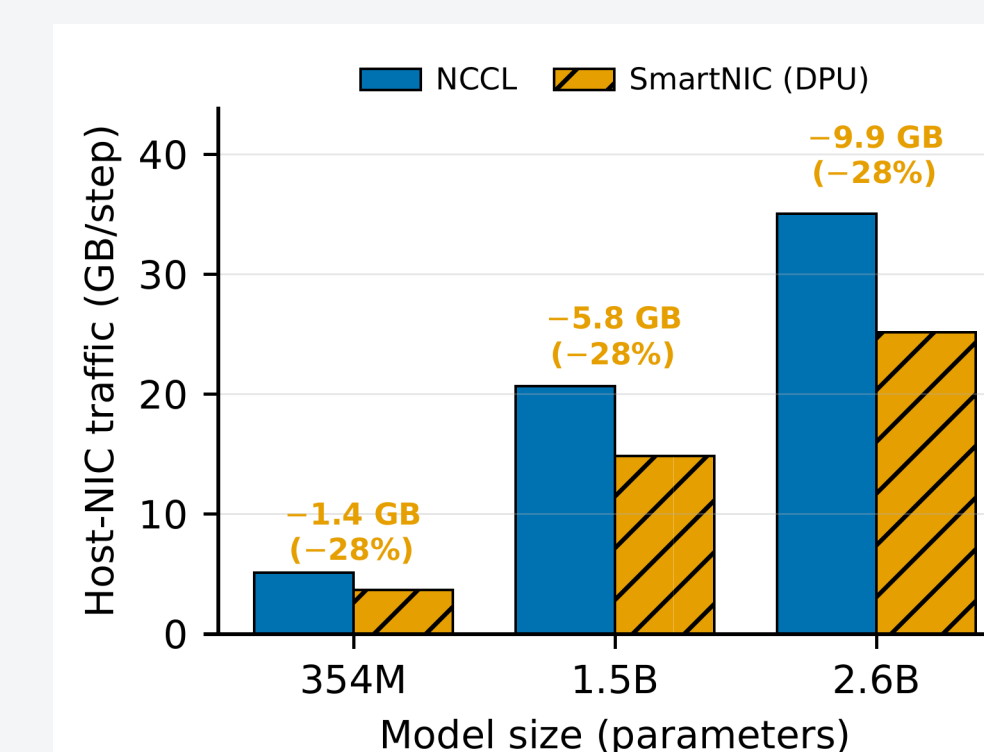
Reduction and quantization approach the DPU's memory-bandwidth limit, far below its compute peak.

100 Gb/s: NCCL fp32 vs. DPU int8

Training step time



Measured host-NIC traffic



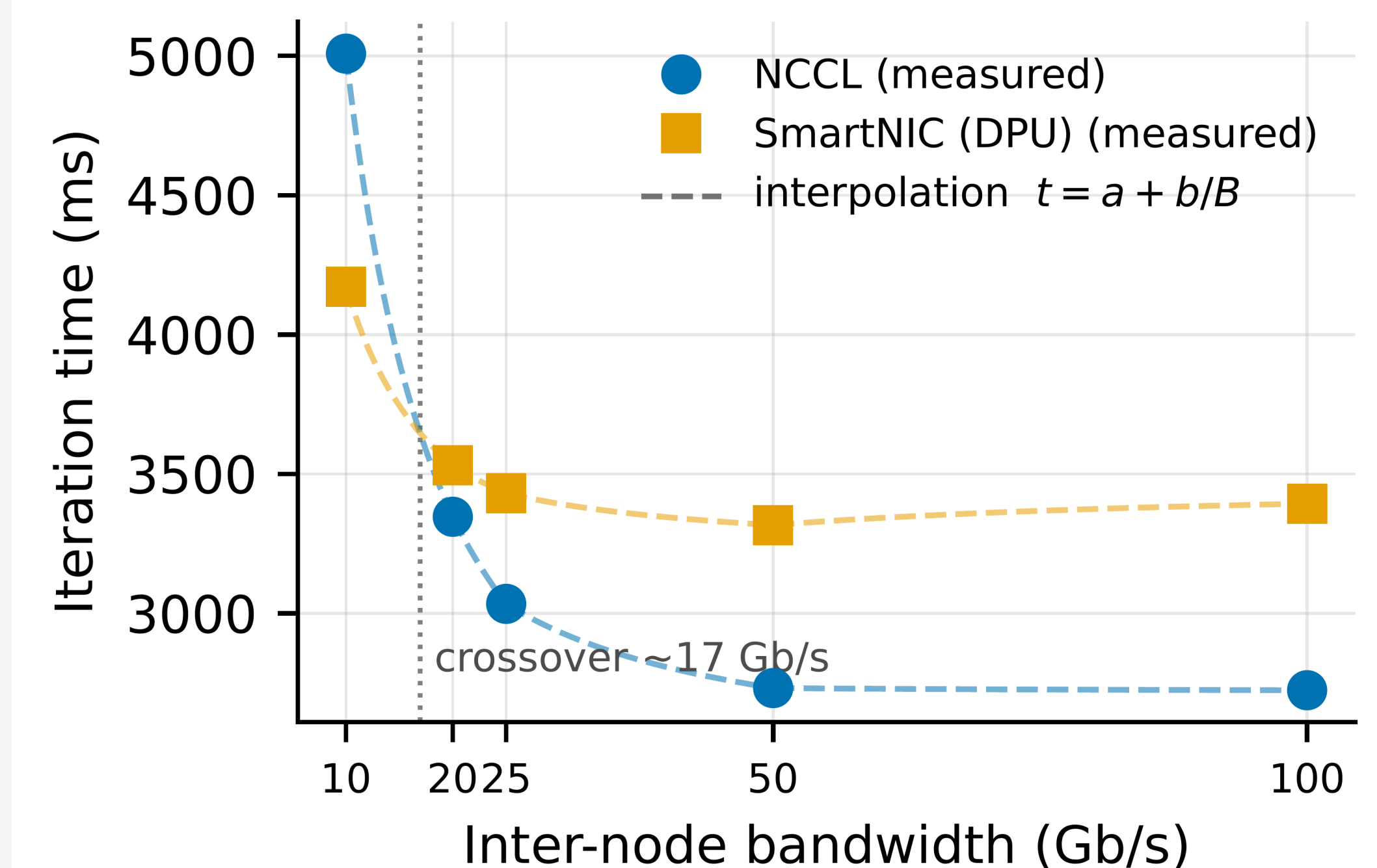
At 100 Gb/s, DPU offload reduces host-NIC traffic by about 28%, but training steps take 1.25 to 2.15 times as long as with NCCL.

Traffic is measured per node per training step using NIC hardware counters.

Setup

4 nodes, each with 2 V100 GPUs and a BlueField-2 DPU.
GPT-style models: 124M, 354M, 1.5B, and 2.6B.

When does offload help?



354M model: DPU int8 vs. NCCL fp32

The estimated int8 crossover is about 17 Gb/s for 354M.

1.20x faster at 10 Gb/s for 354M
1.09x faster at 10 Gb/s for 1.5B

Offload helps when the network is slower than the DPU's processing rate.

Processing ceiling $\sim B_{\text{mem}} / k$.
 B_{mem} : DPU memory bandwidth.
 k : DRAM traffic per gradient byte (about 5 here).

