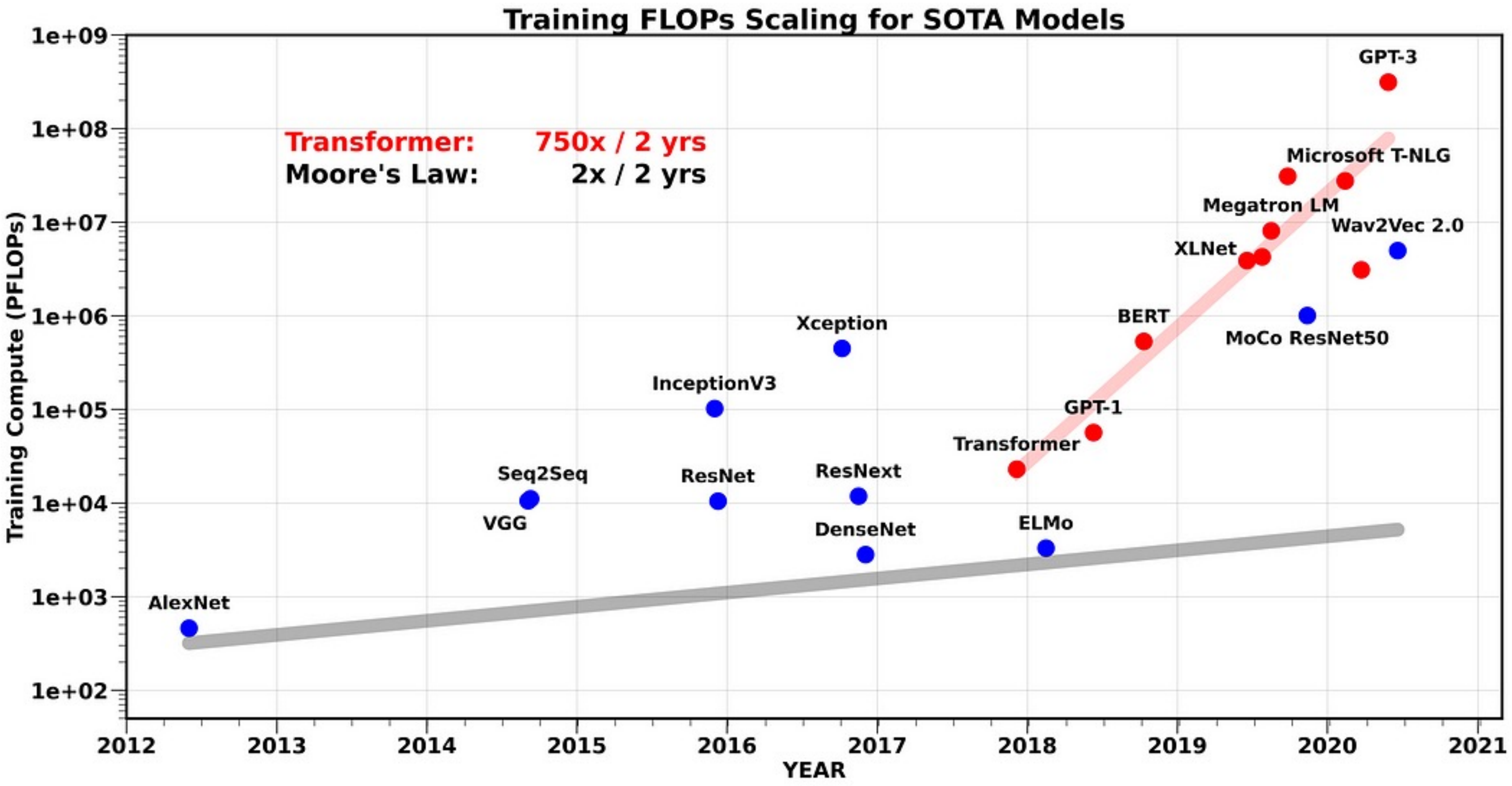
AI data center

Omer Shabtai, SW Arch

Agenda

- Distributed AI training scale and patterns
- SO vs SU
- AI Collectives
- GPU direct RDMA
- Goals
- Evaluation focus

