



UNLEASH THE REVOLUTION IN NEXT-GEN COMPUTING

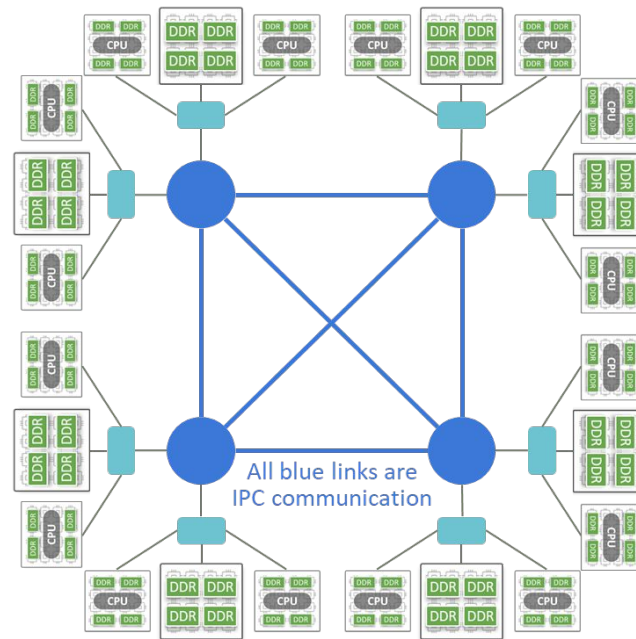
Foundational Networking Silicon for the Accelerated Computing Era



SHRIJEET MUKHERJEE

SHRIJEET@ENFABRICA.NET

:: the Supers : mainframe, ccNuma



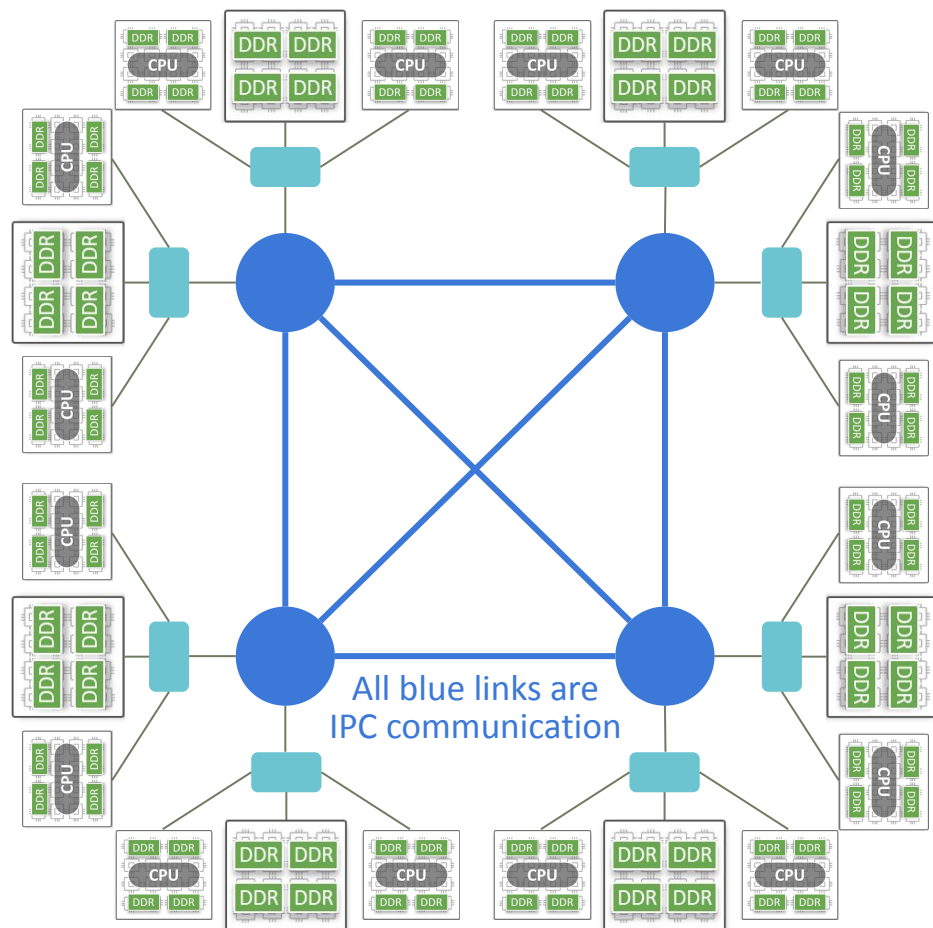
NUMALink 5	2009	7.5 GB/s	500ns (max)	Altix UV
NUMALink 8	2017	13.3 GB/s	300ns (max)	HPE Superdom e Flex

https://en.wikipedia.org/wiki/List_of_interface_bit_rates,

https://www.researchgate.net/figure/Performance-of-random-ordered-ring-latency-for-Altix-POWER5-and-ICE-systems_fig4_220782298,

<https://www.cs.umd.edu/class/spring2017/cmsc714/Readings/SGI-UV-4192.pdf>

:: the Supers : mainframe, ccNuma



Optimal performance is at the “node” level

- 2-4 processor sockets
- SMP shared memory
- NUMA was the answer to SMP scale issues

Cross node performance based on “network performance”

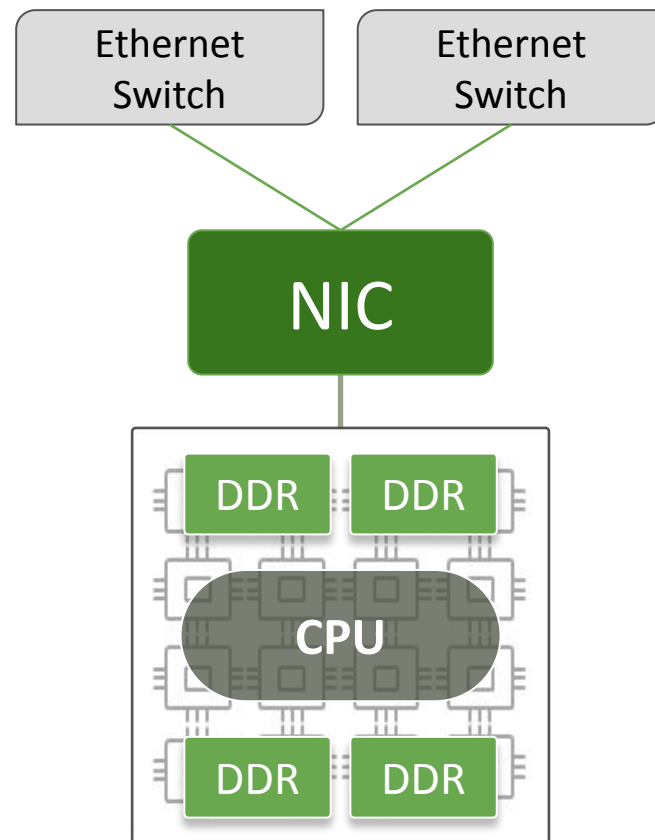
- Multiple paths
- latencies in sub-microsecond range
- All accesses are IPC in nature

Entire system memory is coherent

- Variable latency at high scales
- Needs Torus, dragonfly style topologies
- Needs efficient page locality management

https://en.wikipedia.org/wiki/List_of_interface_bit_rates,
https://www.researchgate.net/figure/Performance-of-random-ordered-ring-latency-for-Altix-POWER5-and-ICE-systems_fig4_220782298,
<https://www.cs.umd.edu/class/spring2017/cmcs714/Readings/SGI-UV-4192.pdf>,
https://wiki.preterhuman.net/SGI-Origin_2000#/media/File:Sgi-origin-2000-dual-rack-uni-koln.jpg,
<http://condor.cc.ku.edu/~grobe/docs/sgi-short-intro/index.shtml#SSMP>

