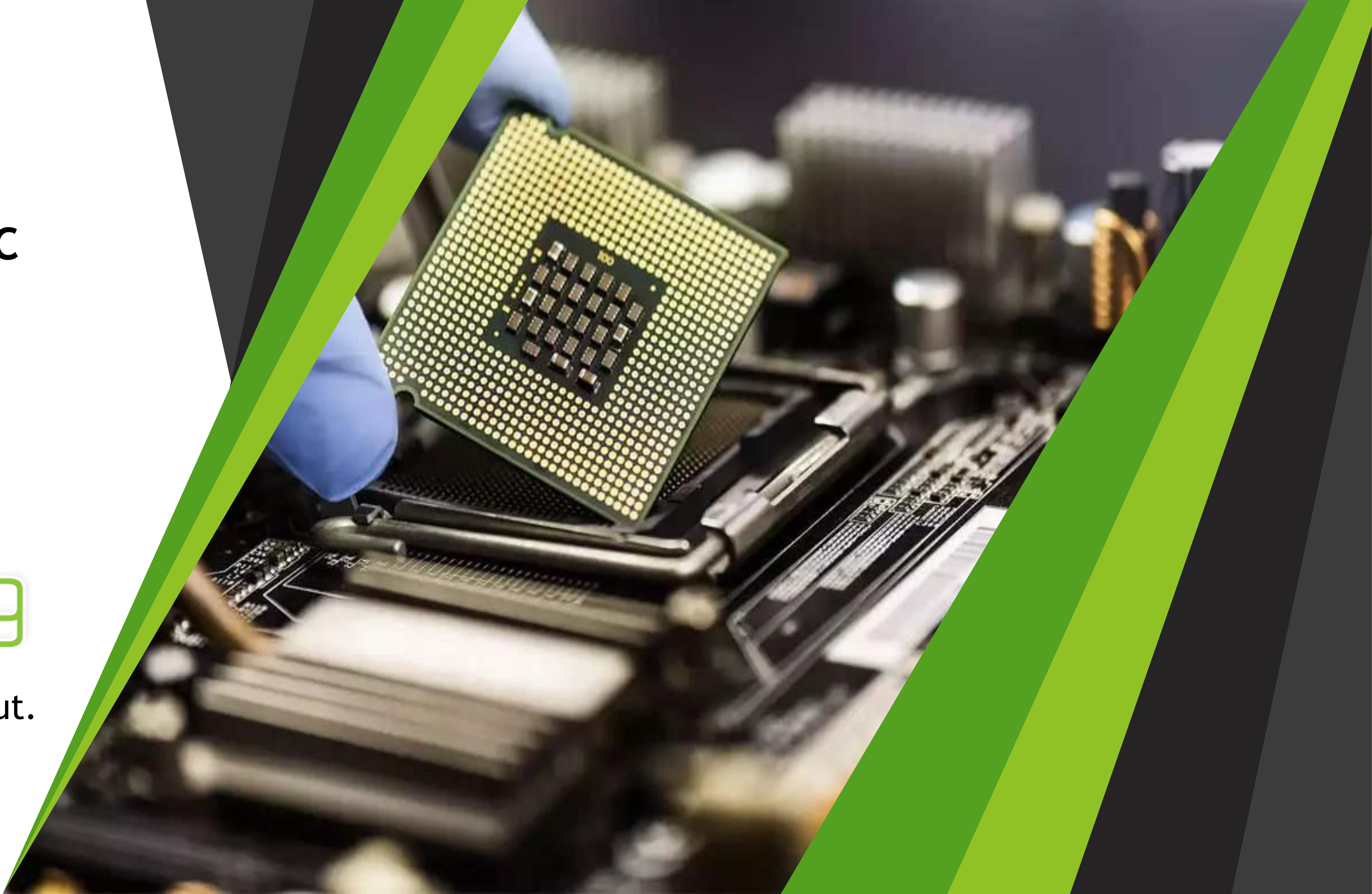
Ethernet at the Core
Enabling Next-Gen AI and HPC Systems

DreamBig SEMICONDUCTOR

Low cost, Low Latency, High Throughput.



Agenda

- AI and HPC workload and interconnects
- NVLink
- RDMA
- UEC
- Scale Out, Scale Up Ethernet
- Putting it all together

Types of Traffic (HPC)

- **Inter-Process Communication (IPC) / MPI Traffic**
  Most HPC applications use **MPI (Message Passing Interface)** for communication between processes running on different nodes.
  - **Characteristics:**
    Latency-sensitive and **bandwidth-intensive**.
    Often consists of **small control messages** (e.g., for synchronization) and **large data transfers** (e.g., matrix rows, vectors).
    Can be **point-to-point** (one process to another) or **collective** (one-to-all, all-to-one, all-to-all).
  - **Traffic Types:**
    MPI_Send / MPI_Recv – peer-to-peer messaging.
    MPI_Bcast, MPI_Reduce, MPI_Allreduce – collective operations generating multicast/broadcast patterns.

Types of Traffic (HPC)

- **Storage I/O Traffic**
  HPC applications often process large datasets, which may reside on parallel file systems like **Lustre, GPFS, or BeeGFS**.
  - **Characteristics:**
    High-throughput, large sequential reads/writes.
    Often **bursty**, especially at checkpointing stages or data staging.
    Can involve **metadata traffic** and **bulk data traffic**.
  - **Traffic Patterns:**
    I/O to shared file systems over InfiniBand or Ethernet.
    I/O node-to-storage node communications.

Types of Traffic (HPC)

- **AI/ML Workloads (in modern HPC)**
  If the HPC cluster is used for machine learning or deep learning:
  - **Characteristics:**
    Heavy data movement between GPUs or CPU-GPU across nodes.
    Use of **NCCL**, **Horovod**, or **gRPC** for communication.
    Can create **high-bandwidth**, **low-latency** traffic over NVLink, InfiniBand, or RoCE.