Single GPU vs workload Flops

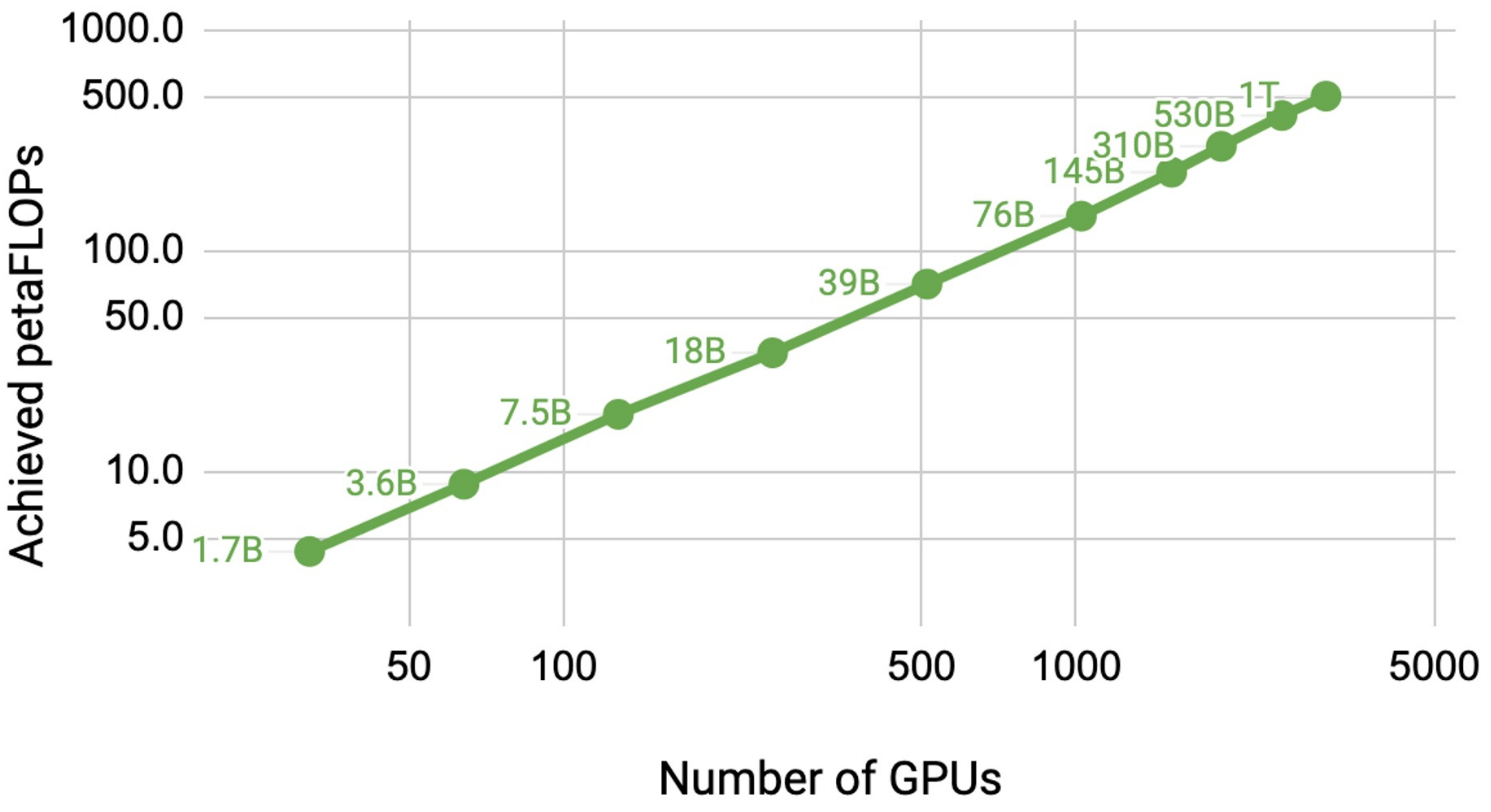


GPU Flops ~2X every 2 years
>> 32X over same period

Need for massive scale-up

AI training at scale

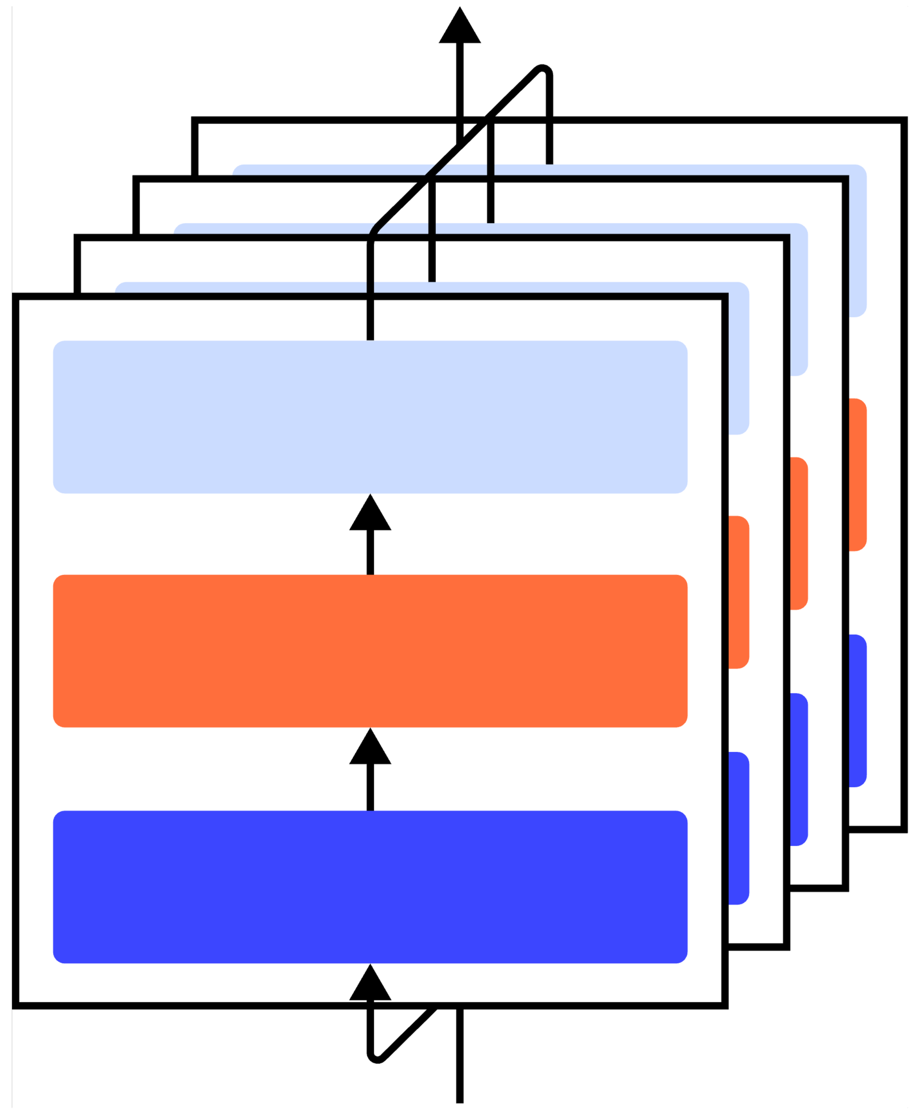
- Efficiency at scale
  - GPUs are strong
  - Goal – achieve linear scaling of training time to compute
- Synchronous training
  - Tail sensitive
  - Tightly coupled
  - Low Entropy
- computation / communication overlap
  - Induce complexity to framework and training