:: borg : rise of scale out computing



Tightly coupled coherent systems scale complexity exponentially

Cross fabric scheduling complexity

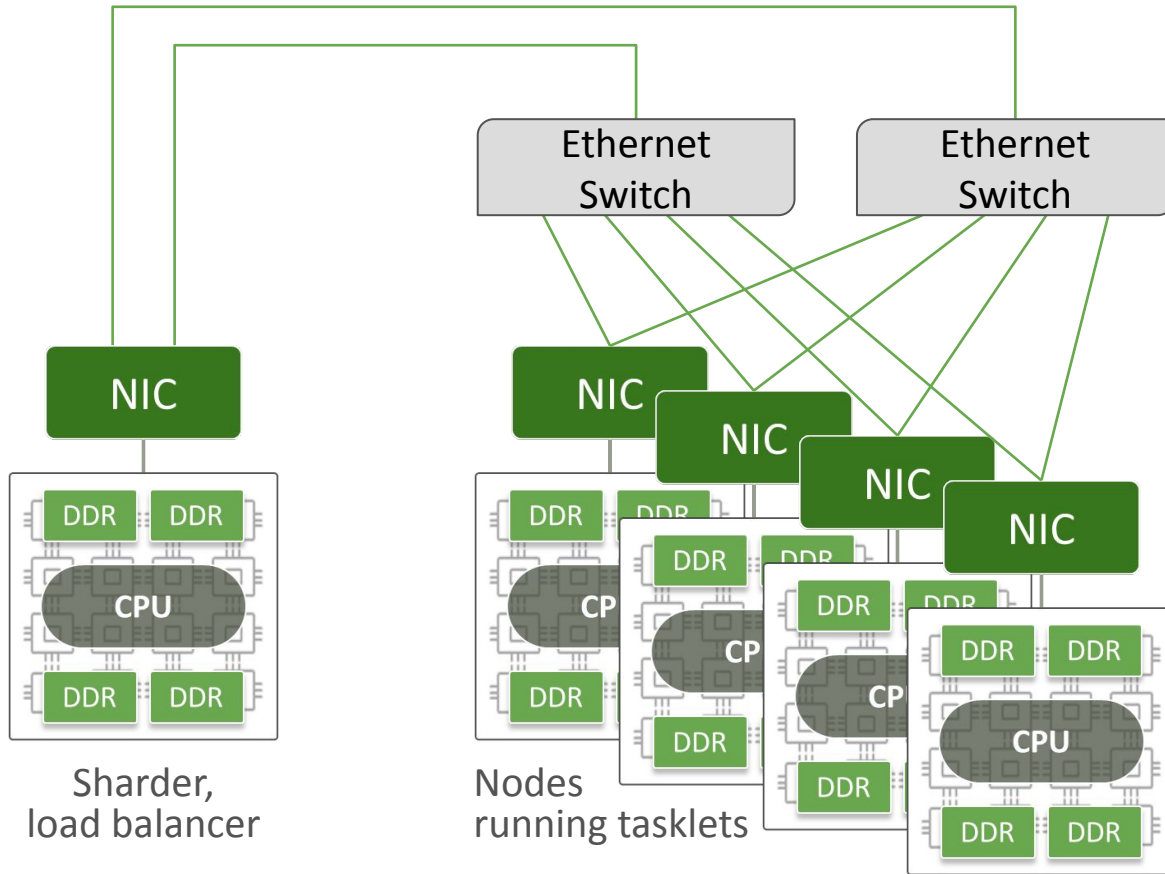
Application level tuning complexity

- Static core pinning – exposed system placement to user
- OS driven placement – high system utilization but variable performance

The answer – Borg

Linux TCP stack	2009	1.25 GB/s	50ms	UCS VIC Intel ixgbe
RDMA on RoCEv2	2018	12.5 GB/s	1-5us	Mellanox Intel

:: borg : rise of scale out computing



All green links are RPC communication

Processing tasks assembled out of tasklets

Tasklets are combined using RPCs to build processing pipelines

- Stages of pipeline can live anywhere

Nodes are sized to house entire tasklet (or multiple)

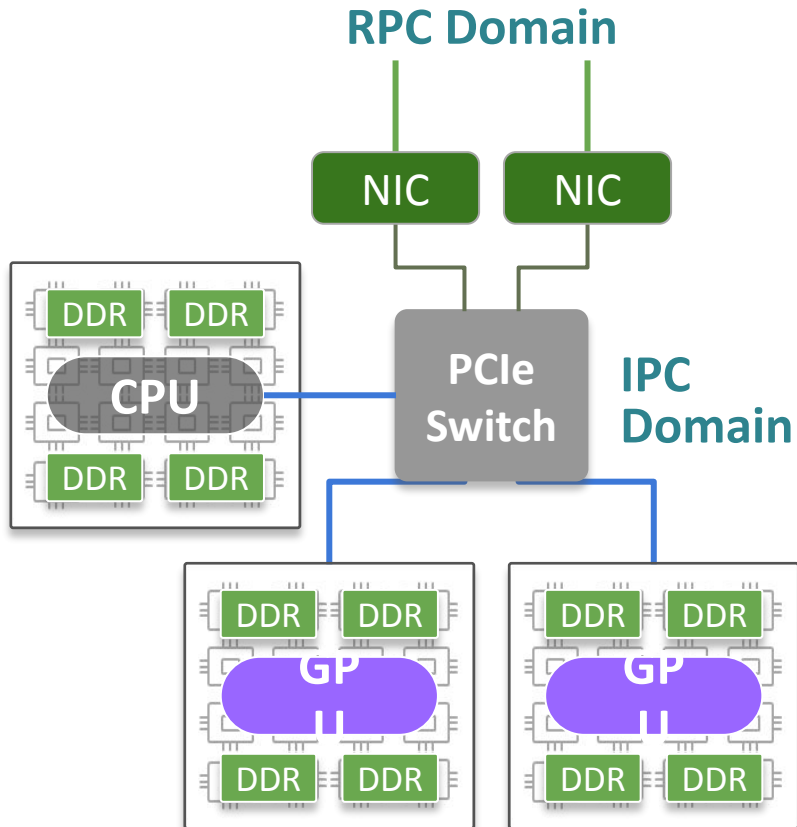
Tasklet's can fail without affecting overall performance

- Redundancy, over-provisioning built-in

Programming model is framework library based

- Enables building RPC boundaries at appropriate interfaces

:: ML monsters : domain specific accelerators



Both models needed to evolve.

A modern truly scalable solution demands

- The tight performance of a supercomputer, with thread scaling
- Resiliency of a cloud scale system

Will adapt the programming model for performance

- Low level (like CUDA, the new assembly language)
- High level, structured kernels for computation and communication

:: let's not “forget” memory

Super computers had strict control over memory hierarchies with tight control

- Load / Store was always the preferred programmer's interface
- Numa machines started stretching these limits

Distributed cloud systems allow wide range of latencies and bandwidth on the network interconnects

- Packet based communication and RDMA are the programmer's interface
- This design is getting stretched by the rapid expansion of memory and coherent communication footprints

Is there a best of both worlds?

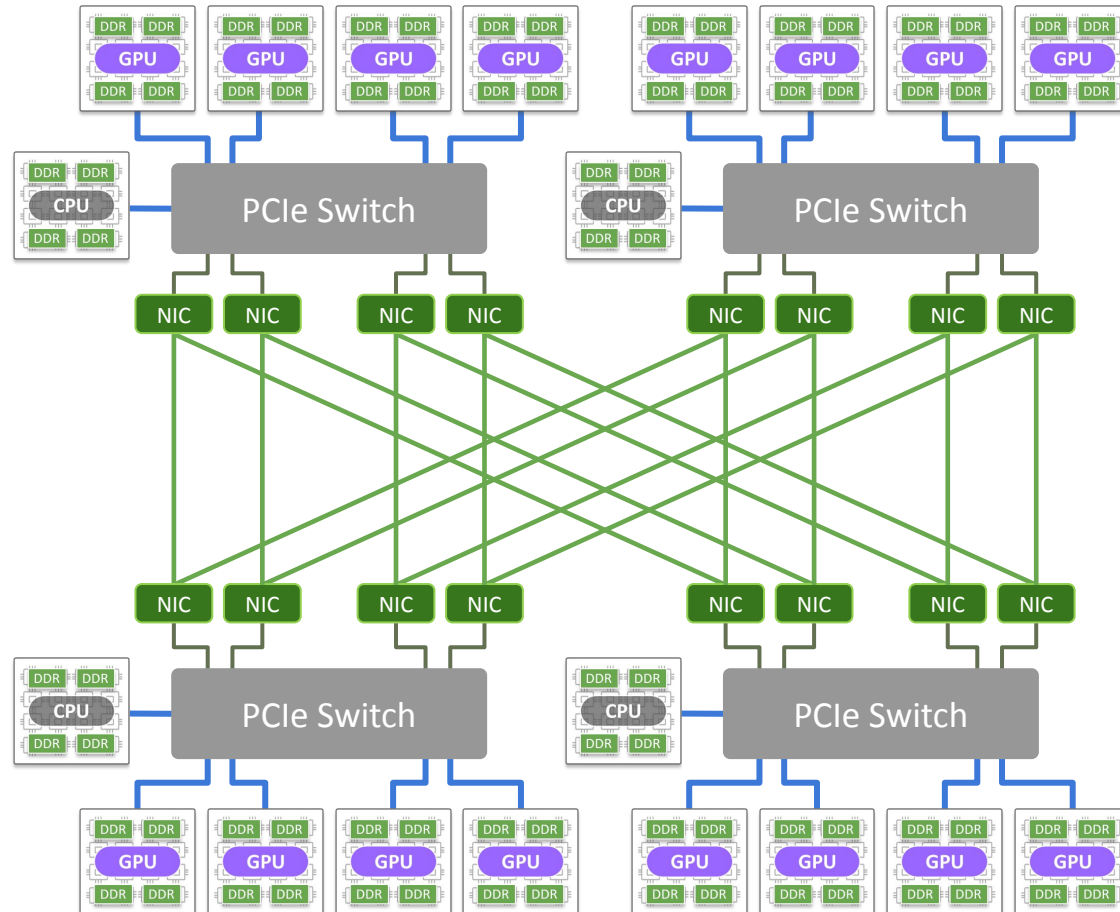
:: let's not "forget" memory

	Latency	Bandwidth / Channel	Max Capacity*	Significance	Programmers View
Reg	0.2ns		KB	In CPU	L1 – dereference pointer
Cache	40ns		KB		L2 – dereference pointer high perf memcpy
DDR (Main)	80-140ns	32-51.2 GB/s (DDR5)	Up to 4TB		
DDR (NUMA)	170-250ns	32-51.2 GB/s (DDR5)	Up to 8TB	CPU independent but local	L3 – dereference pointer high perf memcpy, swap
DDR (CXL)	170-250ns	32-51.2 GB/s (DDR5)	2-4 TB		
DDR (CXL Switched)	300-400ns	32-51.2 GB/s (DDR5)	64TB	Network attached	L4 – memcpy, swap
Far Memory	2-4us	100 GB/s (800g ethernet)	infinite		
SSD	50-100us				L5 – memcpy, swap