Common HPC Network Technologies

- **InfiniBand:**
  Low-latency, high-throughput – most common in large HPC clusters.
- **Omni-Path:**
  Intel’s HPC fabric (somewhat legacy).
- **RoCE (RDMA over Converged Ethernet):**
  HPC over Ethernet networks.
- **Ethernet (1G/10G/25G/100G):**
  Often used for storage or management traffic.

Types of Interconnects (HPC)

- **InfiniBand (by NVIDIA/Mellanox)**
  - Extremely **low latency** (as low as 1 µs)
  - **High bandwidth** (up to 400 Gbps with HDR/NDR)
  - Supports **RDMA (Remote Direct Memory Access)**
  - Hardware offloads for MPI
- **Ethernet (Standard or Enhanced)**
  - Widely available and cost-effective
  - **RoCE (RDMA over Converged Ethernet)** adds RDMA support
  - Increasingly used in modern clusters

Types of Interconnects (HPC)

- **NVIDIA NVLink / NVSwitch**

  Used primarily for intra-node GPU-to-GPU communication or between GPU-rich nodes in AI/HPC hybrid systems.

  - Ultra-high bandwidth (600+ GB/s per GPU in NVLink 4.0)
  - Very low latency
  - Integrated with NVIDIA GPUs (A100, H100, etc.)

  - Use Cases:
    - Deep learning
    - GPU-based scientific computing
    - DGX systems and SuperPods

Types of Interconnects (HPC)

Comparison Summary

Interconnect	Max Bandwidth	Latency	RDMA	Topologies	Common In
InfiniBand	400 Gbps (NDR)	~0.5-1 μs	✅	Fat Tree, Dragonfly, Torus	Most large HPCs
Ethernet	400 Gbps	~5-20 μs	❌ / ✅ (RoCE)	Tree, Leaf-Spine	Budget/mixed clusters
Omni-Path	100 Gbps	~1 μs	✅	Fat Tree	Some Intel systems
NVLink	600+ GB/s	~0.3 μs	✅	Mesh, Fully connected	GPU clusters
Slingshot	200–400 Gbps	~1-2 μs	✅	Dragonfly	Exascale systems

Types of Network Traffic (AI)

- **Data Ingestion Traffic**
  Input data (images, text, audio, etc.) is loaded from **storage** to **compute nodes** (often GPUs or TPUs).
  - **Characteristics:**
    High-throughput, especially during training.
    Often **burst-heavy** and **disk-bound**.
    Depends on whether the data is local or remote (e.g., on a NAS, NFS, or object store like S3).
  - **Examples:**
    Training image datasets from a shared file system.
    Streaming video frames for real-time inference.

Types of Network Traffic (AI)

- **Model Synchronization Traffic (Distributed Training)**
  Occurs in **data-parallel** and **model-parallel** training.
  Gradients, weights, or activations are **exchanged between nodes** (GPUs, TPUs, or CPUs).
  - **Characteristics:**
    Extremely bandwidth-intensive
    Often requires **low-latency** for tight synchronization steps.
    Uses **collective communication patterns** (e.g., all-reduce, broadcast).
  - **Technologies:**
    NCCL (NVIDIA Collective Communication Library)
    Horovod, **DeepSpeed**, **PyTorch DDP**
    gRPC, **MPI**, **AllReduce**
  - **Communication Patterns:**
    AllReduce – Synchronize gradients across all nodes.
    Broadcast – Send model parameters from rank 0 to all others.
    Point-to-Point – Common in model-parallel training.

Types of Network Traffic (AI)

- **Inference Serving Traffic**
  Model is deployed to serve real-time or batch predictions.
  - **Characteristics:**
    Often involves **request-response traffic.**
    In latency-sensitive applications (e.g., recommendation systems), performance is critical.
    Batch inference may involve large input data and models.
  - **Communication Flows:**
    Client → Inference server – Input data
    Server → Client – Predicted results
    May use **load balancers**, **model routers**, or **inference gateways**
  - **Common Protocols:**
    REST, gRPC, HTTP/2
    TensorFlow Serving, **Triton Inference Server**

Types of Interconnects (AI)

Network Technologies for AI Traffic

Technology	Used For	Key Traits
InfiniBand	GPU-GPU comms, training sync	Low latency, high bandwidth, RDMA
NVLink / NVSwitch	GPU-to-GPU comms (intra-node)	600+ GB/s, ultra low latency
RoCE (RDMA over Ethernet)	Training over Ethernet	RDMA benefits on Ethernet fabric
Ethernet (10G–400G)	Inference, data loading	General purpose, varies by use case
gRPC / HTTP	Inference APIs, service mesh	Widely used in model serving

Common Protocols and Interconnects (NVLink)

- **NVIDIA NVLink** is a **high-speed, point-to-point interconnect** developed by NVIDIA
- Allows **direct communication between GPUs**, and
- Between **GPUs and CPUs** (in some systems).
- It overcomes the limitations of traditional PCIe by offering:
  - **Higher bandwidth**
  - **Lower latency**
  - **Faster scalability** for multi-GPU systems

Common Protocols and Interconnects (NVLink)

- **NVLink Use Cases in AI & HPC**
  - **Large Model Training (LLMs, Transformers)**
    - Enables **model parallelism** and **pipeline parallelism**
    - GPUs share weights and activations directly
  - **Data Parallel Training**
    - High-speed **gradient synchronization** across GPUs
  - **Unified GPU Memory**
    - Large data sets can be distributed across multiple GPUs and accessed transparently
  - **Simulation and Scientific Computing**
    - HPC workloads (e.g., CFD, molecular dynamics) needing high inter-GPU communication

Common Protocols and Interconnects (NVLink)

NVLink Versions & Specs

Version	Introduced With	Per-Link BW (bi-dir)	Total BW per GPU	Notes
NVLink 1	Tesla P100 (Pascal)	40 GB/s	80 GB/s	Up to 4 links
NVLink 2	Tesla V100 (Volta)	50 GB/s	300 GB/s	Up to 6 links
NVLink 3	A100 (Ampere)	50 GB/s	600 GB/s	12 links per GPU
NVLink 4	H100 (Hopper)	100 GB/s	900 GB/s	18 links per GPU