Parallelism strategies

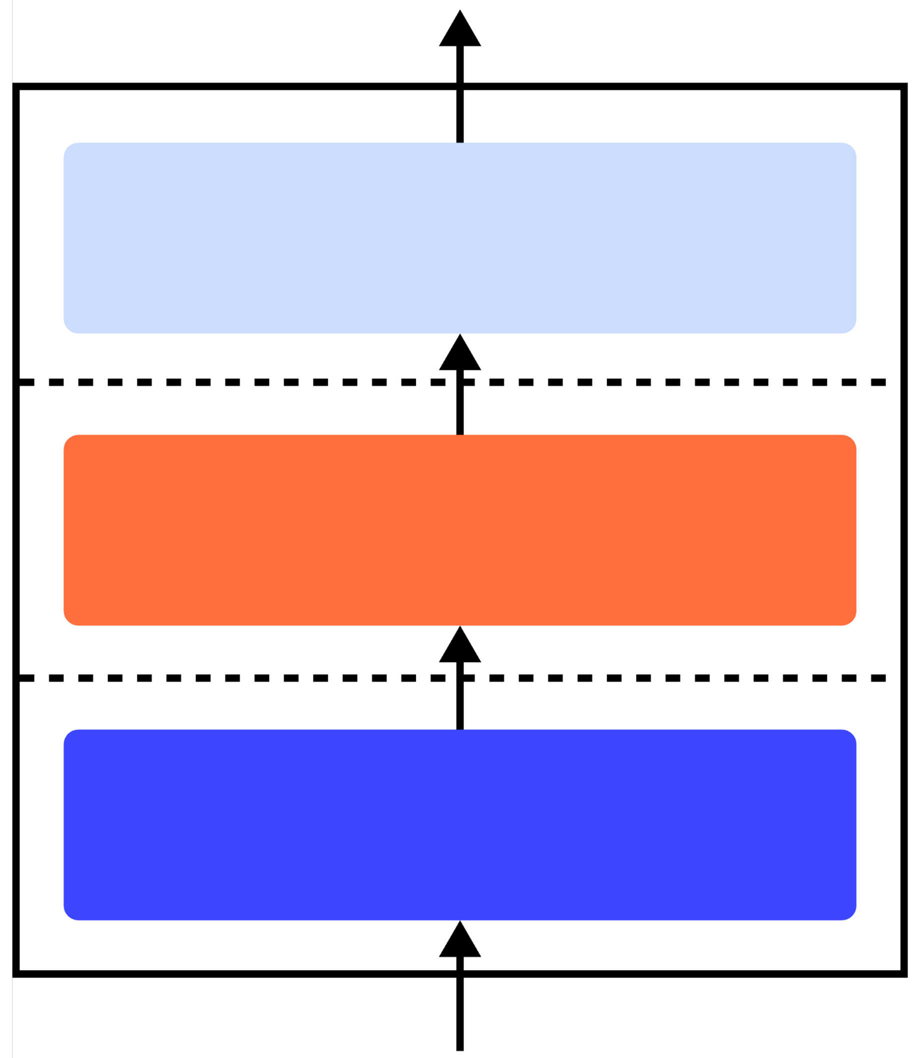
Model and data parallelism

Data Parallelism



Simple to implement

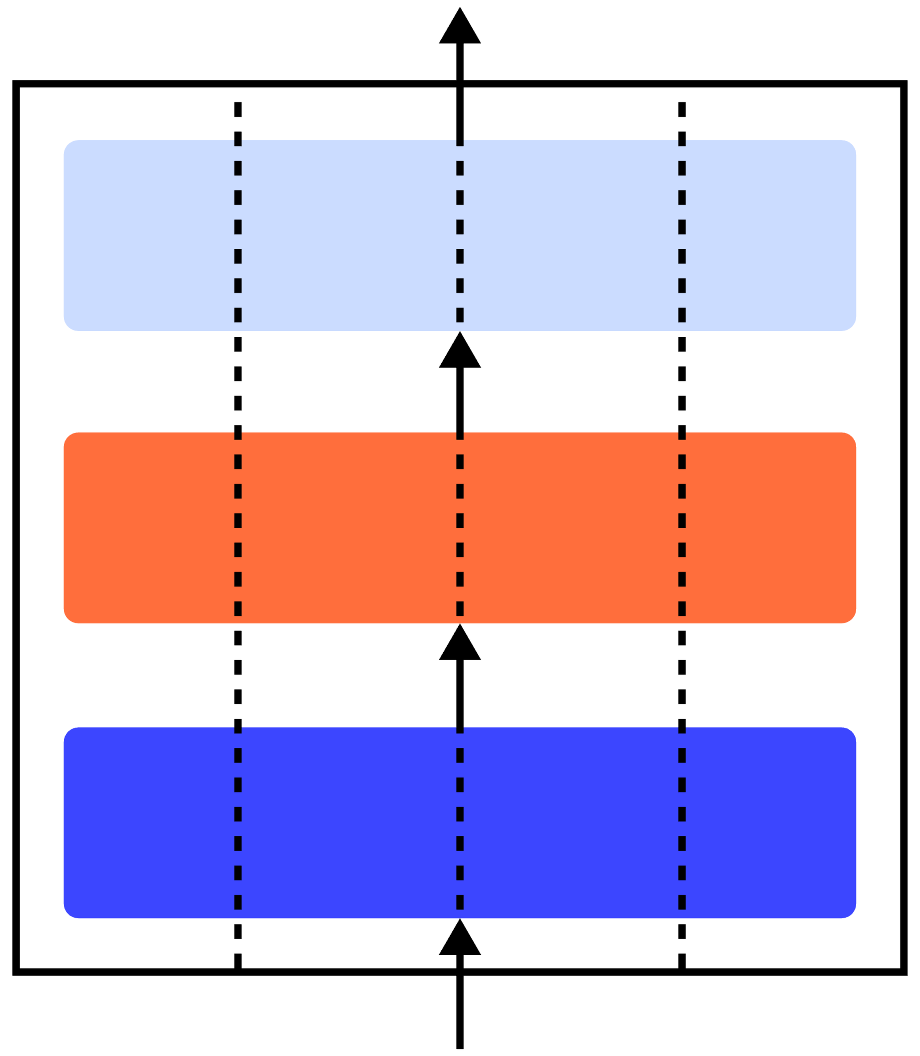
Pipeline Parallelism



Communication cheap

Good performance at larger batch sizes (pipeline stall amortized)