:: let's not "forget" memory

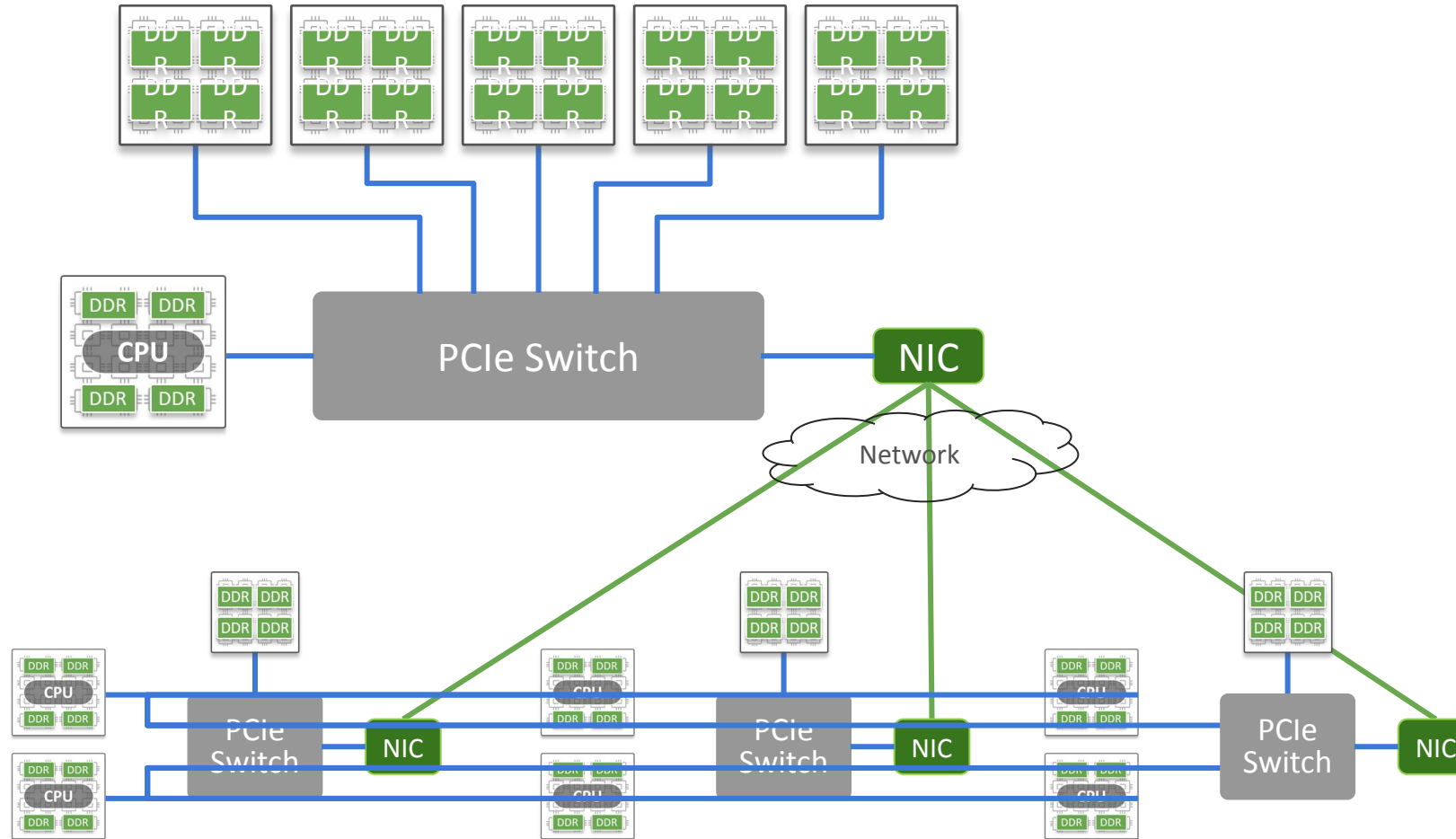
	Latency	Bandwidth / Channel	Max Capacity*	Significance	Programmers View
Reg	0.2ns		KB	In CPU	L1 – dereference pointer
Cache	40ns		KB		L2 – dereference pointer high perf memcpy
DDR (Main)	80-140ns	32-51.2 GB/s (DDR5)	Up to 4TB		
DDR (NUMA)	170-250ns	32-51.2 GB/s (DDR5)	Up to 8TB		
DDR (CXL)	170-250ns	32-51.2 GB/s (DDR5)	2-4 TB	CPU independent but local	L3 – dereference pointer high perf memcpy, swap
DDR (CXL Switched)	300-400ns	32-51.2 GB/s (DDR5)	64TB		
Far Memory	2-4us	100 GB/s (800g ethernet)	infinite	Network attached	L4 – memcpy, swap
SSD	50-100us				L5 – memcpy, swap

