
2023 OFA Virtual Workshop

# Designing Networking Stacks for ML Frameworks

**Raghu Raja**
**raghu@enfabrica.net**

enfabrica

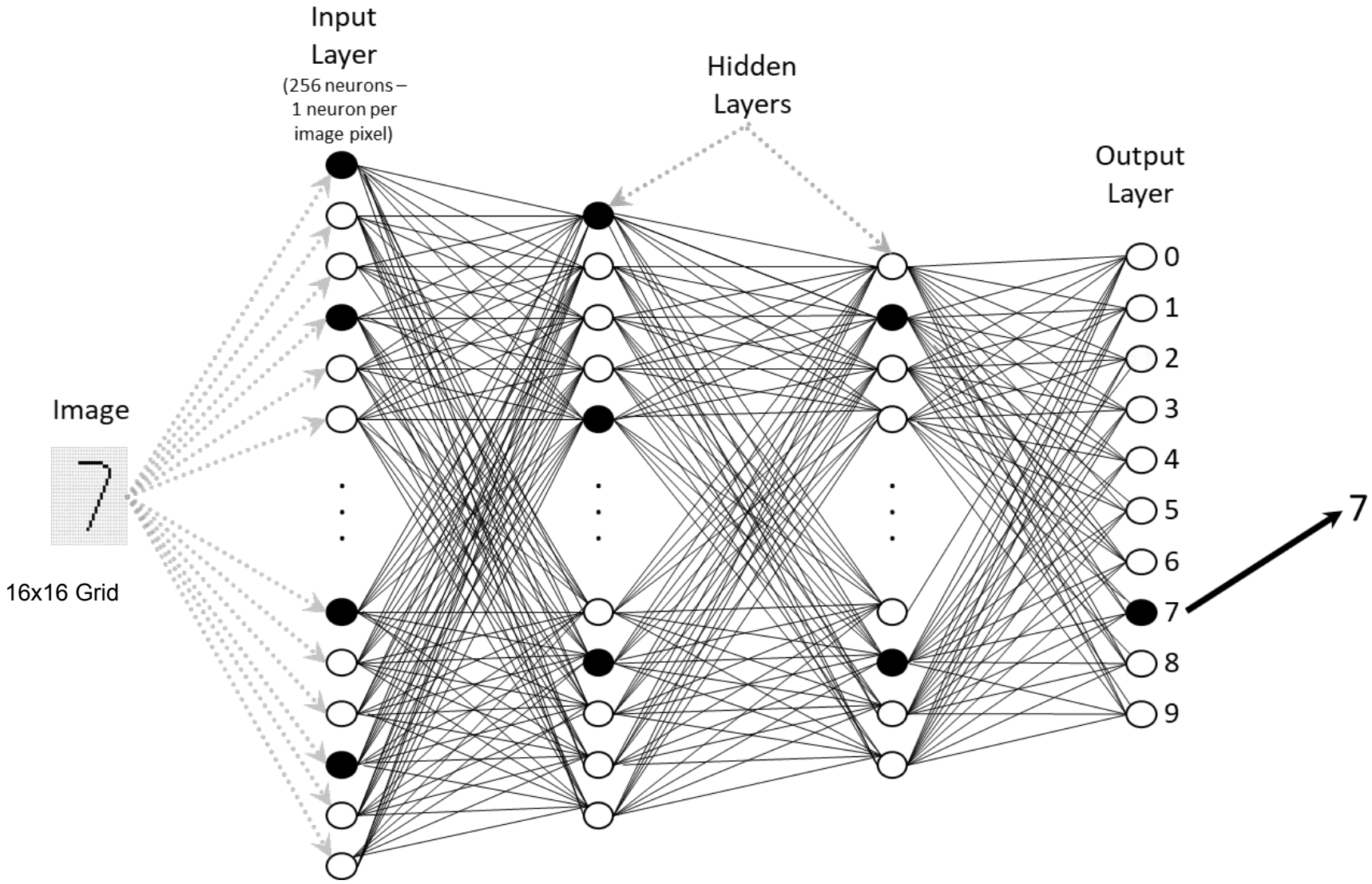
# Agenda

enfabrica

- Understanding Neural Networks
- Taxonomy of ML techniques
- New world, old problems - Design Challenges to consider
- Case study with NCCL
- Call to action