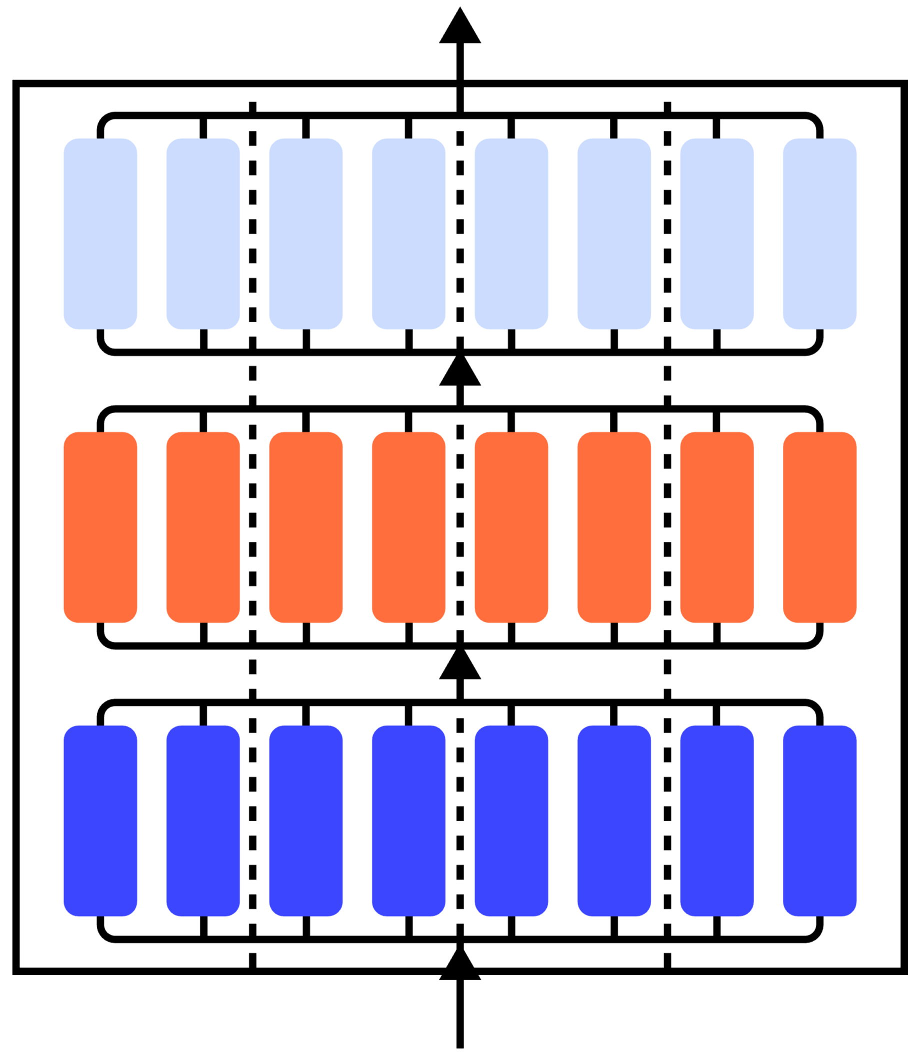
Tensor Parallelism



Communication expensive

Good performance across batch sizes

Expert Parallelism



Communication expensive

Scalable sparsity

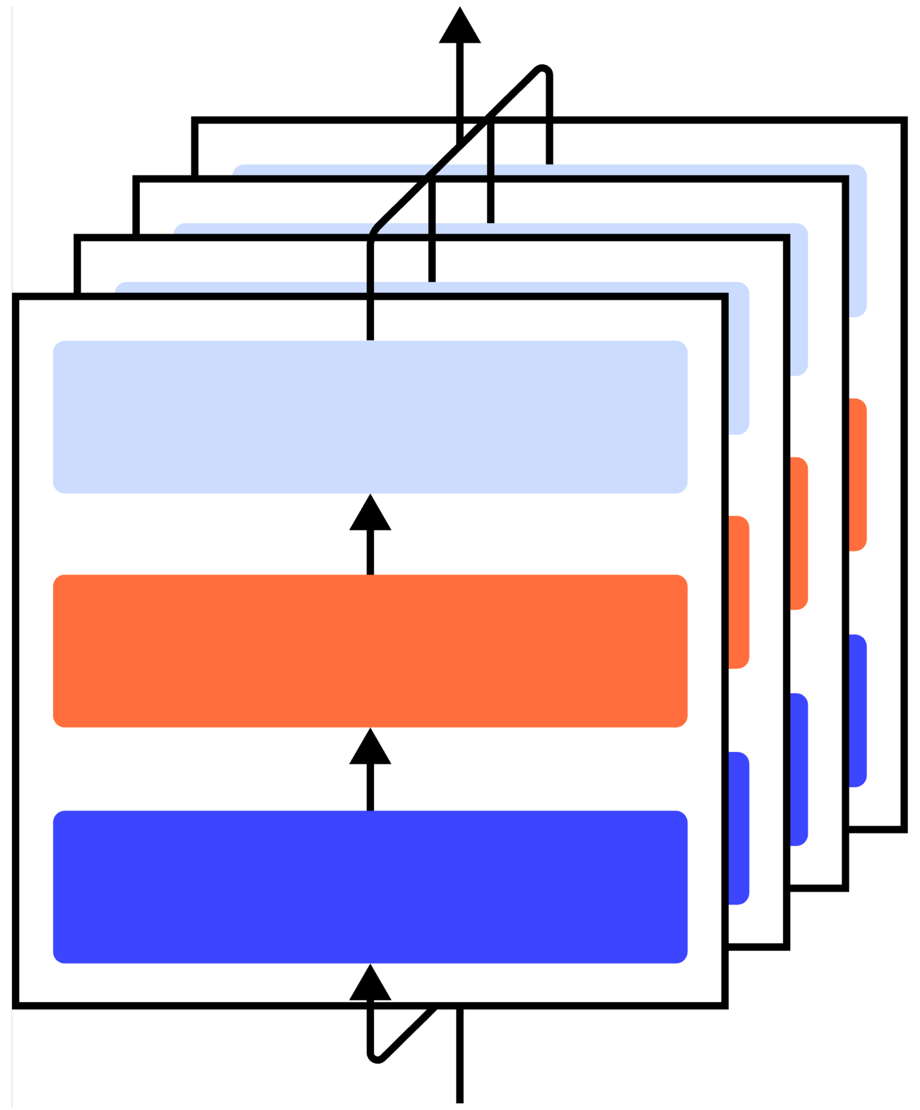
Large batch size → better GPU efficiency and speedup