:: modern systems : compute accelerator

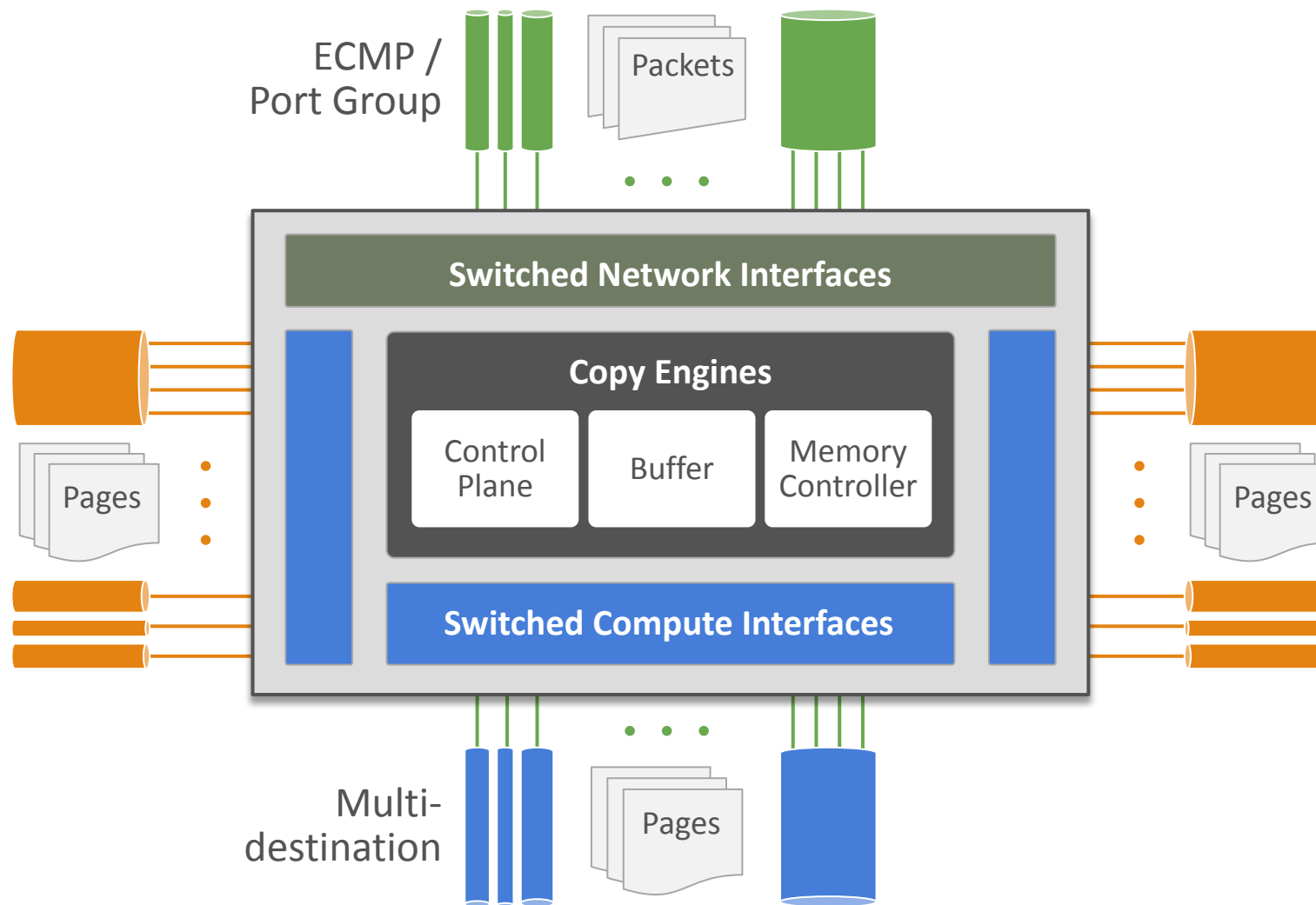


- **Interesting mix of**
 - Scale-out coherent domain
 - Scale-up bulk data movement domain
- **BW on all interfaces are roughly balanced**
- **Ratios of each of the communication types hard wired in the design**
 - CCL's have evolved to try and match the system design
 - Designed for max performance, cost is an afterthought

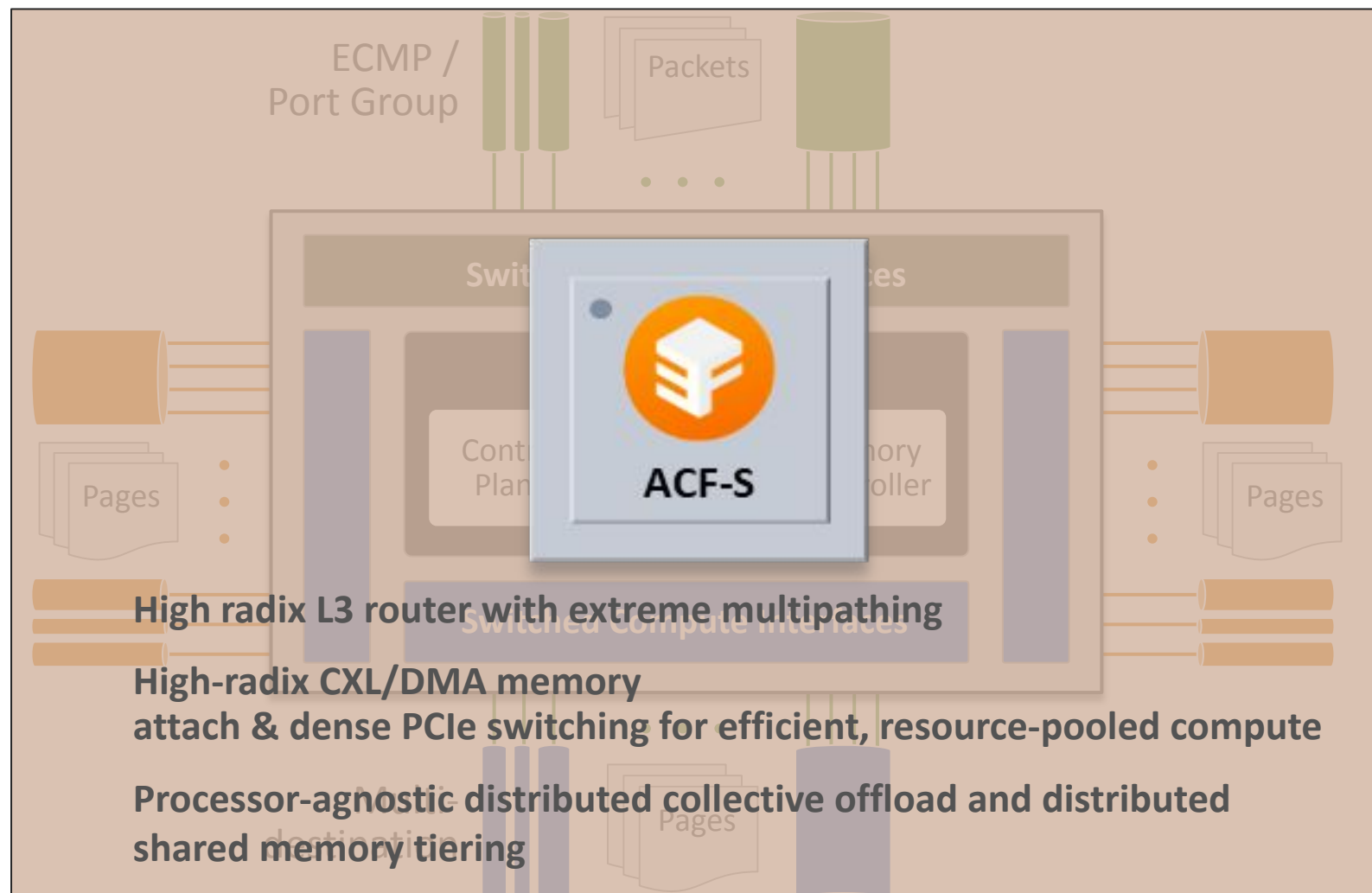
:: modern systems : mem / dbase accelerator