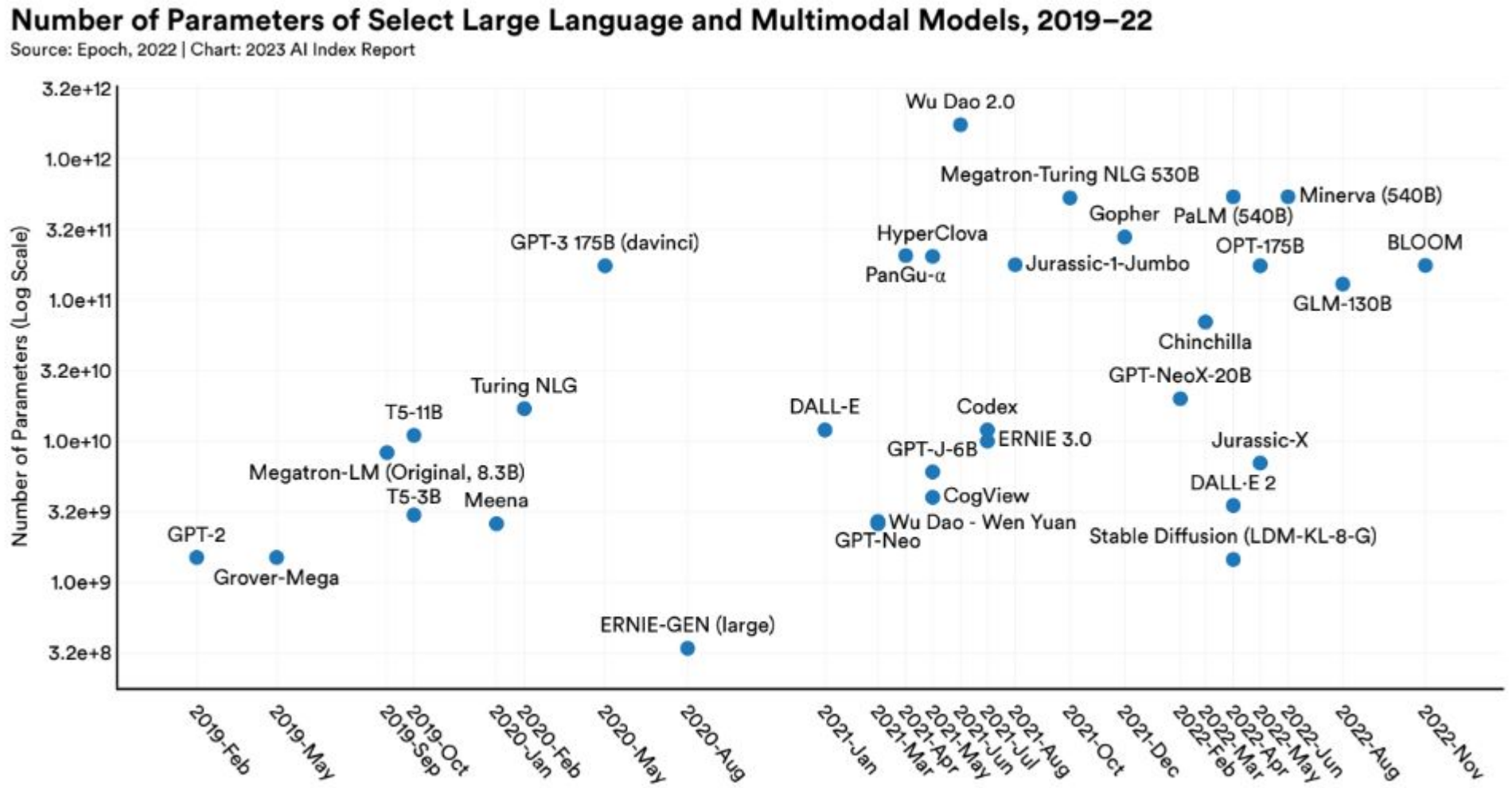
# Neural Networks

enfabrica



# Growth of Model Sizes

enfabrica



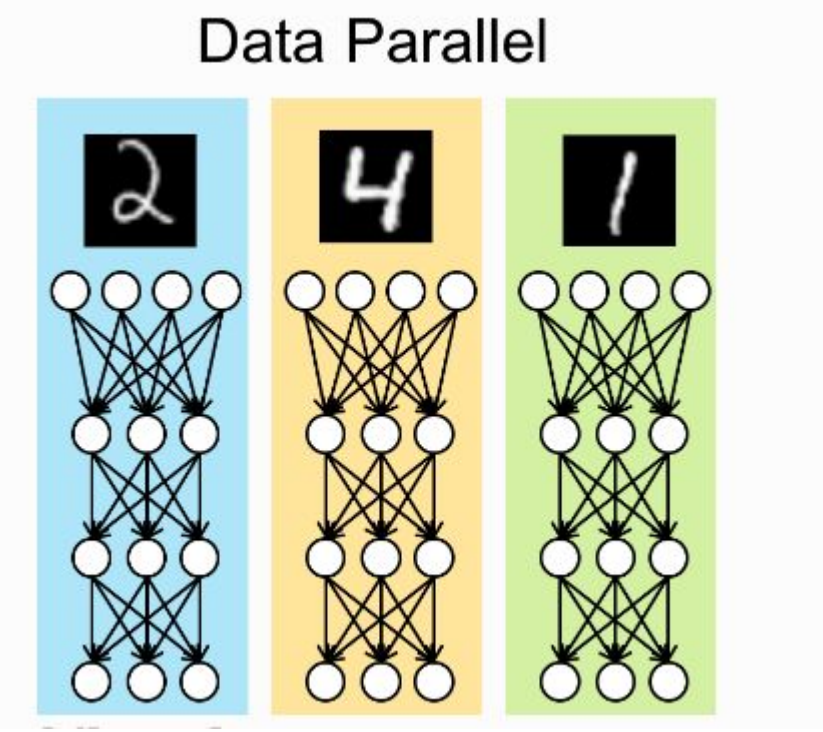
# Taxonomy of Parallelization Techniques

enfabrica

- Data Parallelism
- Model Parallelism
- Fully Sharded Data Parallelism
- Tensor Parallelism
- Pipeline Parallelism
- Mixture-of-Experts (MoE)
- Framework-specific techniques

# Data Parallelism

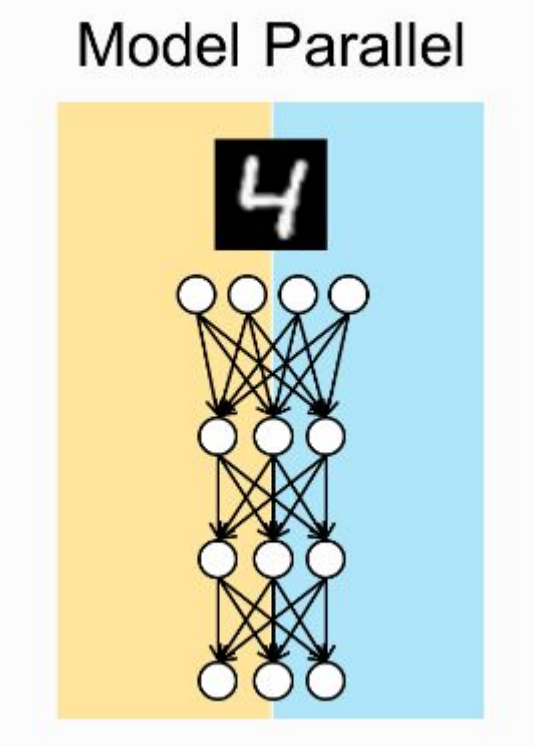
enfabrica




- Data Parallelism
  - Same model across GPUs/compute elements
  - Train each with different samples
  - Stochastic Gradient Descent (SGD)
    - Asynchronous
      - Non-blocking across GPUs
      - Parameter Server model
    - Synchronous
      - AllReduce (Sum) of gradients from all GPUs before updating model
      - SGD with large mini-batch is the most common training model
- ~1.2B parameters