Modern workloads - hybrid parallelism

Parallelism strategies

Typical Collectives

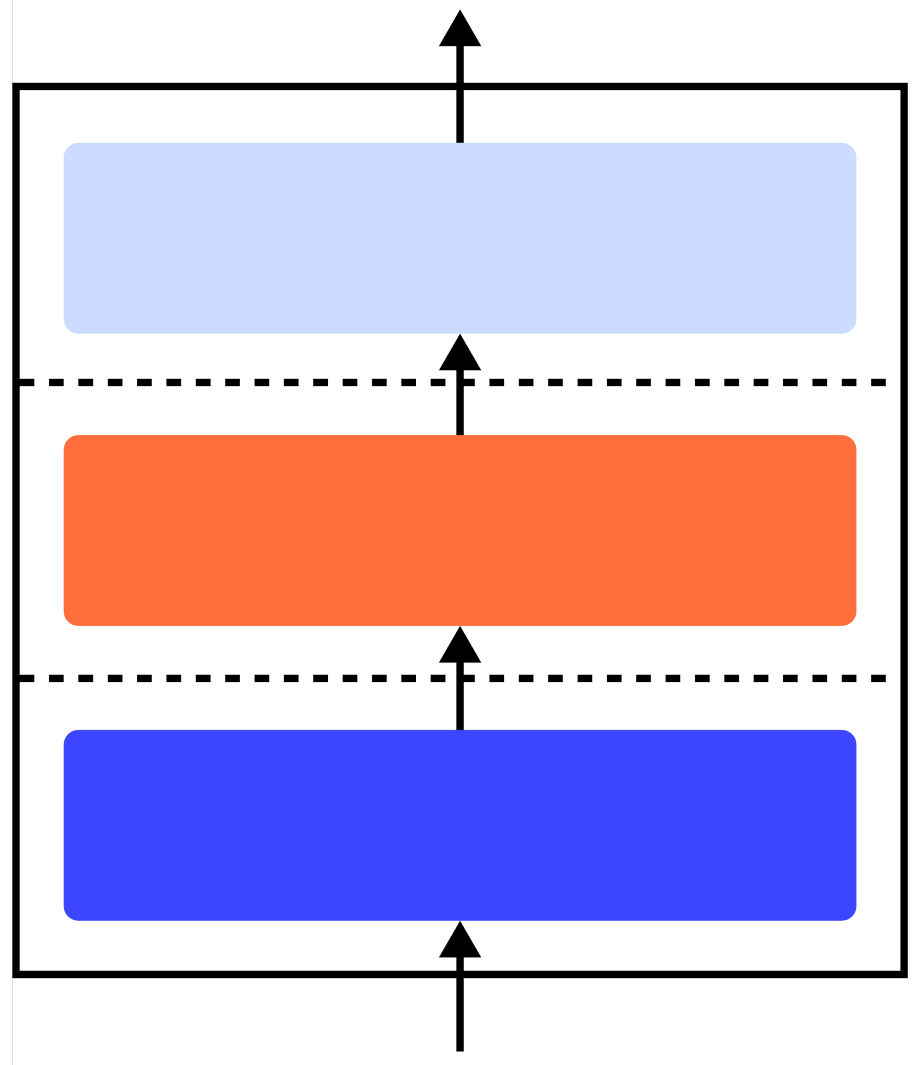
Data Parallelism



All-reduce
Or
All-gather + reduce scatter

Large (~GBs)
Tightly coupled
Low entropy

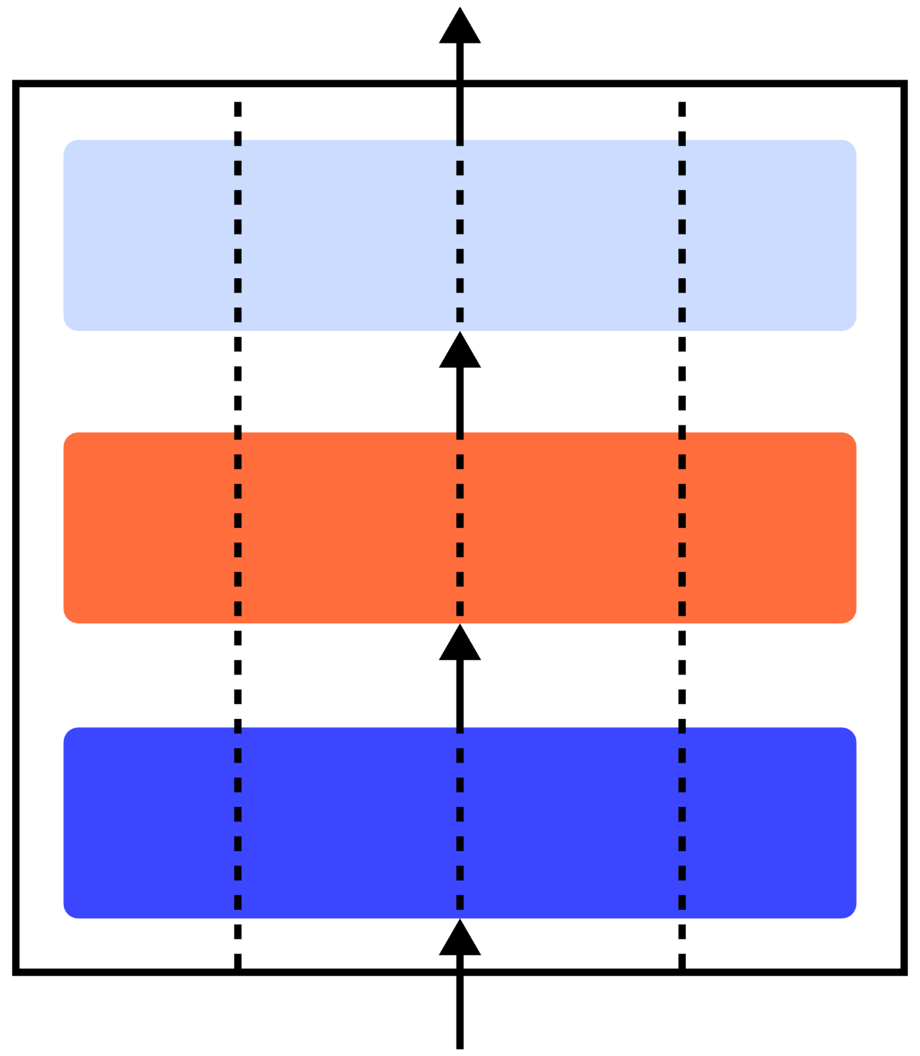
Pipeline Parallelism



Point to point

Small (~MB)
Latency sensitive

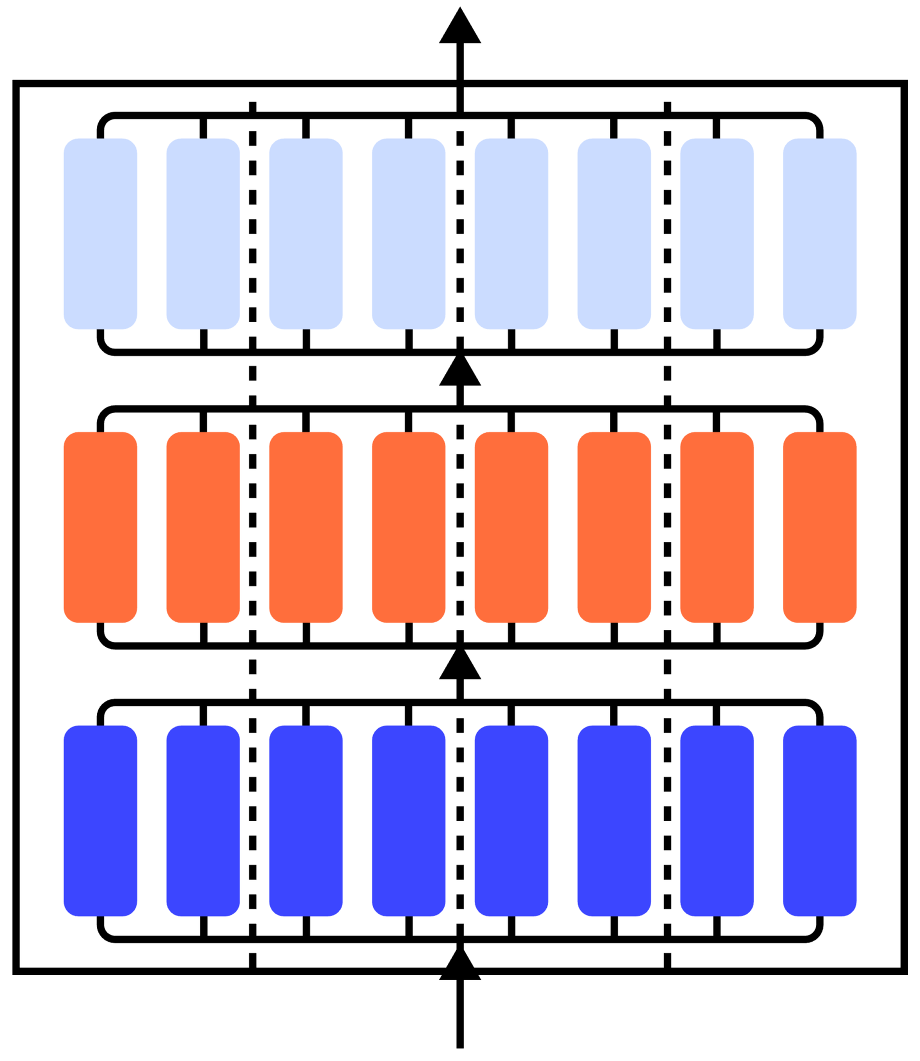
Tensor Parallelism



All to All, reduce scatter

Small, but frequent
Extremely latency sensitive
Scale-up only

Expert Parallelism



All to all

Med (<100MB)
congested

Popular parallelism Combinations

	Data parallel	Tensor parallel	Pipeline parallel	Distributed Models
Light models, CNNs, For edge	X			
DLRM	X			X
FDSP (LLM)	X			X
3D, GPT-3 like (LLM)	X	X	X	
MoE (LLM)	X	X	X	X

AI training paradigm is rapidly evolving

New approaches pop fast

Increasing perf by significant multipliers

Over same HW generation

Optimization usually done for best perf

Network resiliency and jitter are mostly neglected

Bad networking is usually exposed

but hardly detected

AI traffic patters

- **GPU comms**
  - Heterogeneous - Intra-node + inter-node
  - Single flow can saturate wire speed
  - Low flow count - minimize GPU’s management resources (SMs)
  - Limited msg size – minimize GPU’s expensive buffers