:: accelerated compute fabric - superNIC



:: accelerated compute fabric - superNIC



Multiple 800Gbps ethernet interfaces

Multiple PCIe/CXL x16 interfaces

:: what we are building

ultra-scalable networking silicon & software for high-performance / AI compute



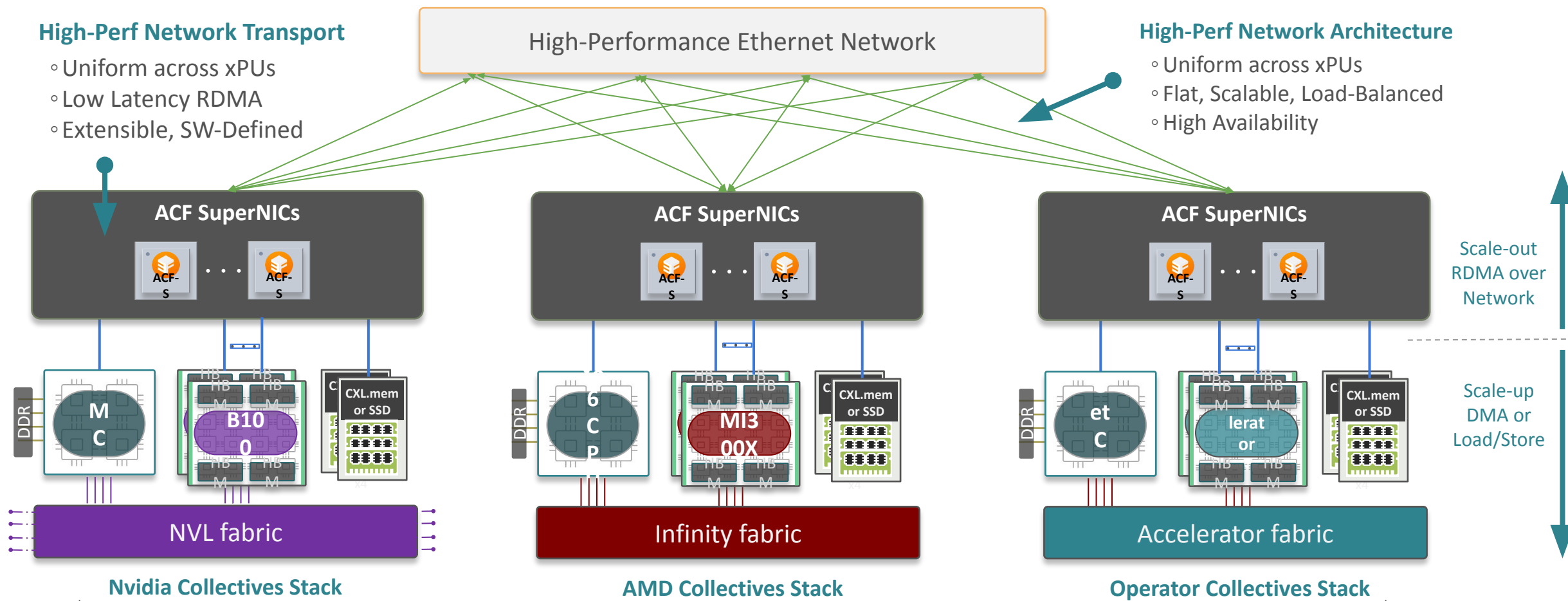
High-Perf Network Transport

- Uniform across xPUs
- Low Latency RDMA
- Extensible, SW-Defined

High-Performance Ethernet Network

High-Perf Network Architecture

- Uniform across xPUs
- Flat, Scalable, Load-Balanced
- High Availability



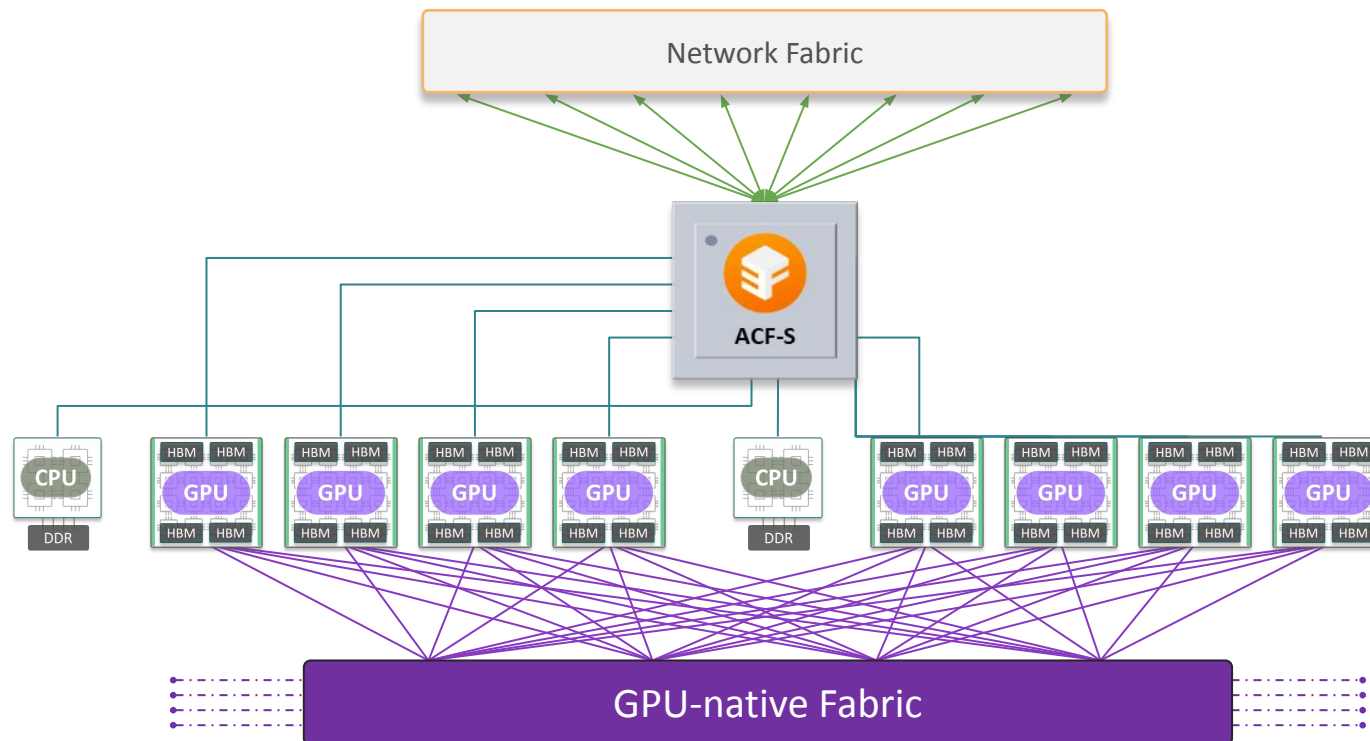
Nvidia Collectives Stack

AMD Collectives Stack

Operator Collectives Stack

Consistent Cluster communication SW Stack : Pytorch □ CCL □ ibverbs □ driver

:: why do it this way?



It's a Collective 8X NIC for Collective GPUs

- 8X scale-out bandwidth of RDMA NICs
- Can load-distribute across GPUs

It Cuts Down Network Latencies

- 50 – 66% fewer device hops
- Better network-to-GPU traffic engineering
- Mitigates incast problem

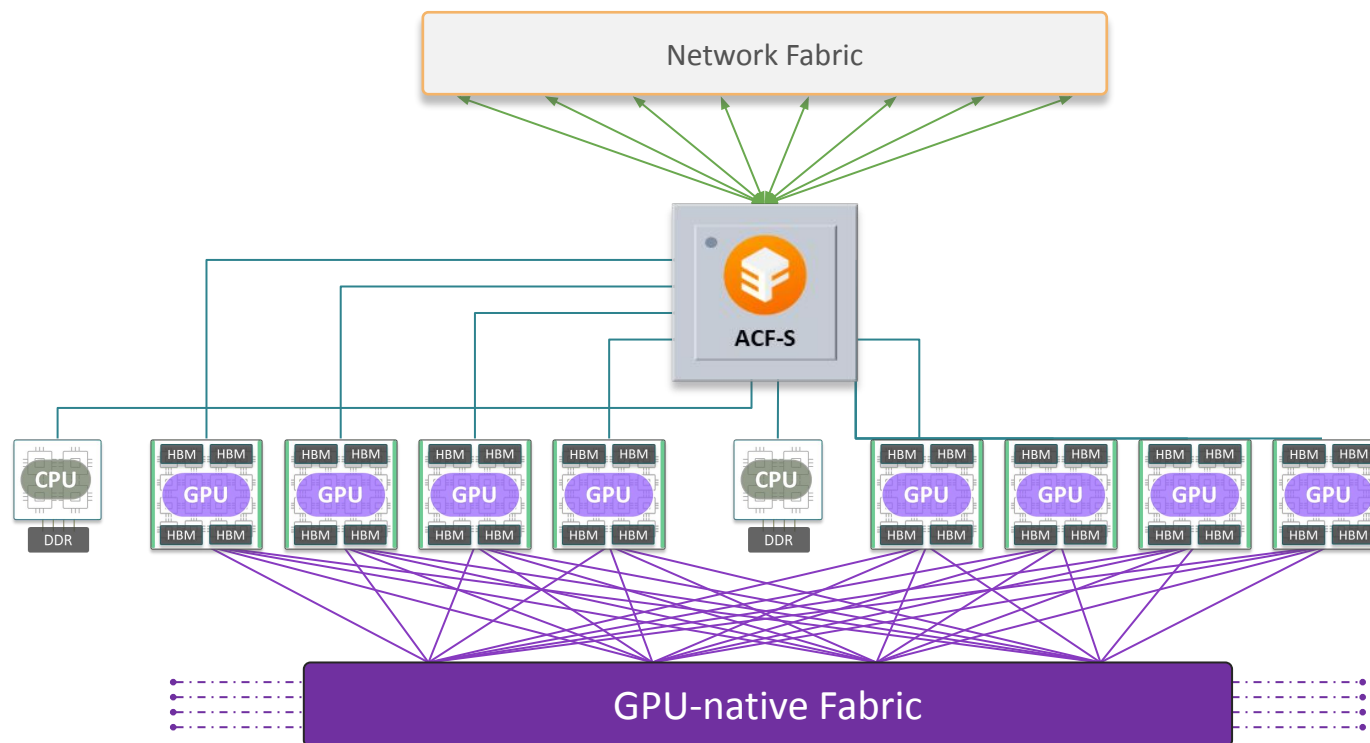
It Lowers AI Cluster TCO

- Allows GPUs to run hotter
- Disaggregates and elasticizes memory

Data movement can, for free

- Shuffle data
- Sparsify or Densify data

:: why do it this way?