# Model Parallelism

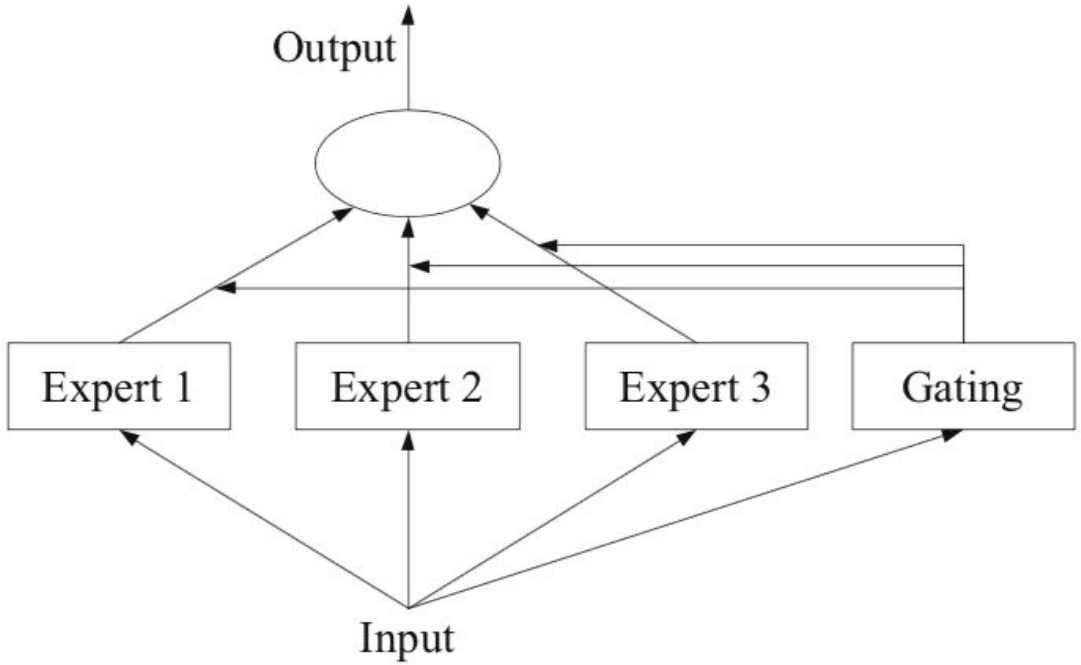
enfabrica




- Model Parallelism
  - Distribute model across compute elements
  - Given batch -> sequential progress across layers. Same input batch fed to all GPUs.
  - MPI-style tightly-couple communication requirements
- Scaling limitations, 8-16 GPUs. ~20B parameters
- Hybrid MP+DP
- Pipeline parallelism. ~100B parameters
- Framework-specific techniques - ~1T+ parameters

# Mixture of Experts

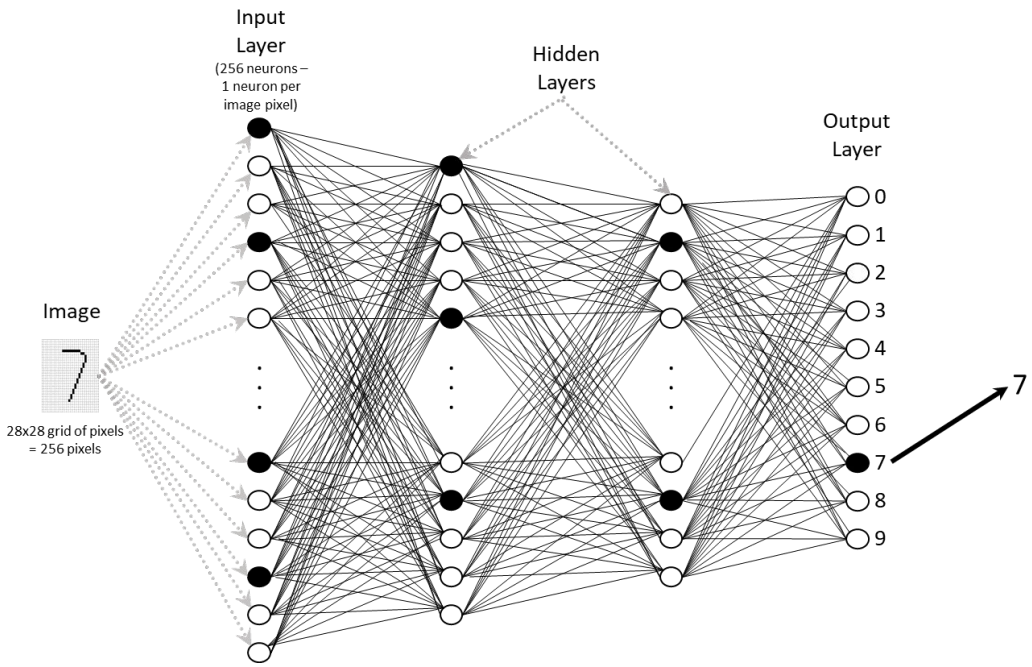
enfabrica




- Limited to encoder-decoder / sequence-to-sequence tasks
- Reduced compute requirements
- Increased memory requirements
- Reduced parameter efficiency
- Google Gshard, Switch Transformer, DeepSpeed-MoE (MoS)
- Sub-linear scaling

Zhi-Hua Zhou. 2012. Ensemble Methods: Foundations and Algorithms (1st. ed.). Chapman & Hall/CRC.

# Inference




- “Deploy” a model with learned weights and biases
- Strict SLAs: latency O(ms) and accuracy
- Deployment scale grows with user counts. High memory requirements.
- LLM / DLRM: Parameters, embeddings, user contexts.

# Role of Collective Communication

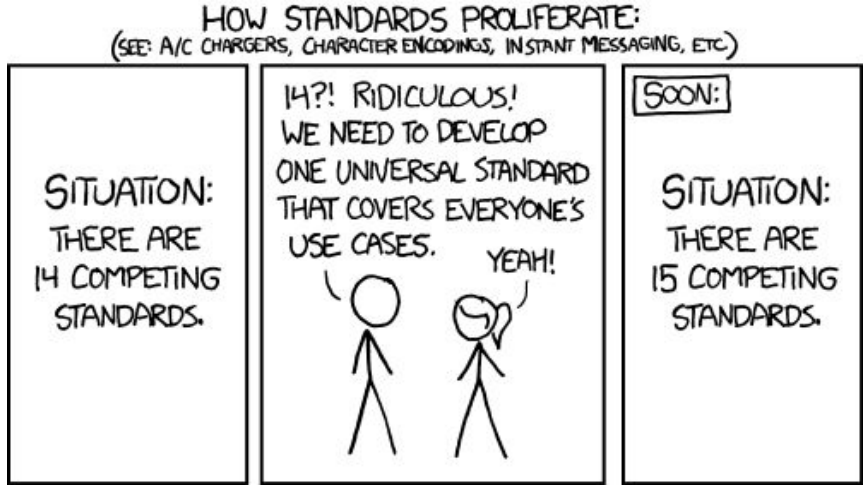
enfabrica

- Initial parameter distribution and batch setup: Broadcast
- Gradient descent: AllReduce w/average in the backward pass
- Synchronization: Barrier
- Fully Sharded Data Parallel: ReduceScatter + AllGather
- Mixture-of-Experts: 2 x AlltoAll forward pass, 2 x AlltoAll backward pass

# The Proliferation of *CCLs

enfabrica

- MPI: The OG CCL
- NCCL: NVIDIA GPUs
- RCCL: For AMD GPUs, via ROCm Hip instead of CUDA
- oneCCL: For Intel Xe GPUs, via oneAPI (libfabric and friends)
- UCC: Collectives over UCX, draws from HCOLL, SHArP, IBM PAMI, etc
- MSCCL = TACCL + SCCL: Topology aware, GC3 DSL
- Alibaba ACCL
- MCR-DL: Mixing network backends
- Open XLA / HLO collectives
- Baidu's allreduce
- Huawei UCG w/OpenMPI
- Xilinx ACCL: For Alveo FPGAs