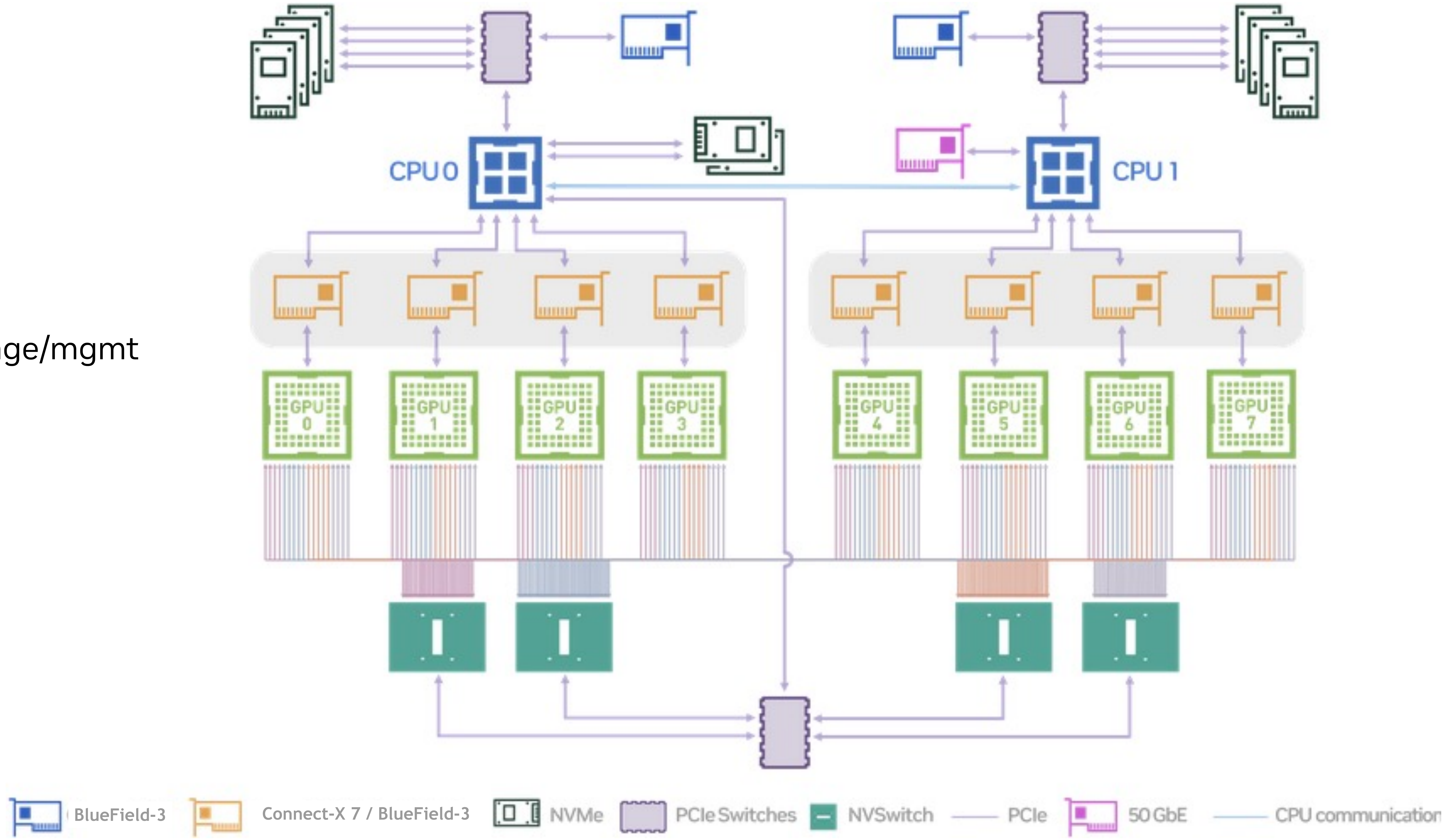
- **Collectives**
  - Synchronous
  - Reliable
  - Some incast guaranties

  Tightly coupled, both BW and tail latency sensitive

- **Accelerated transport - RDMA**
  - Zero copy – mem BW
  - Low Latency – GPU direct

HGX/DGX - The AI building block

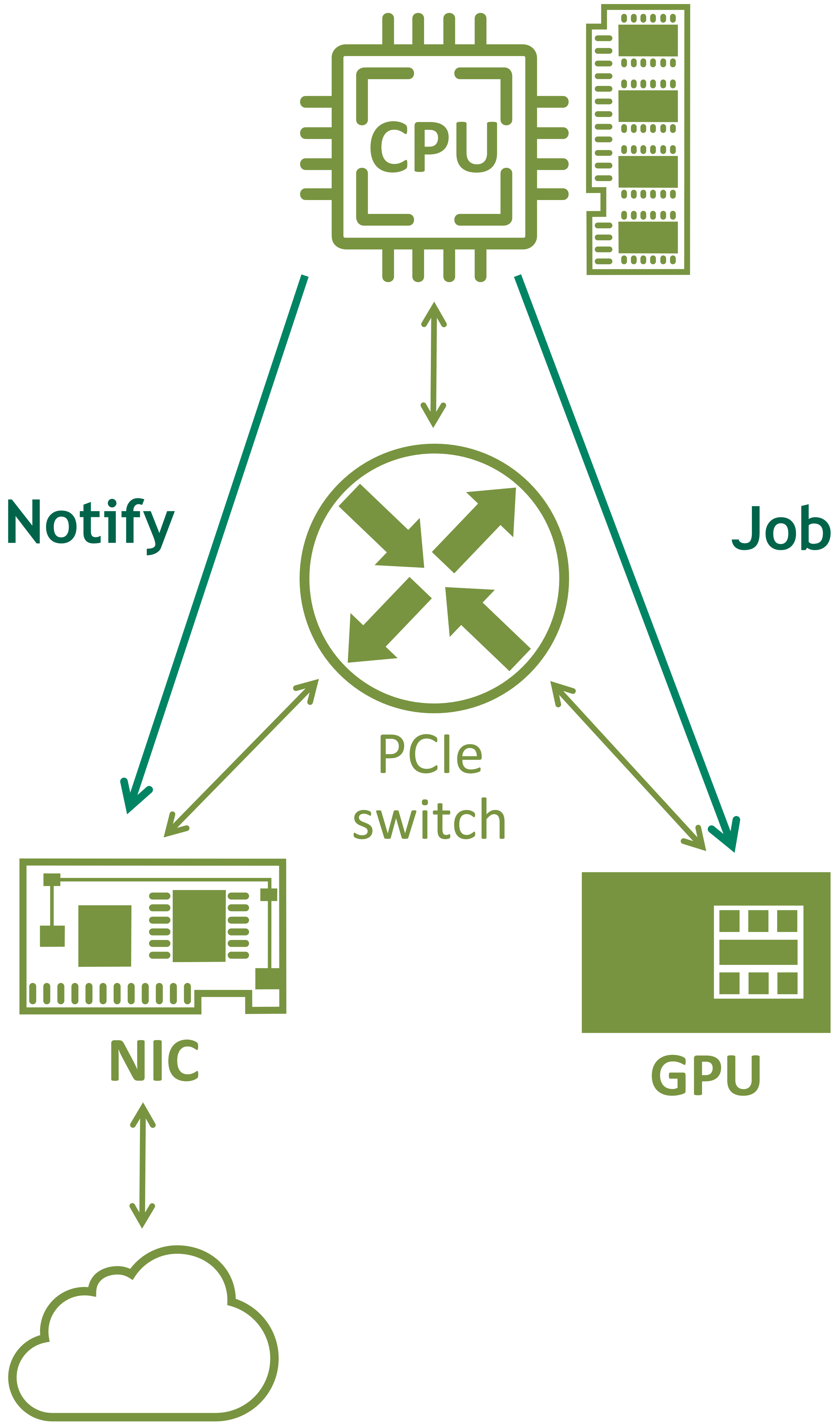
- Nvlink for scale-up
  - 7.2 Tbps GPU-GPU, full mesh
  - in-network aggregation
- 1:1 DPU/GPU for scale-out
  - 400gbps per GPU
- Dedicated interfaces for storage/mgmt