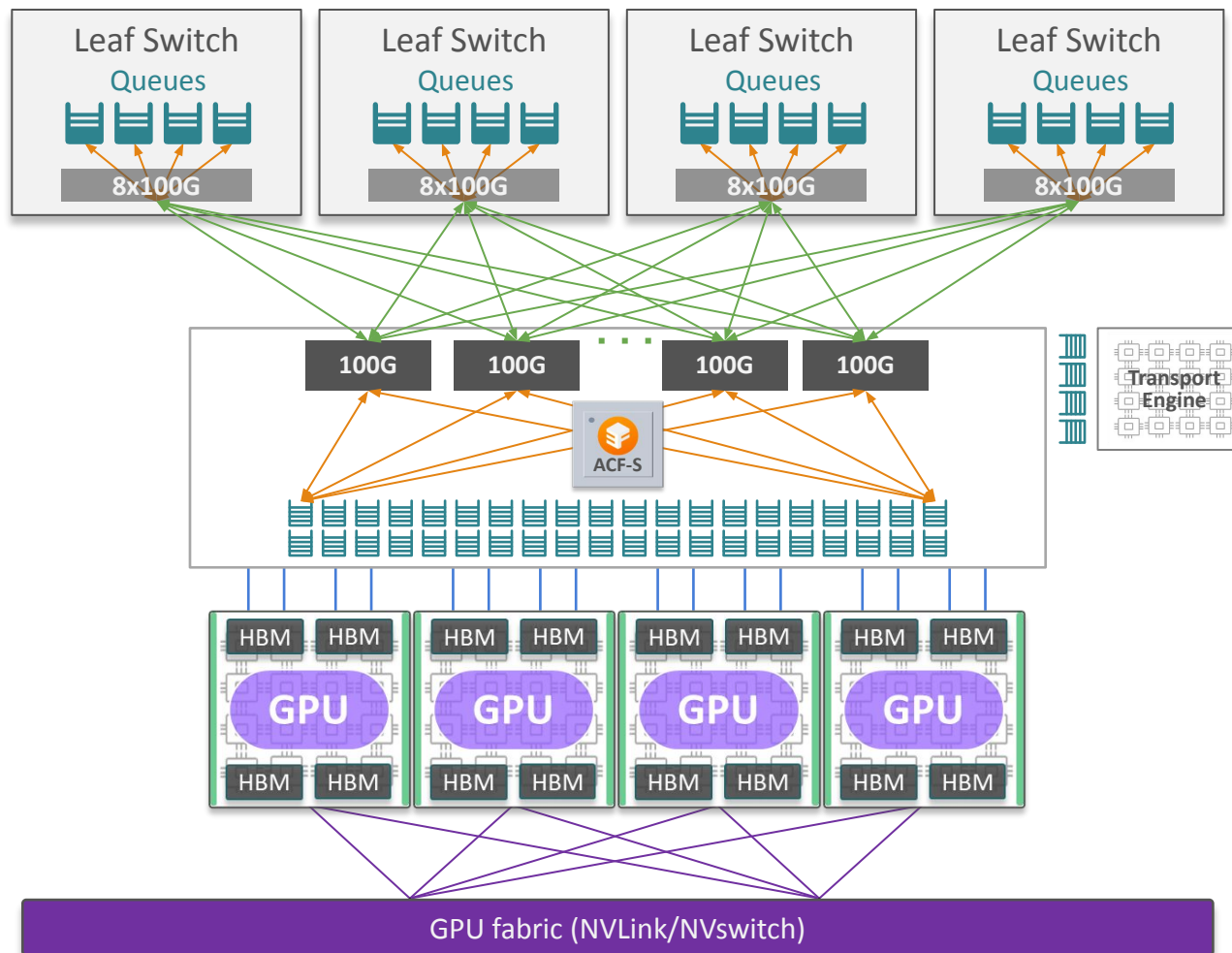
Binds compute facing queues/ops with fabric facing queues

- Queues can communicate in the scale-up domain coherently
- Queues can communicate using RDMA in the scale-out domain

Provides bandwidth matched to local busses

- Effectively acts like the page mover in a NUMA controller using RDMA on the remote links
 - Standard ethernet features line link aggregation and MP routing to provide link scaling

:: scale and resiliency at scale : the game



Multi-railed coherent interconnects

- Enables bandwidth aggregation
- Coupled with sophisticated data movers enables routing around hotspots

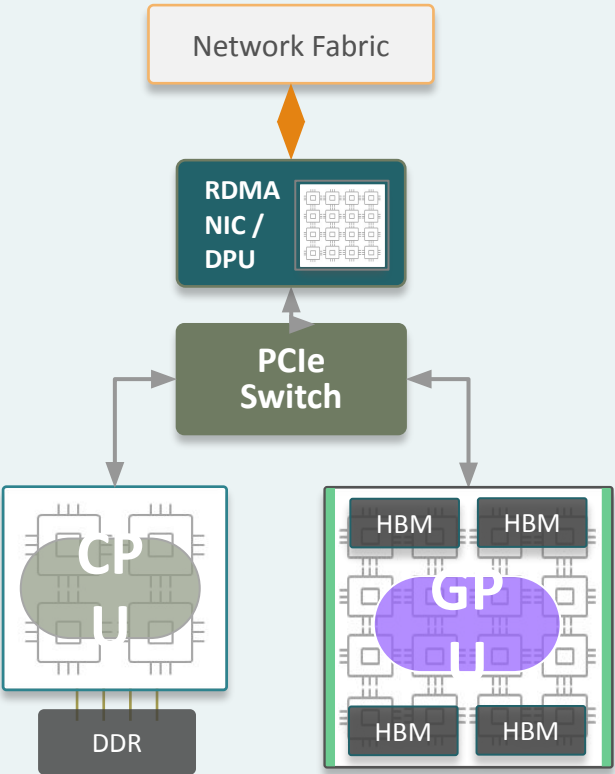
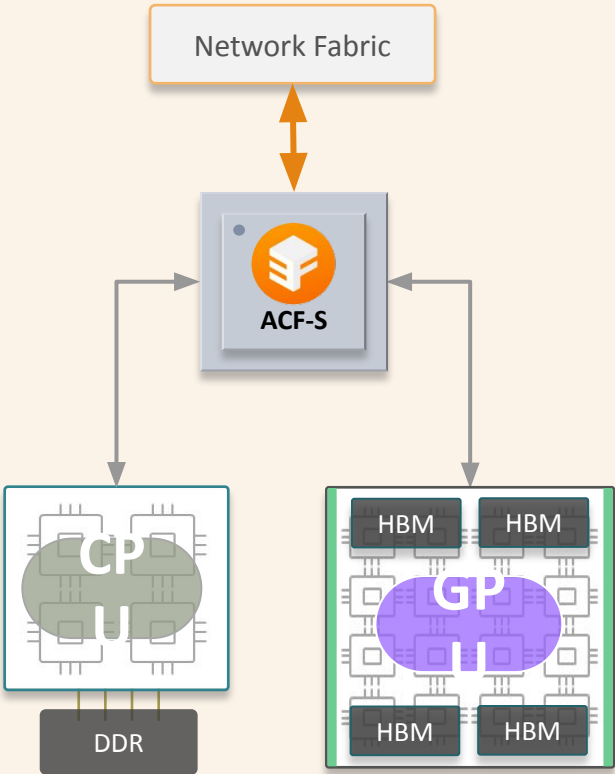
Multi railed ethernet networks

- Enables per packet routing
- Application Queue to network Queue mapping enables precise QOS and rate management

The combination

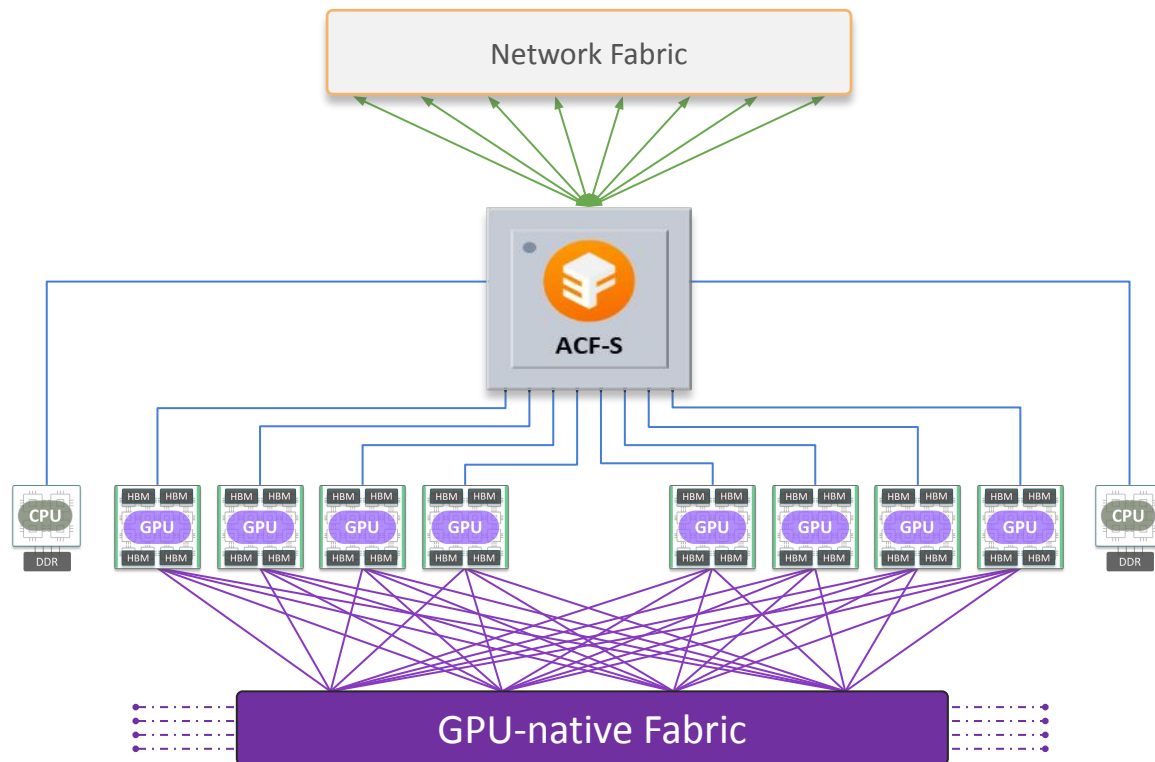
- Build wider networks with lower depth
 - Spread messages over multiple links
- Dramatically increase Resiliency
 - Use network multi-pathing and failover mechanisms to not strand compute