Common Protocols and Interconnects (NVLink)

- **Limitations / Considerations**
  - **Limited to NVIDIA GPUs** – Not a general-purpose interconnect
  - **Requires system support** – Only available in specific motherboards and chassis (e.g., DGX)
  - **Topology awareness** – Software must account for NVLink topology for optimal performance
  - **Not exposed to all software layers** – Some frameworks may not fully utilize NVLink without explicit tuning

- Need to use NVSwitch for connecting 8+ GPUs
- Latest NVSwitch 3.0 supports maximum 18GPUs
  - **Limitations / Considerations**
  - **Only available in NVIDIA-designed systems** (DGX, HGX)
  - **Expensive and power-hungry** (suited for large-scale enterprise/academic setups)
  - Not available in consumer hardware or standard desktop builds
  - Requires **deep integration** between hardware, firmware, and software (NCCL, CUDA-aware libraries)

Common Protocols and Interconnects (NVLink Spine)

NVLink Spine: A Supercharged Data Superhighway
At Computex 2025, **NVIDIA CEO Jensen Huang** introduced the **NVLink Spine**, part of their NVLink Fusion architecture—an interconnect fabric so powerful that he claimed it can **move more traffic than the entire Internet**

NVLink Fusion is **NVIDIA's next-generation interconnect architecture**, designed to **unify and scale GPU communication** across **entire racks of servers**, enabling them to operate **like a single massive GPU**.

Attribute	Description
System	NVLink Spine (NVLink Fusion)
Total Bandwidth	~130 TB/s
Comparison to Internet	Surpasses estimated global Internet peak (~112 TB/s)
Hardware Composition	5,000 coaxial cables, NVLink switches, 72 GPUs, 1 rack
Applications	Rack-scale AI training; trillion-parameter models

Common Protocols and Interconnects (NVLink 5)

- GPU-level interconnect:
  18 × 100 GB/s links per GPU → 1.8 TB/s bidirectional total

- Switch-level capacity:
  Supports up to **576 GPUs** in a non-blocking fabric
  14.4 TB/s switch bandwidth and **1 PB/s** aggregated per rack
  Enables massive-scale GPU clustering with built-in SHARP offloads for efficiency

Common Protocols and Interconnects (RDMA/RoCEv2)

- RDMA = Remote Direct Memory Access
- RDMA is the technology which allows a network host to access main memory of another host without involving the CPU.
- It improves data throughput and performance and frees up CPU and resources.
  - ✓ This results in higher data transfer rates and lower latencies.
  - ✓ It supports zero-copy operation by allowing the NIC to copy data directly from the wire to the application memory or from application memory to the wire – no data copy between application memory and kernel buffers.

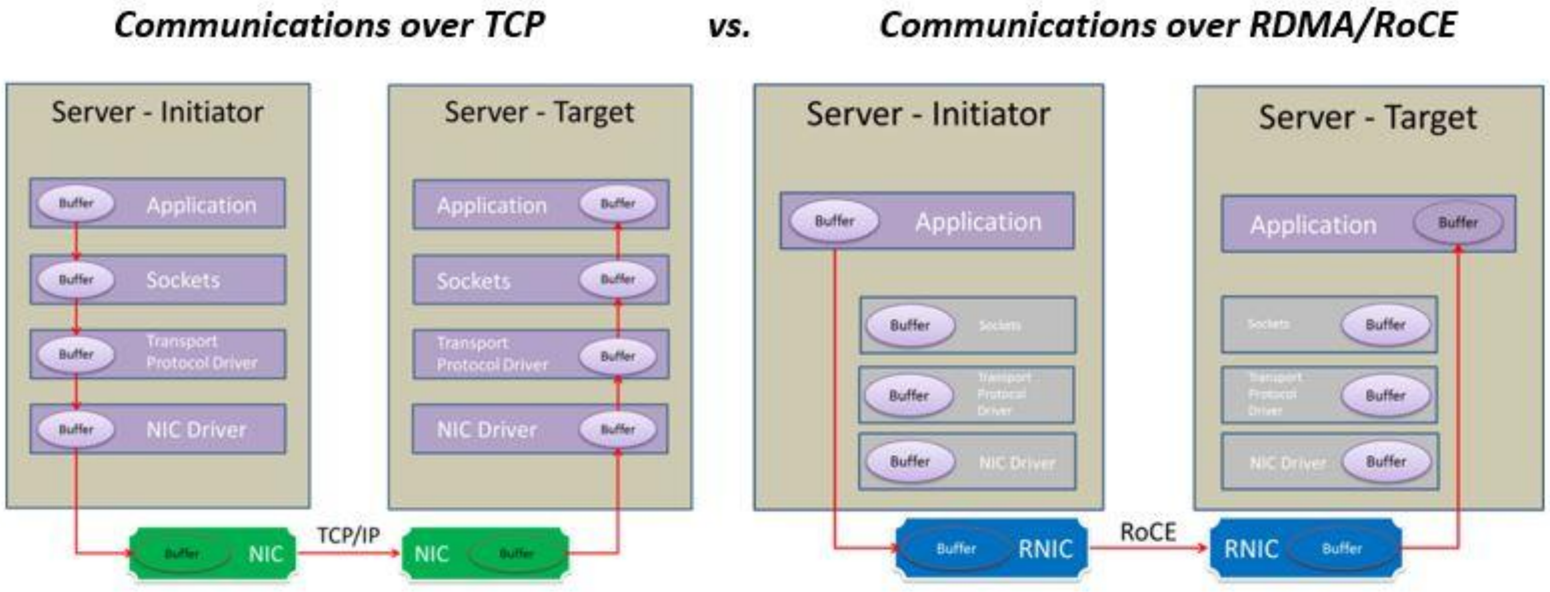


Diagram courtesy of https://developer.nvidia.com/blog/doubling-network-file-system-performance-with-rdma-enabled-networking/

Where The RDMA Protocol Works - AI (Meta)

- RDMA over Ethernet for Distributed AI Training at Meta Scale
  - https://dl.acm.org/doi/pdf/10.1145/3651890.3672233
- RoCE clusters supporting 32,000 GPUs
- Backend network is a specialized fabric that connects all RDMA NICs in a non-blocking architecture, providing high bandwidth, low latency, and lossless transport between any two GPUs in the cluster, regardless of their physical location.
- This backend fabric utilizes the RoCEv2 protocol, which encapsulates the RDMA service in UDP packets for transport over the network.