GPU Direct RDMA

Quick overview

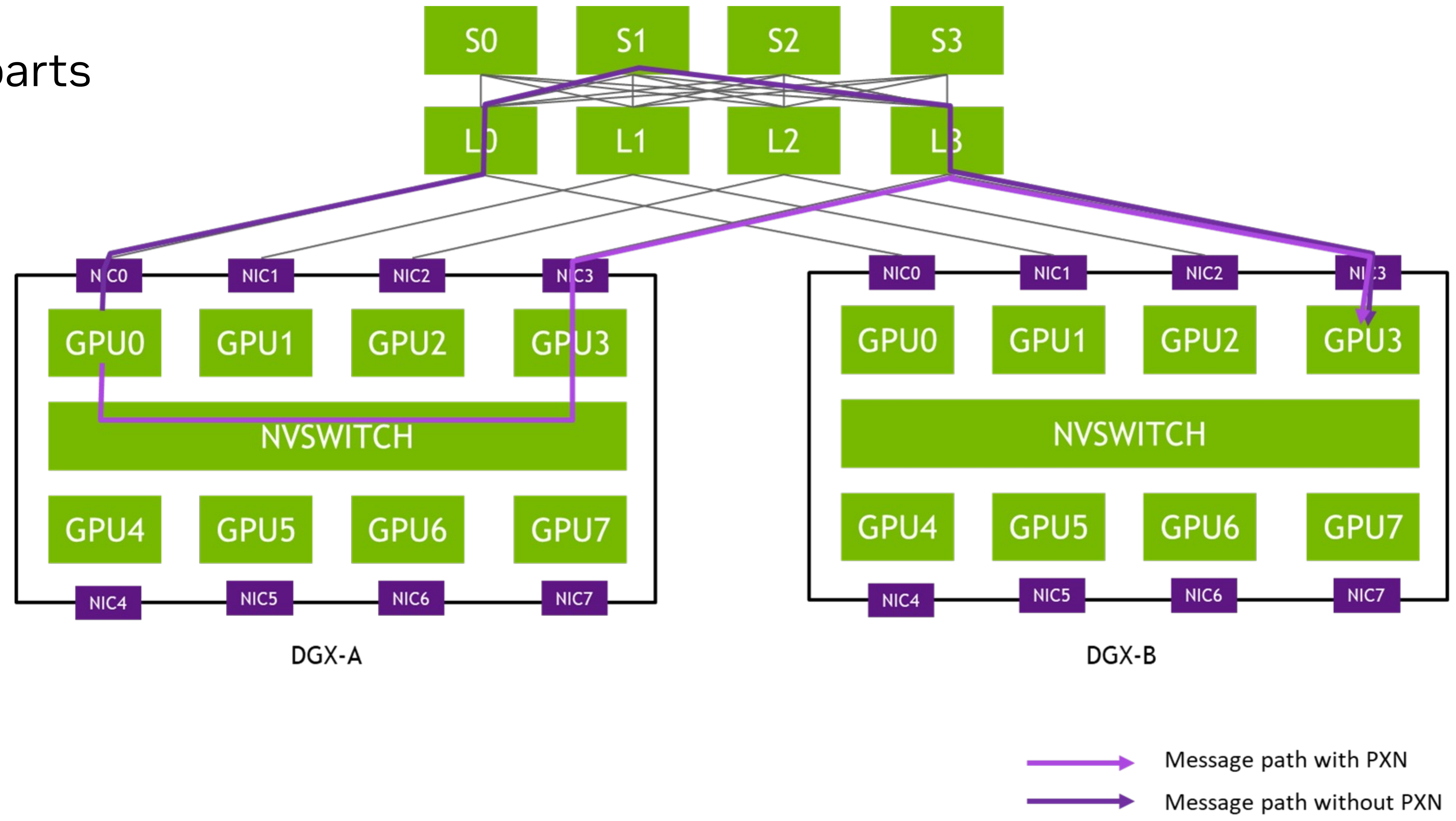
- GPU Direct RDMA
  - CPU submit a job to the GPU
  - GPU executes the job
  - CPU notifies NIC that the data is ready
  - NIC reads the data from the GPU's memory
  - NIC sends the data to the network