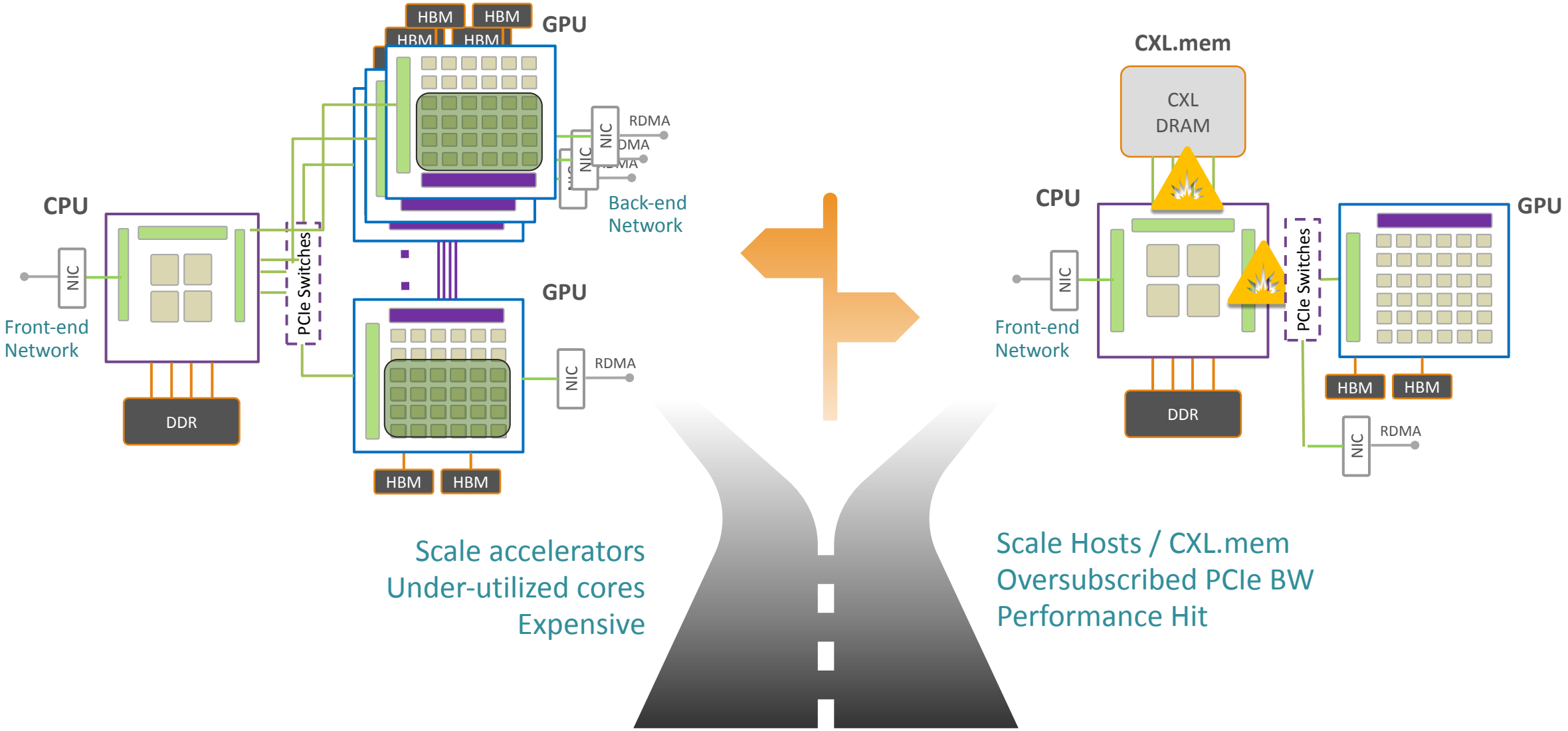
# Networking for ML: Design Challenges

enfabrica

- Plenty of FLOPS, oversubscribed and underutilized
- 800G ethernet is here. Networking bandwidth has caught up with memory bandwidth
- Link utilization and load balancing: Collectives → Incast issues → Congestion control
- But we also want faster collectives!
- Mosaic of L3/L4 transport design choices: message semantics, ordering constraints, etc
- Model sizes outgrowing accelerator HBM capacities
- Desire to build vendor-agnostic software stacks
- Managing complex system topologies across networking, compute, and storage.