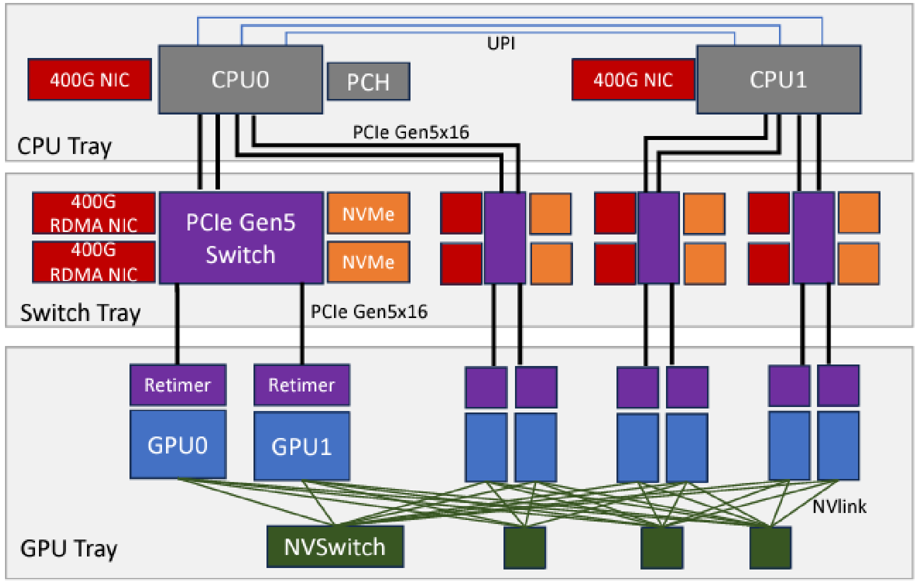


Figure 4: Grand Teton platform

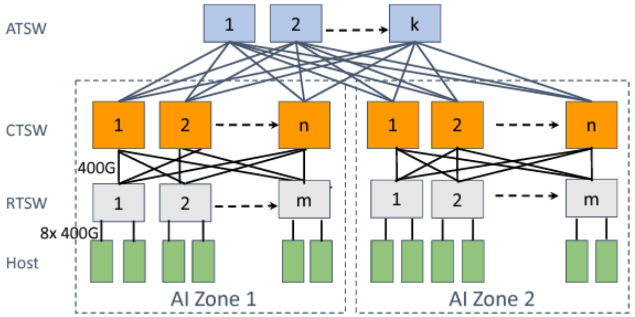


Figure 6: Backend Network Topology

Where The RDMA Protocol Works - AI (Meta)

- RDMA over Ethernet for Distributed AI Training at Meta Scale
    - https://dl.acm.org/doi/pdf/10.1145/3651890.3672233
- For larger jobs, RDMA NICs enable GPUDirect technology, so that GPU-to-GPU traffic can bypass host and host memory bottlenecks.
- GPUDirect RDMA provides direct communication between NVIDIA GPUs in remote systems. This eliminates the system CPUs and the required buffer copies of data via the system memory, resulting in 10X better performance.

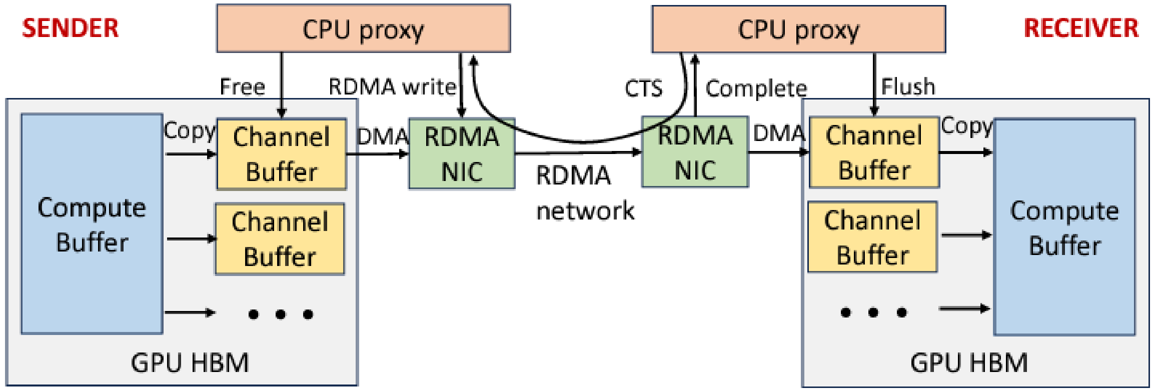


Figure 14: GPU to GPU communication architecture.