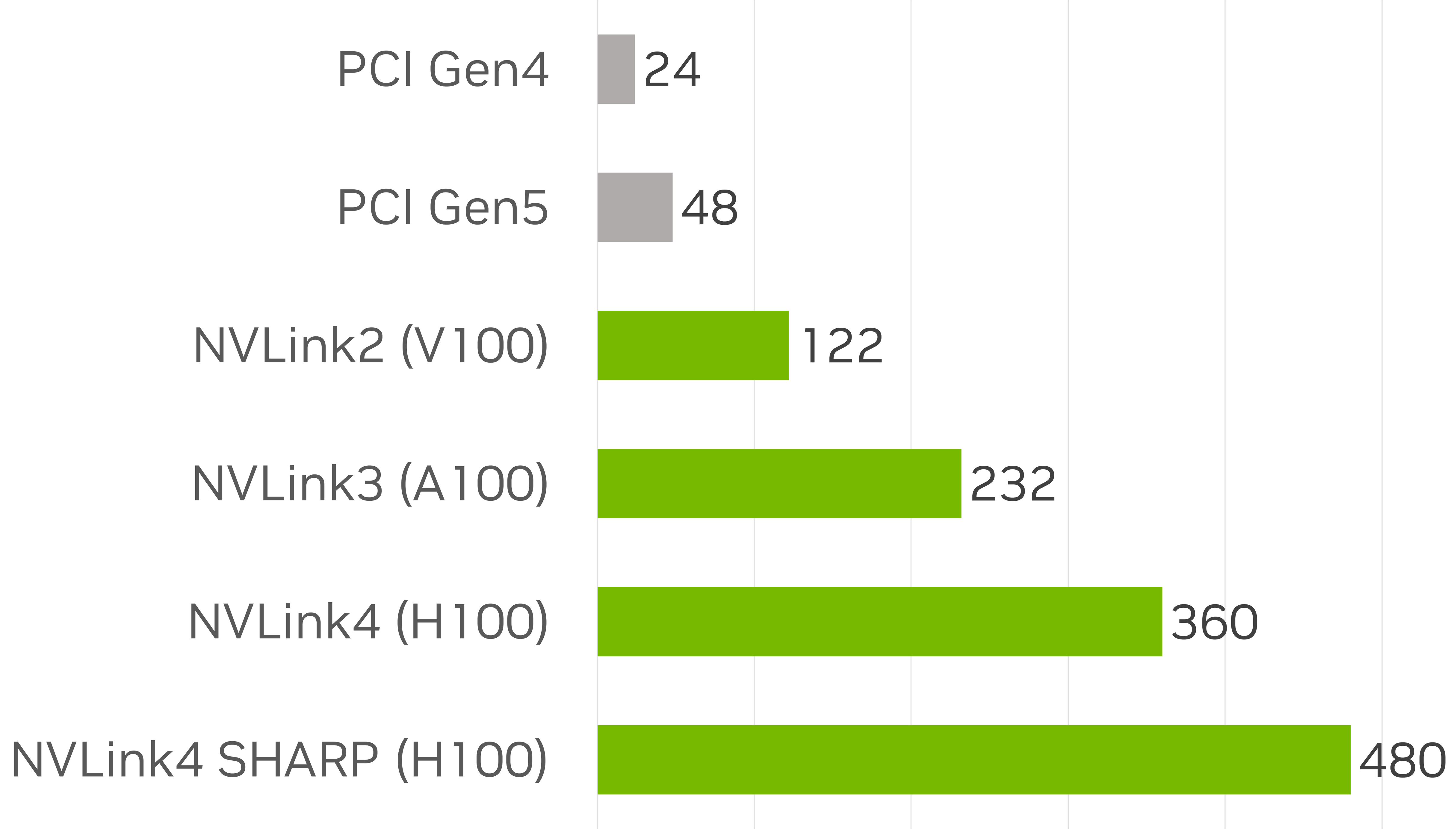
Scale up vs scale out

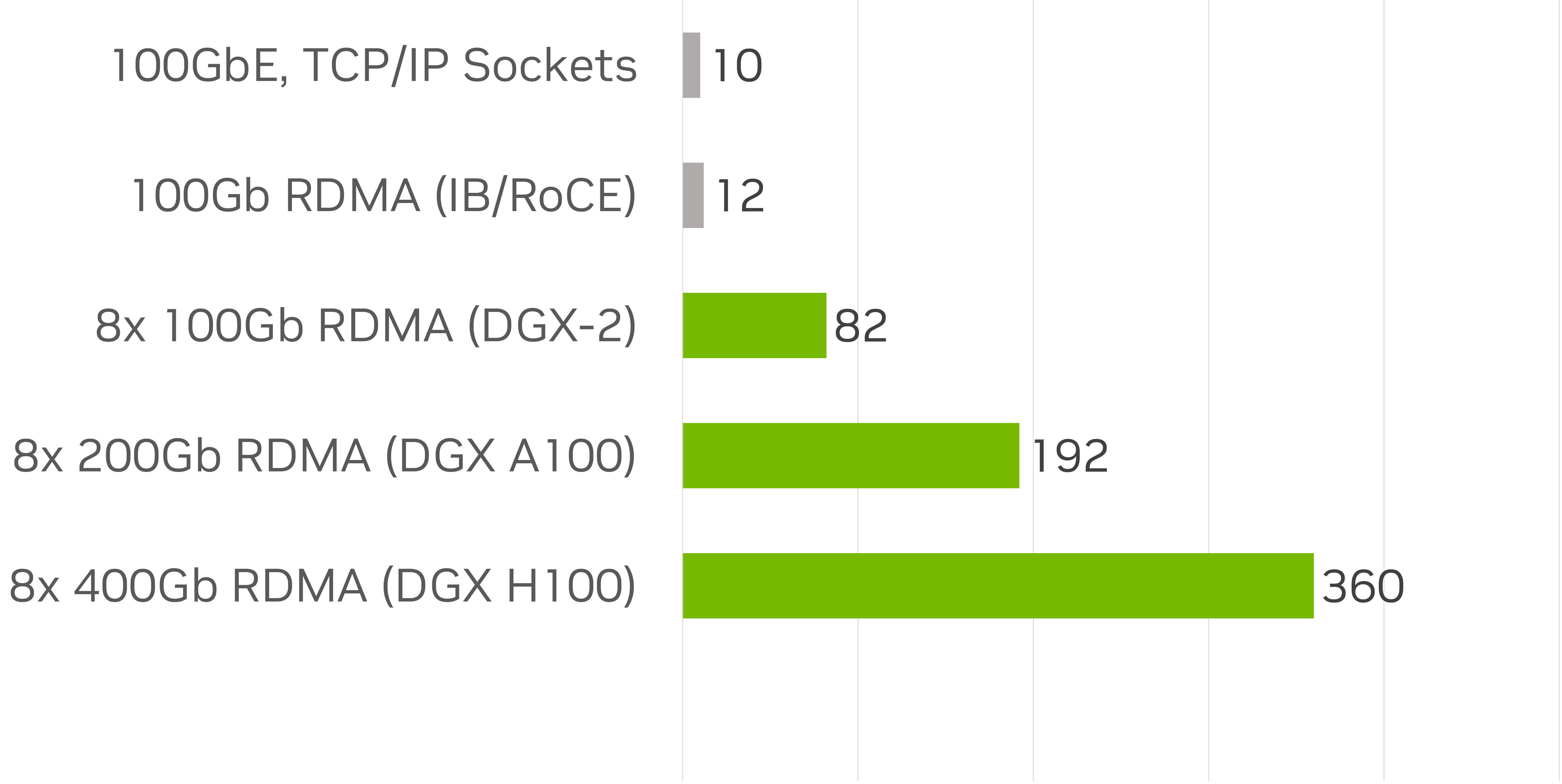
- Nvlink provides ~9X BW at lowest latency
  - All-reduce, all-gather speed-up of up to (8X in Hopper)
  - Tensor parallel – mandatory
  - Message aggregation for all to all

- Implications
  - Inter-node GPUs communicate only with their adjacent counter-parts
  - Rail optimized topology for better latency and scale
    - Increase 1-hop reach by 8X
    - Better isolation and latency guarantees



NCCL Bandwidth





NCCL Tests Allreduce Bus Bandwidth in GB/s

AI datacenters fabric requirements

Any work load, any location

Full cluster utilization

Full Bisection BW,
Latency/jitter sensitive

Allocation agnostic,
Workload agnostic

Expensive
Virtualized

A-symmetry resiliency
Global awareness

Tightly coupled
collective operations

Performance isolation
(chatty neighbor)

Adaptive Routing and Congestion Control

A generic solution

- End to end solution, Per packet load balancing for RDMA
  - Up to 95% bisection BW utilization, at low tail latency
  - AR Convergence time << Congestion control convergence time
  - Global Fabric asymmetry and failiure support
  - Zero-copy OOO delivery – reduce dependency on tail events.
- Topology agnostic
  - Asymmetric topology
  - fragmented job allocation
- Workload agnostic
  - Doesn’t relay on defined patterns like rings/trees
- Congestion control focus:
  - Incast scenarios
    - Collective first
  - Blocking fabric (mainly failures)
    - Slow forming

AI Training networking infrastructure

Evaluation

E2E Training
- DDP
- LLM (3D, FSDP, MoE)
- DLRM
- …

Collectives
- NCCL
  - P2P,A2A,AR,AG,RS

RDMA
- Bisection BW and latency under full load

Resiliency
- Link failiure
- Slow receiver
- BER

Workload Isolation
- Congestion-free noise
- Congested-prune noise

Single workload
- Consistency
- Allocation agnostic

NVIDIA