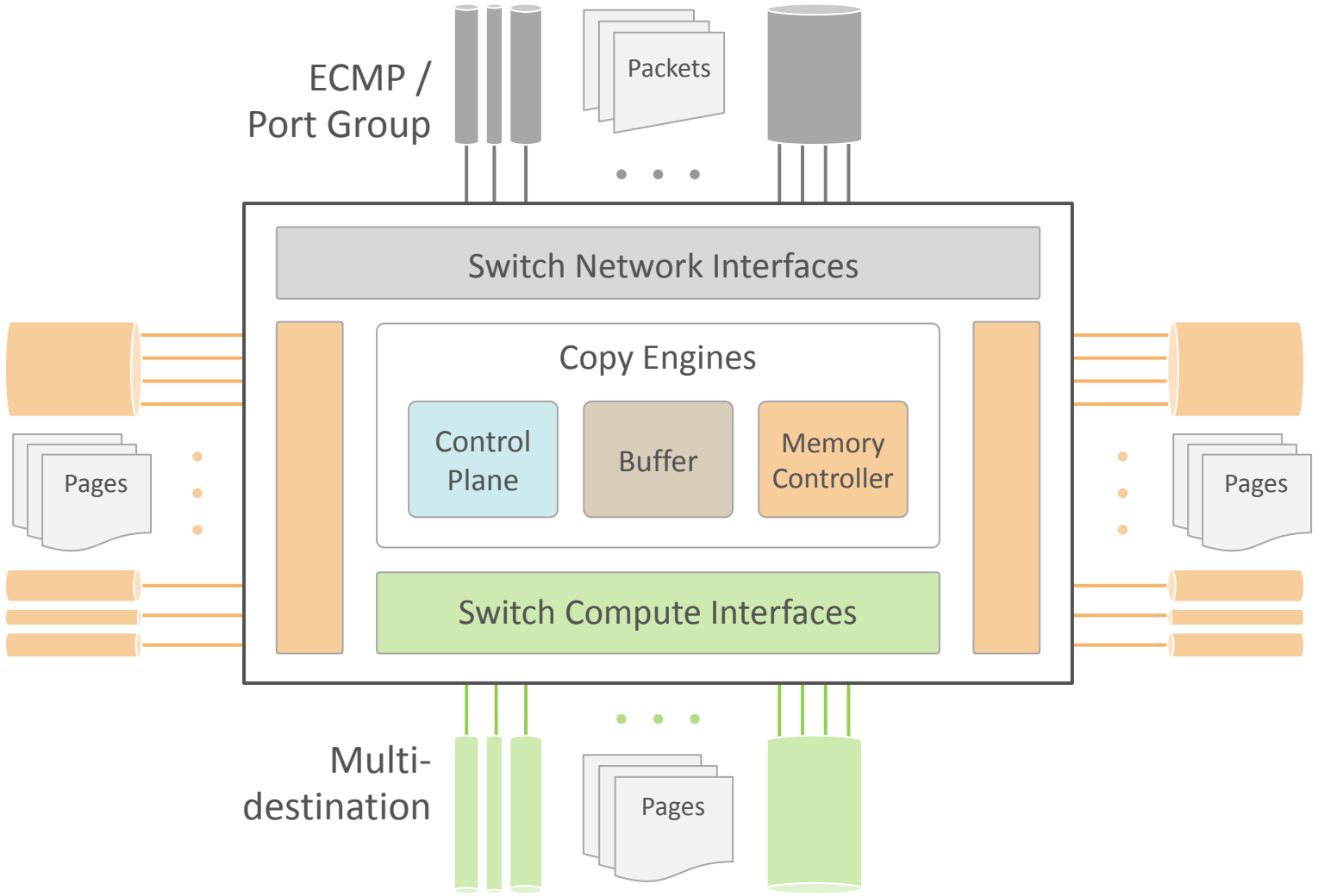
# System Architectures

enfabrica



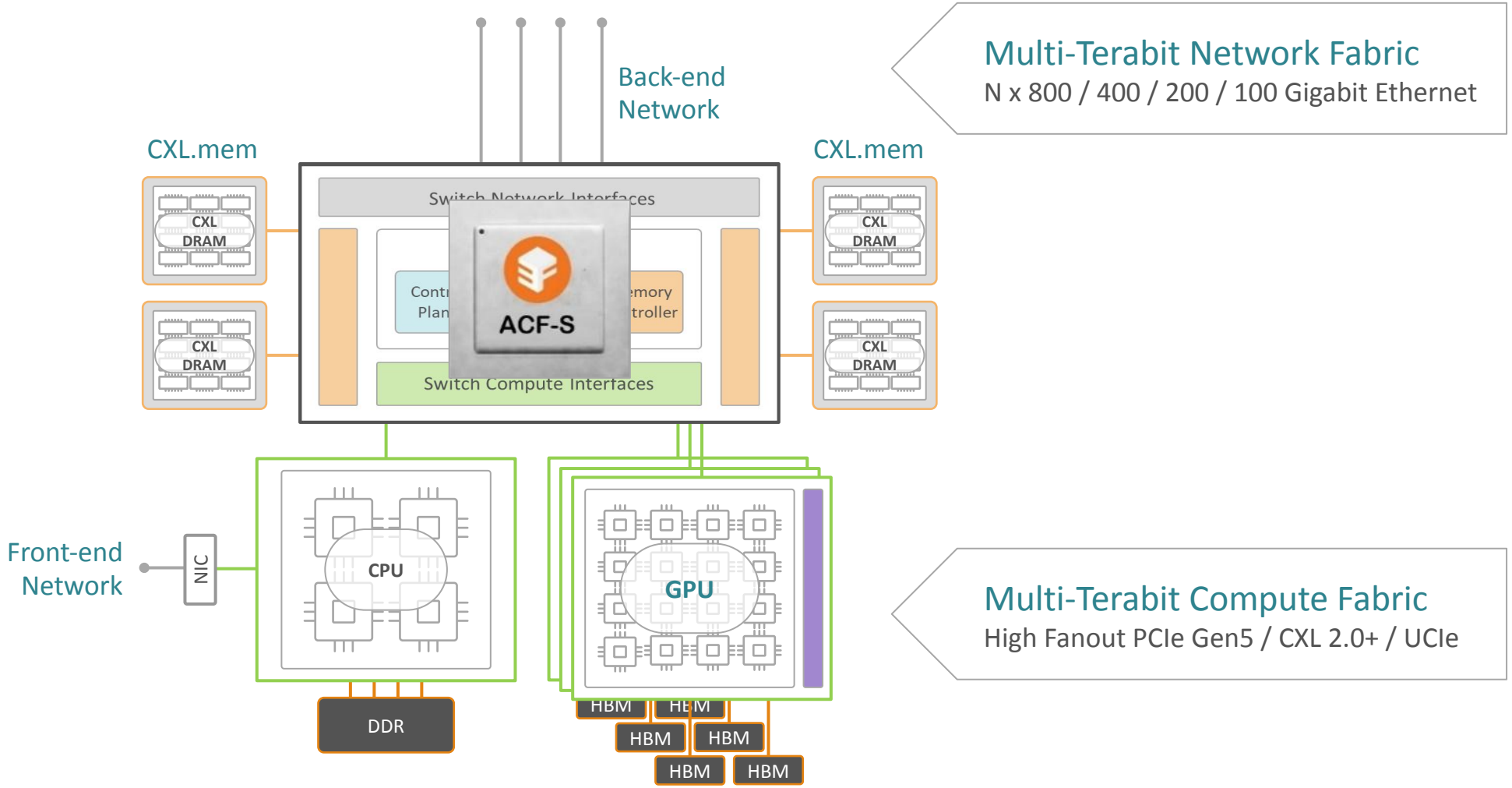
# Enfabrica - ACF-Switch

enfabrica



# An Incarnation of ACF-S

enfabrica

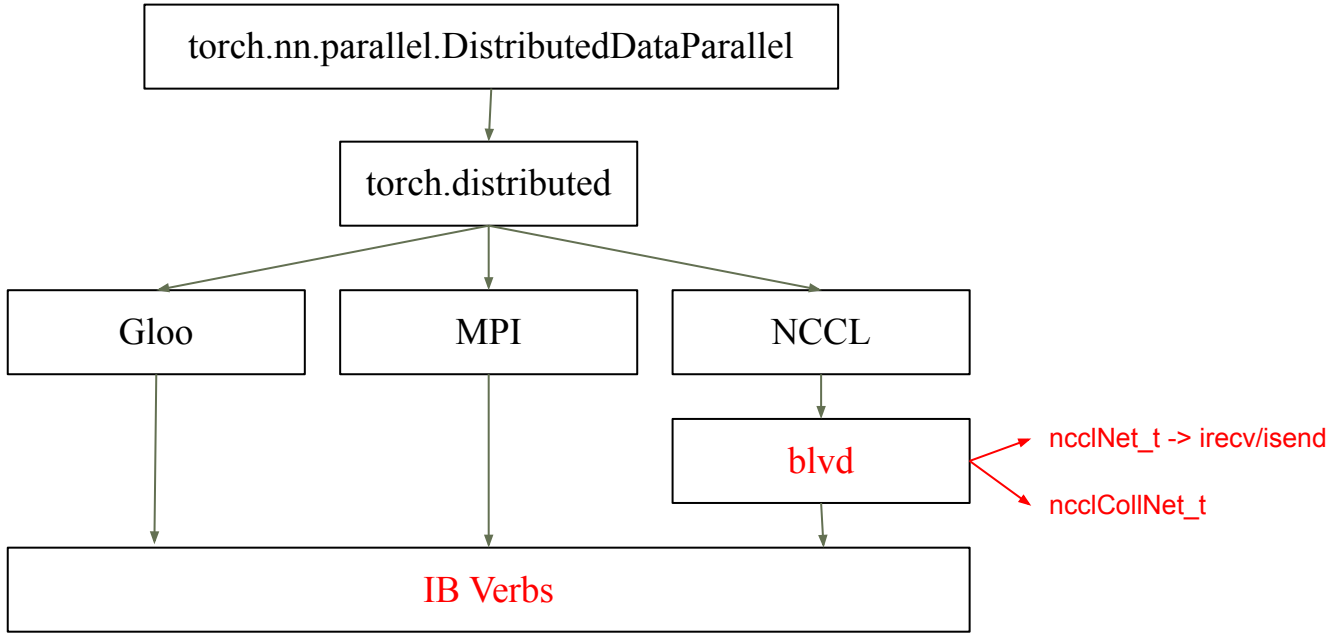


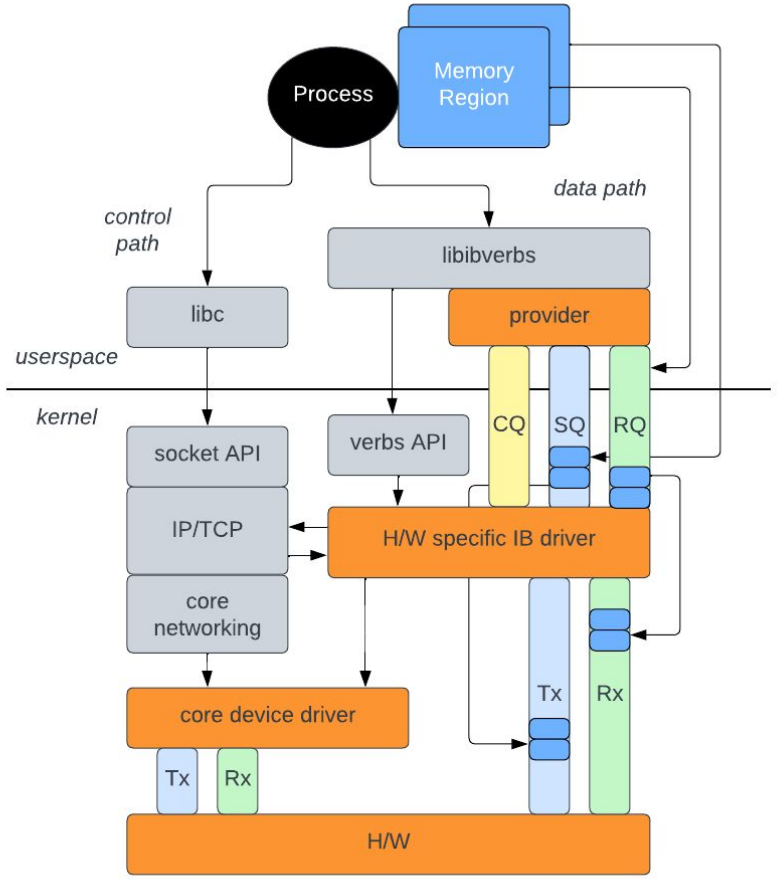