Where The RDMA Protocol Works - HPC

- Swift-X: Accelerating OpenStack Swift with RDMA for Building an Efficient HPC Cloud
    - https://shashankgugnani.github.io/publications/ccgrid_17.pdf
- The OpenStack Object Store project, known as **Swift**, offers cloud storage software so that you can store and retrieve lots of data with a simple API
- Introduced an RDMA-based communication module in the client, object server and proxy server for low latency communication

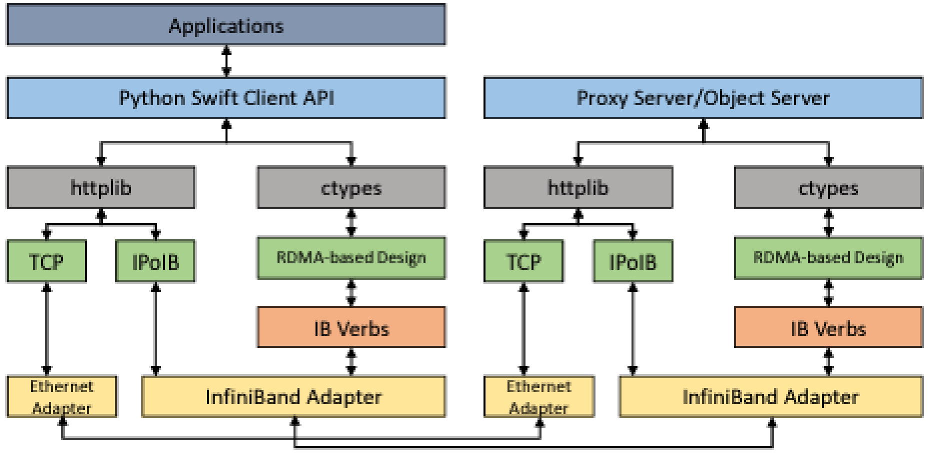


Figure 4. Technology Overview

Where The RDMA Protocol Works - AI (AliBaba)

- Alibaba HPN: A Data Center Network for Large Language Model Training
  - https://ennanzhai.github.io/pub/sigcomm24-hpn.pdf
  - Primary capacity goal of containing 15K GPUs
  - The additional capacity goal of our data center, therefore, is to be able to support the scale at 100K GPUs.
  - We equip each host with 9 NICs each with 2×200Gbps
  - Each of these eight NICs serves for a dedicated GPU (named rail), and thus each GPU has a dedicated 400Gbps of RDMA network throughput, resulting in a total bandwidth of 3.2Tbps.

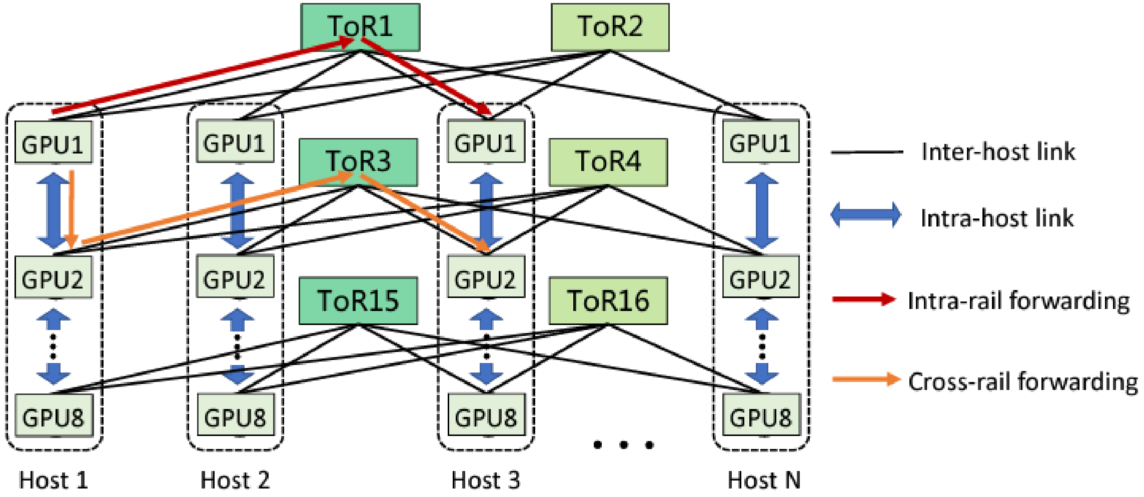


Figure 11: Rail-optimized network under dual-ToR.

Common Protocols and Interconnects Limitations

- **AI Training Clusters Need:**
  - Ultra-low latency
  - High bandwidth (e.g., 400 Gbps–800 Gbps+)
  - Support for large models, dense GPU clusters
  - Efficient collective ops like **AllReduce**, **Broadcast**
- **Problem:**
  - Traditional Ethernet (even 400G/800G) **was not designed for tightly coupled, latency-sensitive compute**
  - Custom fabrics like **InfiniBand** or **NVLink** dominate HPC and AI, but lack **standardization** and **broad interoperability**
- **Industry Answer:**
  - Make **Ethernet itself "Ultra"** by enhancing it to meet these advanced demands, while preserving its **openness and scale**.

Ultra Ethernet Consortium (UEC)

What Is the Ultra Ethernet Consortium (UEC)?

- **Launched in July 2023**
- Operates under the **Linux Foundation**
- Backed by major industry players:
  - **AMD, Intel, Broadcom, Cisco, HPE, Arista, Meta, Microsoft, Dell**, and others
  - NVIDIA joined August 2024
- **Goal:**
  - Redesign Ethernet from the ground up to handle **AI-scale computing**, **high-performance storage**, and **supercomputing-like workloads.**
- **Core Focus**
  - Low latency, RDMA, congestion control, AI-optimized transport
- **Why It Matters**
  - Aims to replace/supplement InfiniBand with open, fast Ethernet

Ultra Ethernet Consortium (UEC)

The Ultra Ethernet Consortium’s role is to define and promote next-generation Ethernet standards that go beyond traditional datacenter use by focusing on:

- **High Performance Networking**
  Compete with or surpass InfiniBand, NVLink, and custom fabrics
  Deliver low-latency, high-bandwidth, and scalable networking for AI clusters
- **AI-Ready Networking Fabric**
  Build an Ethernet-based fabric optimized for large-scale distributed AI training
  Support collective operations like AllReduce, used by frameworks such as PyTorch or TensorFlow
- **RDMA & GPU-Aware Networking**
  Develop standardized RDMA (Remote Direct Memory Access) over Ethernet
  Improve interoperability between NICs, GPUs, and CPUs
- **Transport and Congestion Control**
  Develop custom congestion control, packet pacing, and fair queuing for Ethernet
  Replace or enhance TCP with more performance-aware transport layers
- **End-to-End Fabric Architecture**
  Provide architectural guidelines, APIs, and protocols to optimize the full network stack for performance and scale