:: conventional nic vs. acf-s

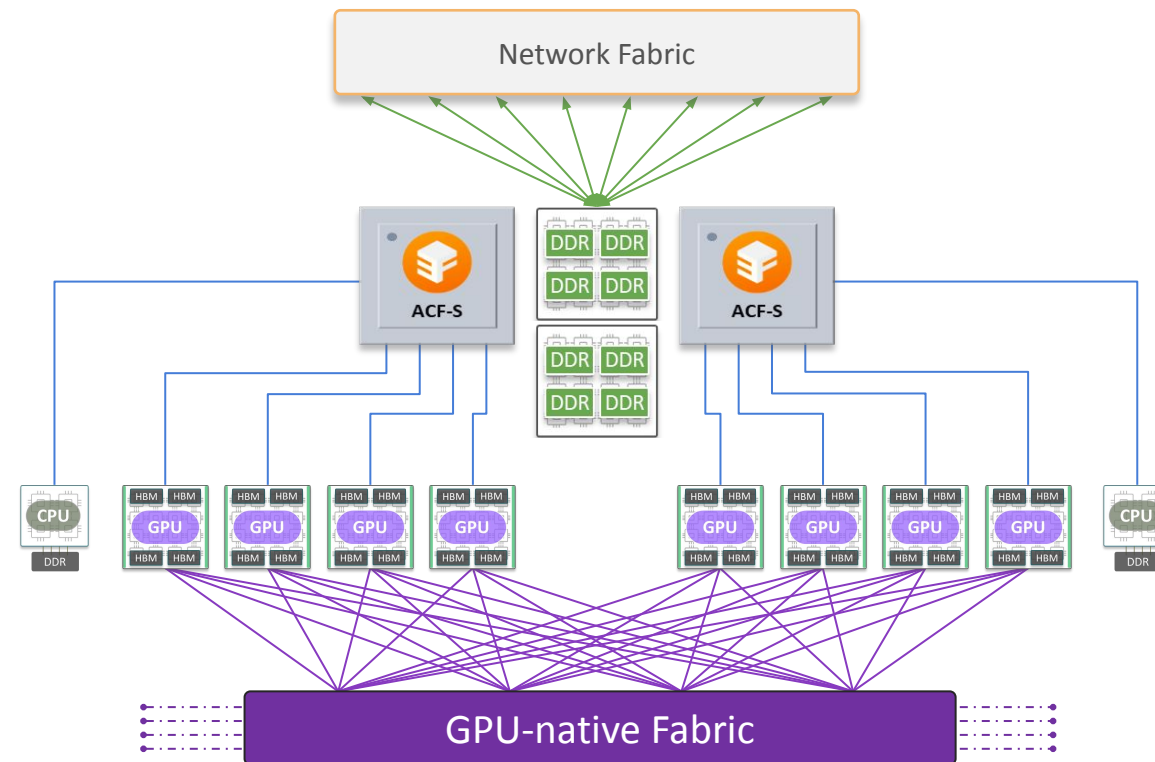
	RDMA NIC / DPU	Solution Property	ACF-S	
	400 Gbps	Network Bandwidth	3.2 Tbps	
	1-2 ports @ 400G	Network Radix	4 ports @ 800G 8 ports @ 400G	
	4 – 8 @ 100G	Single-flow 800G	32 ports @ 100G	
	Not yet	GPU-direct RDMA	Yes	
	Yes	PCIe link aggregation	Yes	
	No	Embedded CXL switch	Yes	
	No	AI Transport	User SW-Defined RoCE, TCP, Spray support	
	Fixed or Vendor-Configured RoCE	AI Transport API	Verbs	
	Verbs		DCQCN or Pacing + Fast Flow Control (FFC); Packet Spray	
	DCQCN + PFC	Congestion avoidance	Verbs	
	Small	Incast Buffer	Large, Shared	

:: use cases : large training fabric

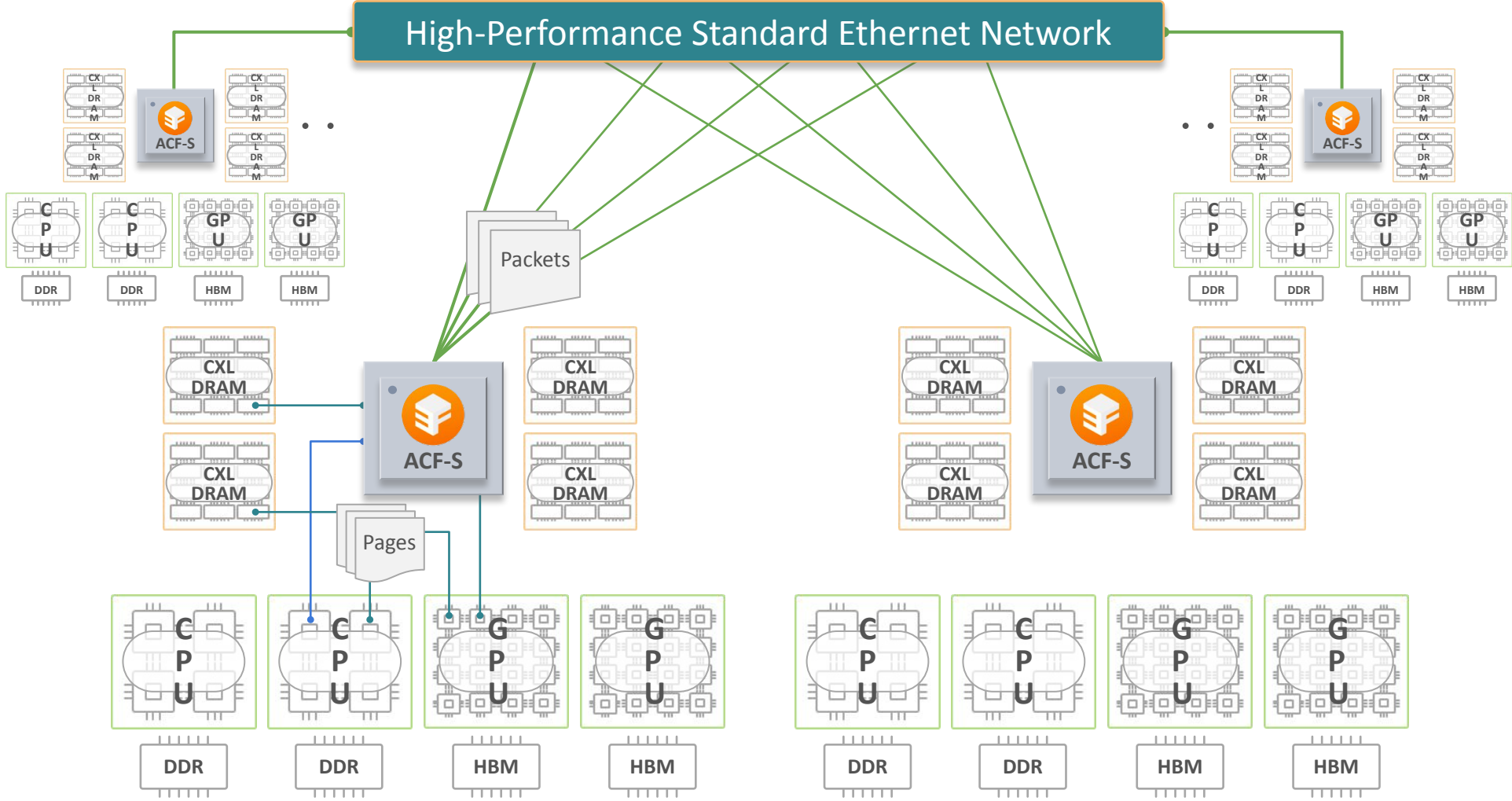
Highly aggregated accelerator node optimized for latency, bw and power