**Multi-Terabit Network Fabric**
N x 800 / 400 / 200 / 100 Gigabit Ethernet

**Multi-Terabit Compute Fabric**
High Fanout PCIe Gen5 / CXL 2.0+ / UCIe

# blvd: NCCL Plugin and more

enfabrica

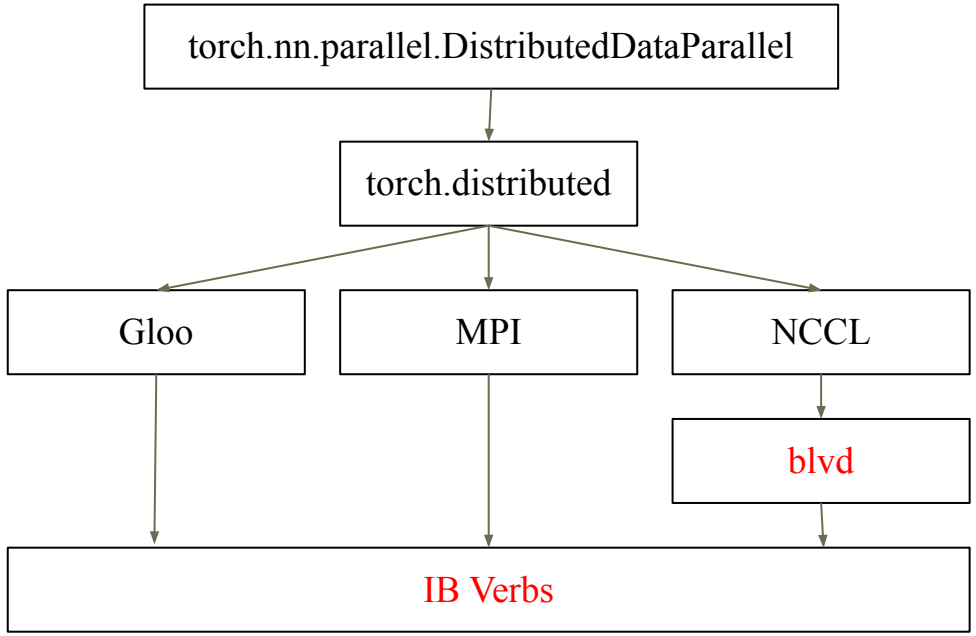



(See Shrijeet Mukherjee's talk from earlier today for details: "**RDMA and Linux TCP**")