Scale Up Ethernet

- **Scale-up Ethernet** refers to **increasing performance or bandwidth **within a single server (or node)** by adding **more powerful components**, such as faster Ethernet NICs, more ports, or higher link speeds.
- **Focus:**
  Maximize performance per node
  Improve **intra-node** or **intra-chassis** networking (e.g., GPUs, CPUs, storage inside a server)

- **How It's Done:**
  Upgrading NICs from 100G to 400G or 800G
  Using **multi-port NICs** (e.g., 2×400G or 4×200G)
  Connecting internal GPUs/CPUs via **high-bandwidth internal Ethernet**
  Leveraging **PCIe Gen5** with fast Ethernet offload

- **Goal:**
  Improve **latency**, **bandwidth**, and **efficiency** of workloads **inside the box**
  Example: Making a DGX-class server fully Ethernet-connected for high-performance GPU comms

Scale Out Ethernet

Scale-out Ethernet refers to expanding across **multiple nodes or systems** using Ethernet to create a **larger distributed cluster or fabric**.

- **Focus:**
  Cluster-level scalability
  Build **distributed AI/HPC infrastructure** by connecting many servers

- **How It's Done:**
  Leaf-spine topologies using high-speed Ethernet switches (400G, 800G)
  Standardized or optimized **RDMA over Ethernet (e.g., RoCE)**
  Smart NICs with offloads (e.g., NVIDIA BlueField, Intel Mount Evans)
  Congestion control, flow isolation, and collective ops optimization

- **Goal:**
  Build large-scale systems (like **AI supercomputers**) over Ethernet instead of InfiniBand or proprietary interconnects

Scale Up vs Scale Out

Key Differences: Scale-Up vs Scale-Out Ethernet

Feature	**Scale-Up Ethernet**	**Scale-Out Ethernet**
Scope	Within a node (server)	Across multiple nodes (cluster)
Goal	Maximize local compute/network efficiency	Expand system horizontally for scalability
Usage	GPU-to-GPU traffic, local storage I/O	Distributed training, distributed storage (e.g., Ceph)
Example	8-GPU server with 800G internal bandwidth	AI cluster with 256 nodes connected over 400G Ethernet
Focus Area	Bandwidth density, latency, NIC offload	Routing, congestion control, collective comms
Challenges	Heat, bus contention, socket NUMA	Network congestion, tail latency, fairness
Technologies	PCIe Gen5, SmartNICs, GPU Direct, NUMA-aware NICs	RoCEv2, SmartNICs, Ultra Ethernet, ECMP, PFC

Putting It All Together

AI, HPC frameworks, NCCL, MPI, and **scale-up, scale-out hardware**

Top layer:

AI frameworks (PyTorch, TensorFlow, JAX and HPC applications.

Middle layer:

Distributed training frameworks (DDP, Horovod, DeepSpeed, Ray).

Communication:

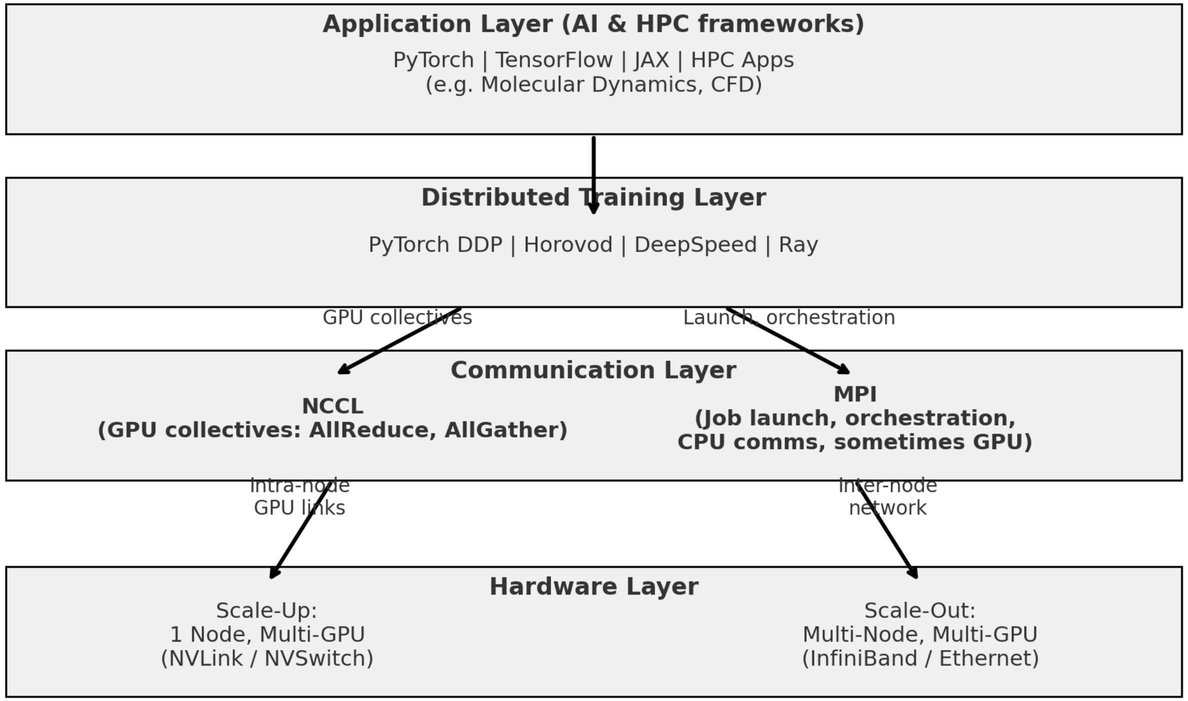
NCCL → GPU-to-GPU collectives (AllReduce, etc.)

MPI → job orchestration, CPU comms, sometimes GPU orchestration.

Hardware:

Scale-Up → multiple GPUs in one node (NVLink/NVSwitch).

Scale-Out → multiple nodes with GPUs over InfiniBand/Ethernet.



Application Layer (AI & HPC frameworks)
PyTorch | TensorFlow | JAX | HPC Apps
(e.g. Molecular Dynamics, CFD)
Distributed Training Layer
PyTorch DDP | Horovod | DeepSpeed | Ray
GPU collectives
Launch, orchestration
Communication Layer
NCCL
(GPU collectives: AllReduce, AllGather)
MPI
(Job launch, orchestration,
CPU comms, sometimes GPU)
Intra-node
GPU links
Inter-node
network
Hardware Layer
Scale-Up:
1 Node, Multi-GPU
(NVLink / NVSwitch)
Scale-Out:
Multi-Node, Multi-GPU
(InfiniBand / Ethernet)

Putting It All Together

All these strategies are **complementary**:
A modern AI cluster needs **scale-up bandwidth** (to connect GPUs inside a server efficiently), and **scale-out fabric** (to connect those servers to each other).

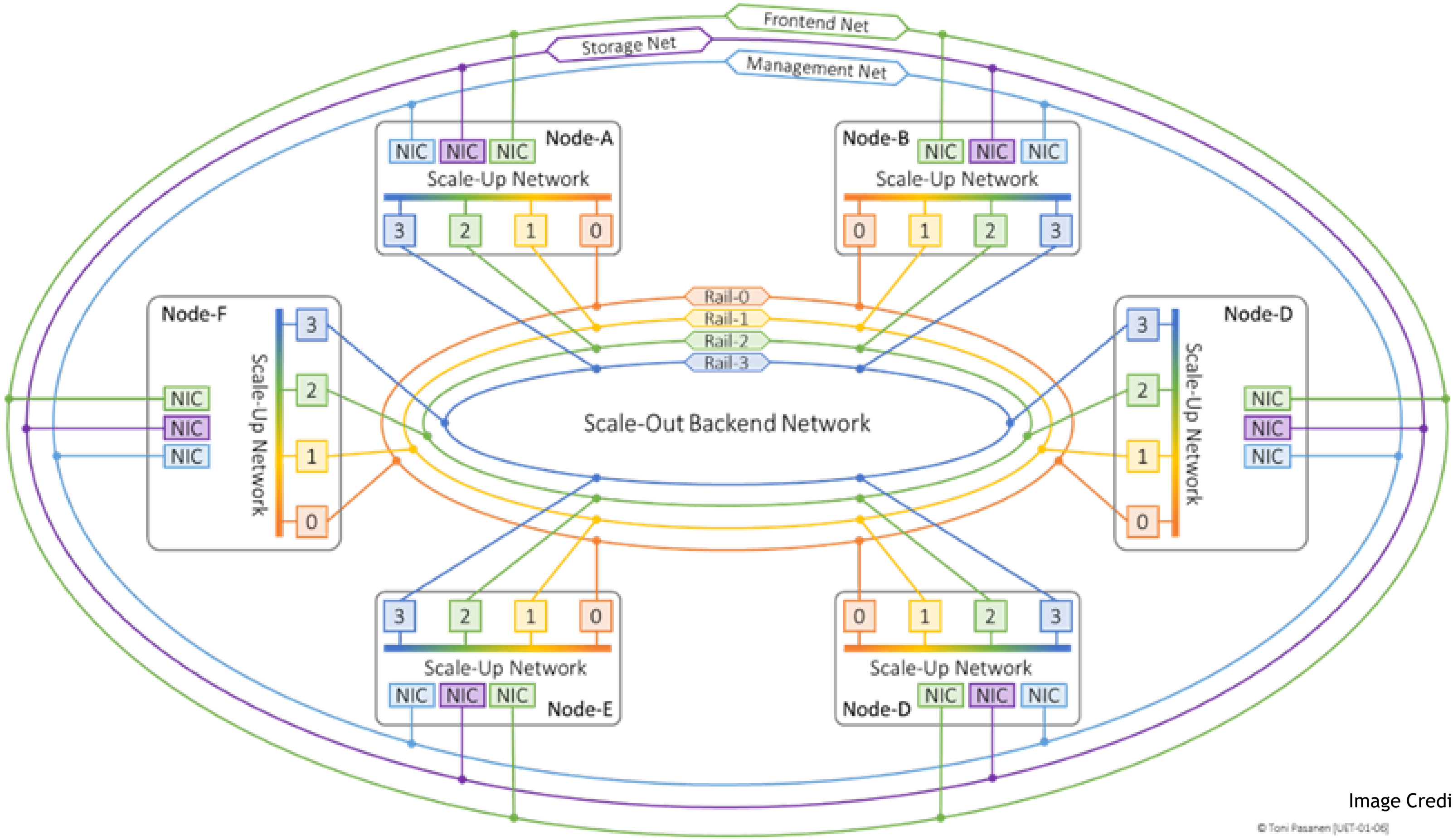
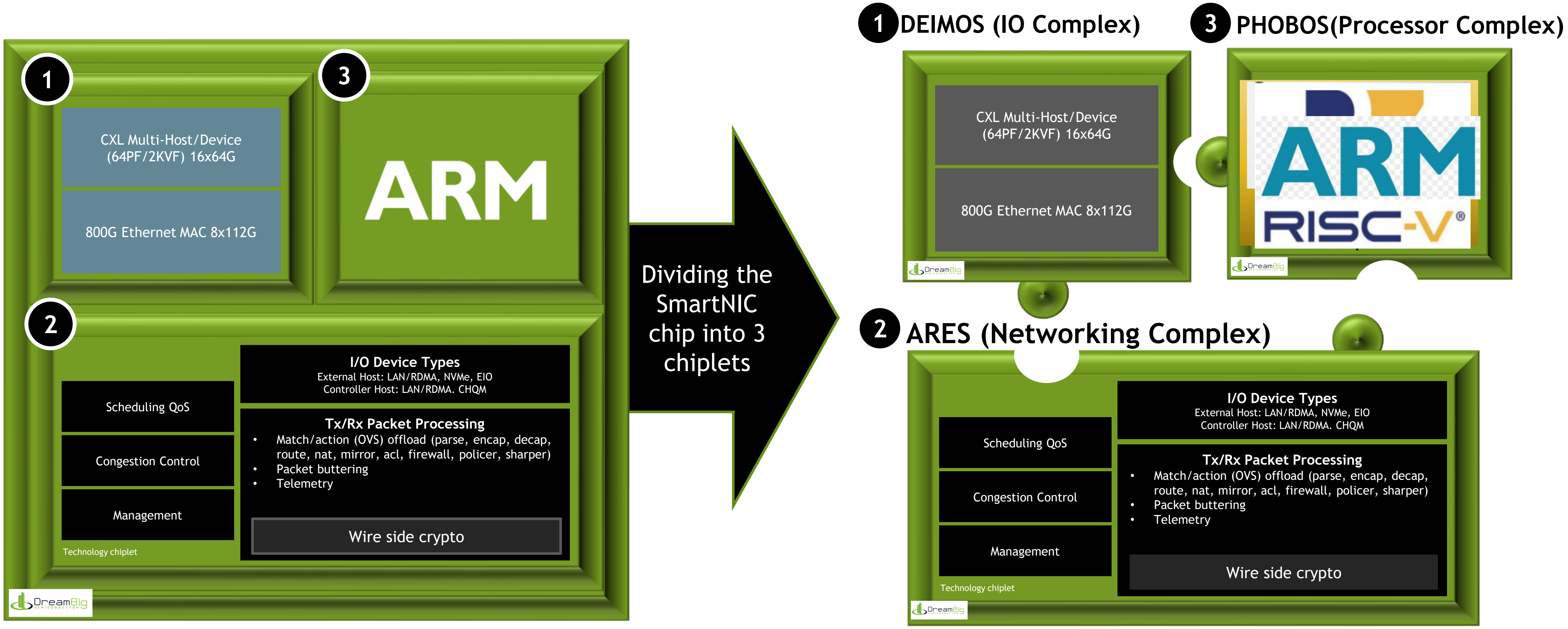