# blvd: Design Highlights

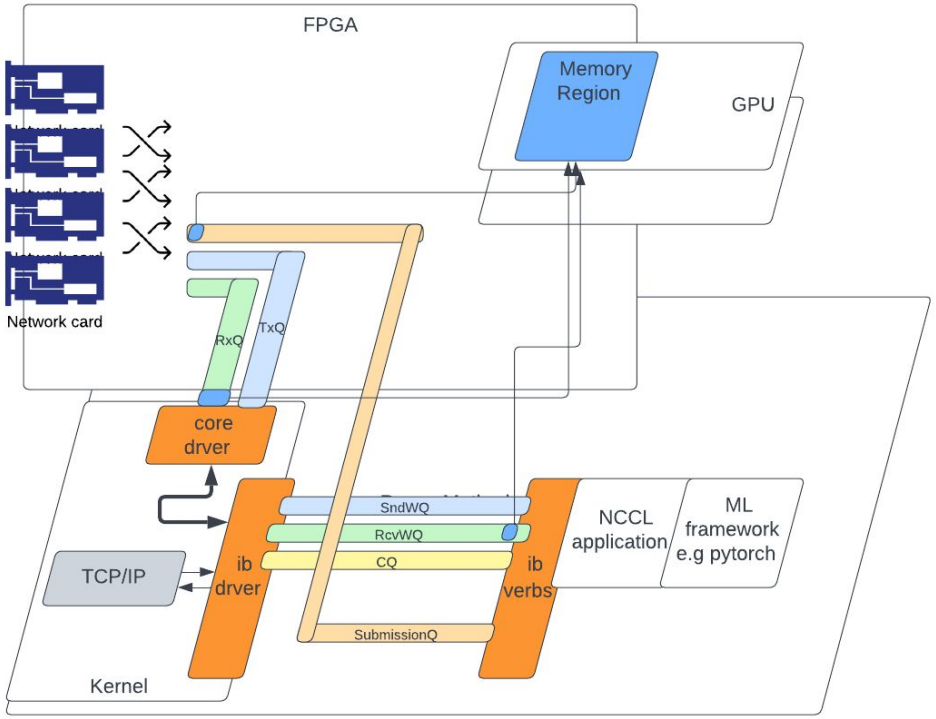
enfabrica




- Zero-copy from the app → NCCL → blvd → TCP → H/W
- Works with upstream kernel without needing custom hooks for peer-to-peer communication
- librdmacm API for connection management. Standard libibverbs APIs.
- Supports all NCCL PTR types: HOST, CUDA, DMABUF
- Message framing extensions
- Test vehicle to prove out ACF architectural concepts and primitives
- Existing verbs plugin is not general enough. Non-standard assumptions and custom hooks for P2P communication.

# FPGA-based Emulation Platform

enfabrica

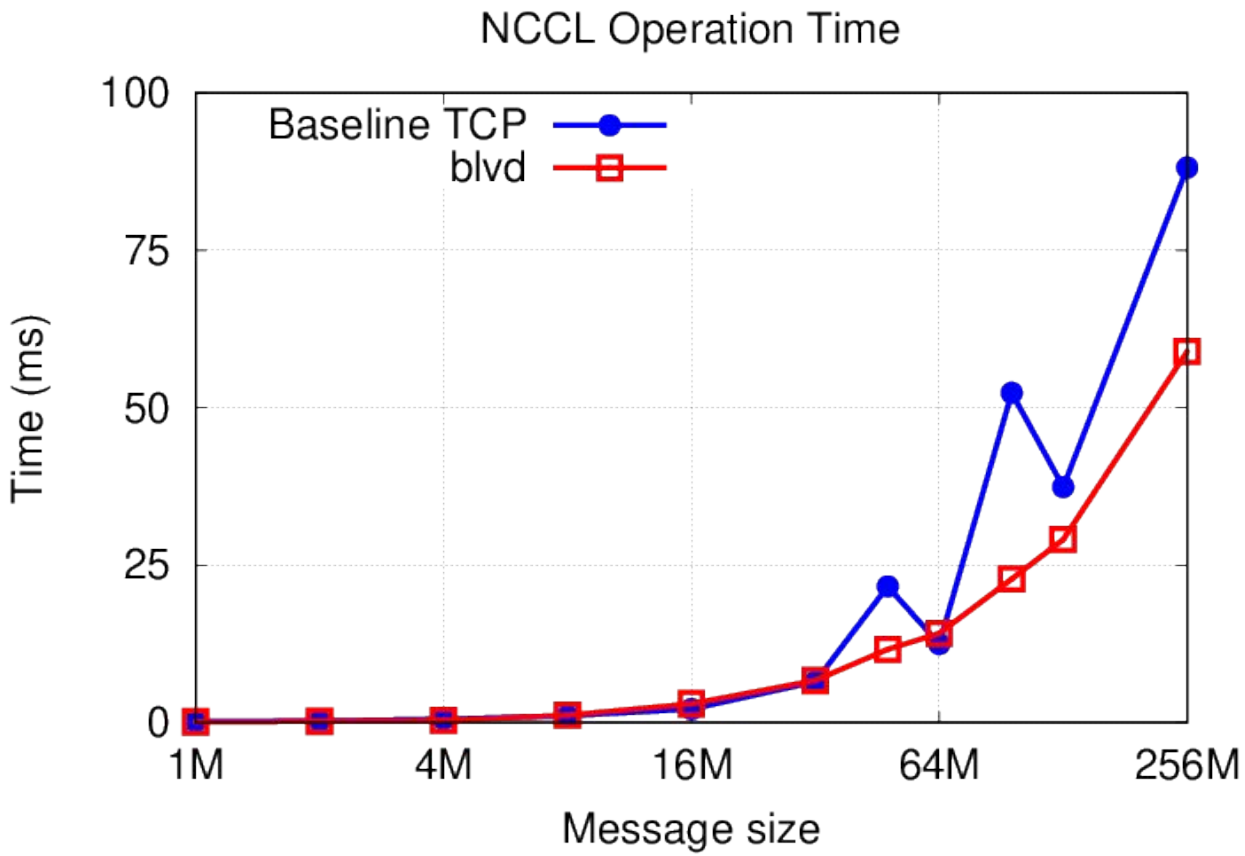




- AMD Ryzen motherboard
- Virtex UltraScale+  XCVU37P
- Xilinx PCIe Gen3 x16 lane (126Gbits/s) interface to the CPU
- 2 x PCIe Gen3 x16 lanes to the GPUs, limited to ~40Gbps
- 4 x 100G Ethernet CMACs with routing lookup BCAM
- Running unmodified 5.13.x kernel
- Purely an **experimental platform**

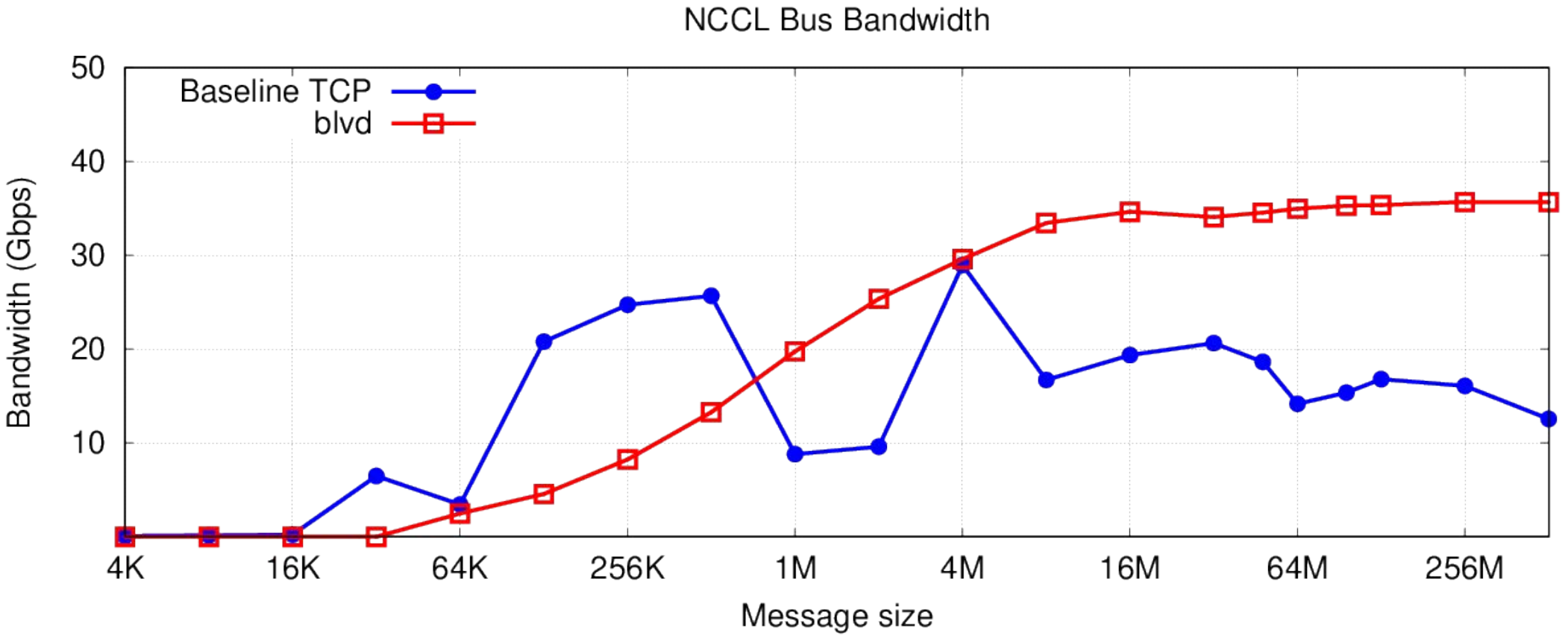
# Evaluation Results

enfabrica



# Evaluation Results

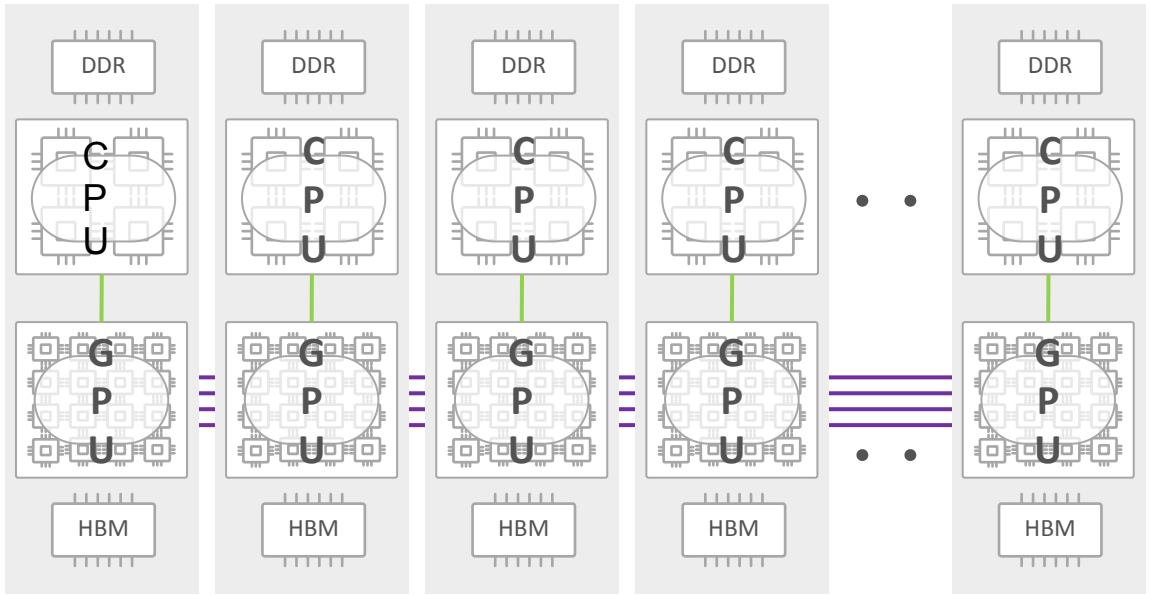
enfabrica



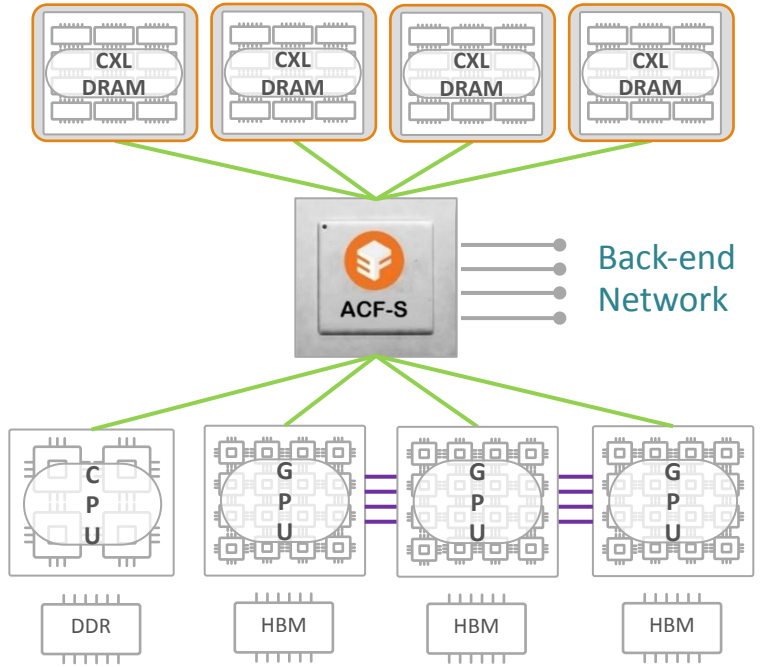

https://github.com/NVIDIA/nccl-tests/blob/master/doc/PERFORMANCE.md

# Next Steps

enfabrica




Scales in Horizontal slices for User Data Store
O(n) Cost Scaling w/ GPU, CPU




Hub-and-Spoke, Dynamic Dispatch of User Data
Enables horizontal slicing and much more …
O(log(n)) Dynamic Cost Scaling w/ GPU, CPU
Tiered, Equidistant Memory Latency

# Call to Action

enfabrica

- For the *CCL builders and maintainers
  - Better plugin interfaces that have an impedance match with hardware vendor efforts
  - Allow more control over the topology management
  - Better documentation of design choices in addition to user/admin guides
- For the OFA community
  - Co-design ML stacks, beyond writing plugins.
  - More active involvement in ML framework development communities
- For the HPC community: Come up with better names for designs! /s

# Thank You.

Questions, comments, or collaboration:

raghu@enfabrica.net + hello@enfabrica.net