Image Credits: https://nwktimes.blogspot.com/

DreamBig pioneering a game changing Chiplet approach



1

CXL Multi-Host/Device (64PF/2KVF) 16x64G

800G Ethernet MAC 8x112G

3

ARM

2

Scheduling QoS

Congestion Control

Management

I/O Device Types

External Host: LAN/RDMA, NVMe, EIO
Controller Host: LAN/RDMA. CHQM

Tx/Rx Packet Processing

- Match/action (OVS) offload (parse, encap, decap, route, nat, mirror, acl, firewall, policer, sharper)
- Packet buttering
- Telemetry

Wire side crypto

Technology chiplet

Dividing the SmartNIC chip into 3 chiplets

1 DEIMOS (IO Complex)

CXL Multi-Host/Device (64PF/2KVF) 16x64G

800G Ethernet MAC 8x112G

3 PHOBOS(Processor Complex)

ARM

RISC-V®

2 ARES (Networking Complex)

Scheduling QoS

Congestion Control

Management

I/O Device Types

External Host: LAN/RDMA, NVMe, EIO
Controller Host: LAN/RDMA. CHQM

Tx/Rx Packet Processing

- Match/action (OVS) offload (parse, encap, decap, route, nat, mirror, acl, firewall, policer, sharper)
- Packet buttering
- Telemetry

Wire side crypto

Technology chiplet

Physical Design

- Physical design is the most resource intensive portfolio. It requires state-of-the-art technology and software to get the job done and that is why a lot of capital, human and machine resources are spent on physical design. That is why DreamBig has spent a lot of time and capital to develop world-class in-house capability to make sure the tape-out is as innovative as the design itself.

- **In-House Complete RTL-to-GDS Capability:**
    - Utilizing most advanced process nodes and EDA tools
    - In-house specialized techniques to overcome challenges in advance nodes
    - Expertise from around the world:
        - Constraint design
        - DFT Insertion
        - Topographical synthesis
        - Placement, CTS and routing
        - Signoff

DreamBig SmartNIC Features

Connectivity:

- PCIe 5.0/CXL 2.0
- 25/50/100/200/400/800 GbE network ports

Performance:

- 800 Gbps packet throughput

Virtualization:

- SR-IOV with PFs and VFs

Offloads:

- Programable packet parser
- Programable hierarchal schedular
- Checksum, LSO (with/without tunneling) offloads
- RSS multi-queue packet receive logic
- SDN acceleration with Match/Action offload
- IPsec tunnel and transport offload with AES-GCM
- RDMA over Converged Ethernet (RoCE v2) with RC and UD

Join Our World Class Team

- **Hardware Development Areas:**
  - High Performance, Low power ASIC Design
  - SoC - Integration of Cutting Edge IPs (PCIe 5.0/6.0, CXL 2.0/3.0, 800G Ethernet)
  - Micro Architecture and Logic Design
  - RTL Design using Verilog and SystemVerilog
  - HW Verification - UVM, Formal Verification
  - FPGA Prototyping
  - Design tools for Simulation, Synthesis, Timing Analysis and RTL Checking (Lint, CDC/RDC, LEC)
  - Silicon Validation and Board Design
  - Backend physical design team (4nm)
- **Software Development Areas:**
  - Linux kernel programing
  - Device drivers (specifically network drivers)
  - Networking stacks (L2/L3/L4)
  - Switching and routing (vSwitch and OVS)
  - RDMA (Remote Direct Memory Access)
  - Storage and NVMe
  - DPDK (Data Plane Development Kit)
  - Firmware Development

Official Website

LinkedIn page

For more information write to
info@dreambigsemi.com
hr@dreambigsemi.com

Q & A