Enhanced node Higher BW, Collective acceleration and optimizer state management



:: ACF as a global memory component

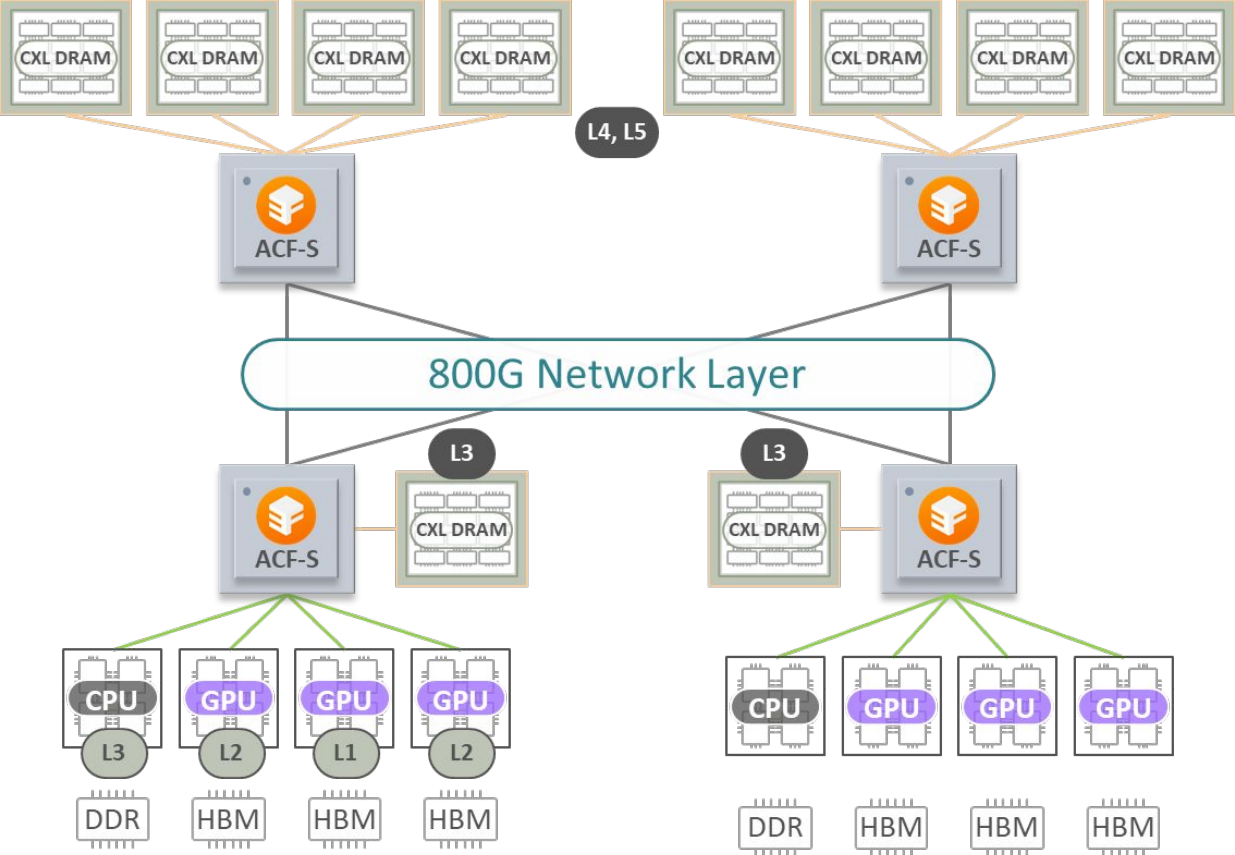


Tier 3
RDMA Network
Memory
1000s of TBs

Tier 2
ACF-S attached
CXL.mem
10s of TBs

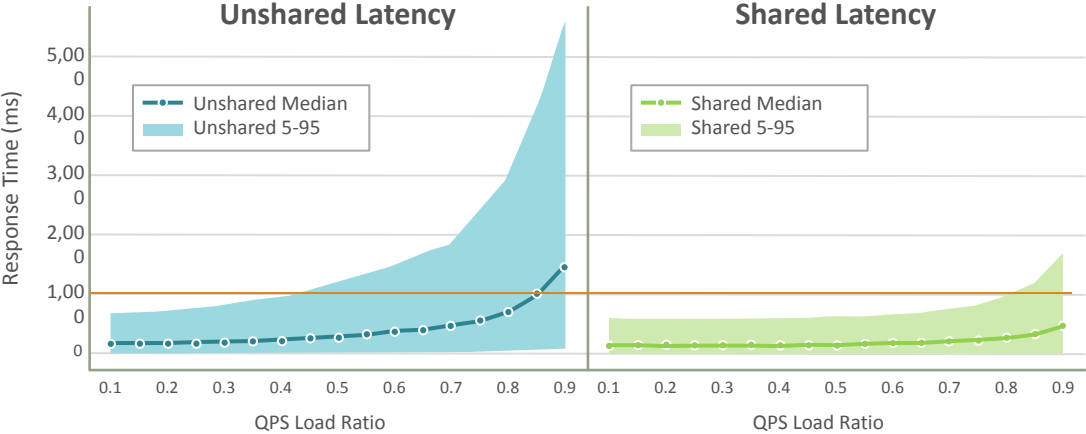
Tier 1
DDR/HBM
1s of TB

:: LLM inference at scale



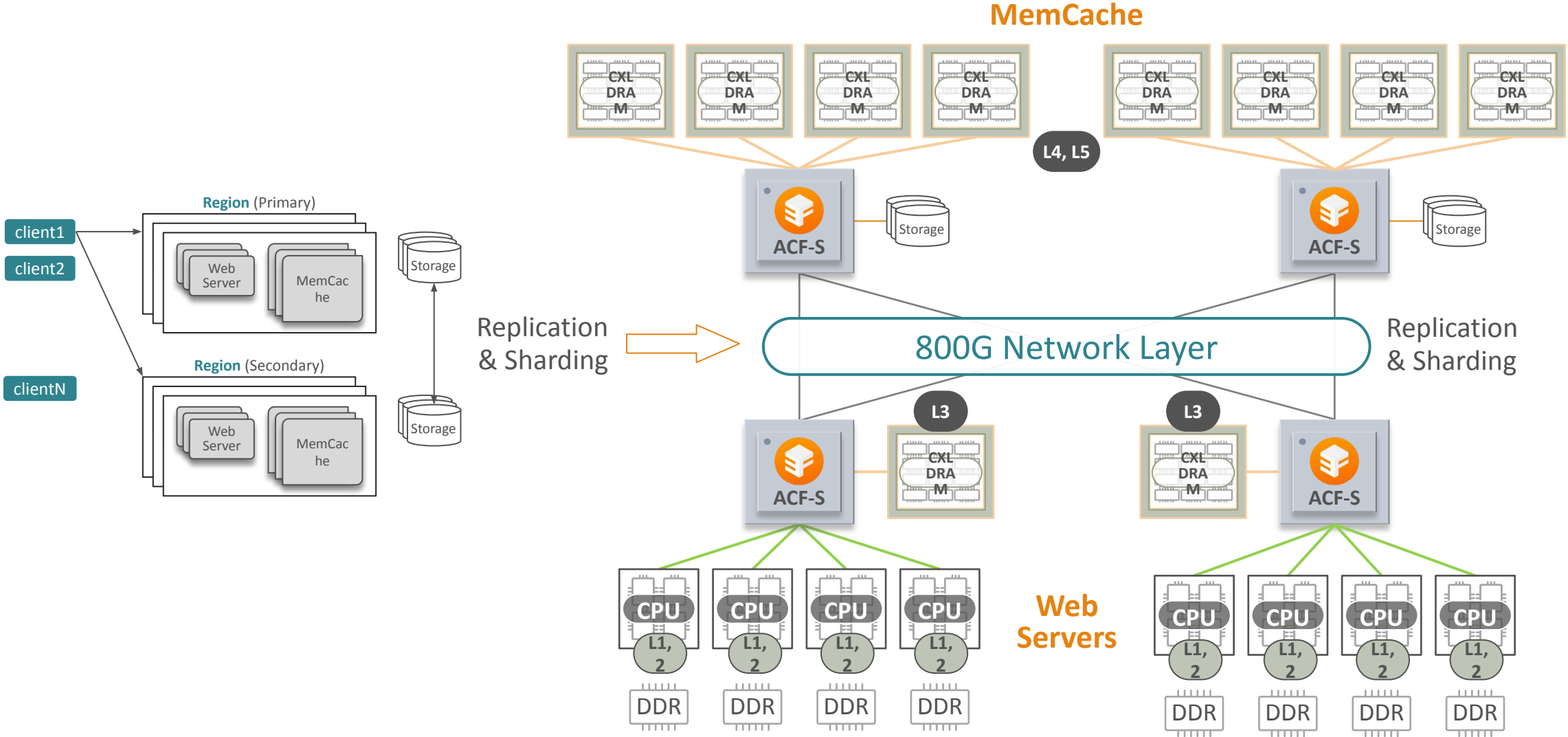
Example DLRM	Required QPS	# Required GPUs	# Required CPUs	User Context Capacity
User Context in CPU DRAM	1K	128	16	80K
User Context in ACF DRAM	1K	64	8	80K

Response Time Distribution, 200ms Avg Response Time with 4 Servers



GPU capacity needs to be severely overprovision to meet latency requirements

:: use cases : in memory dbase



:: 8 terabit/sec acf-s pilot system for customer testing



GPU networking node

- 4 x 800G OSFP Ethernet
- 10 x16 PCIe cabled

In-Network Memory Node

- 4 x 800G OSFP Ethernet
- 8 - 18TB CXL DDR5



8 Tbps AI Networking Node

- Connect any combination of GPUs, CPUs, CXL memory, SSD to network
- Programmable Network Transport: RoCE, RDMA over TCP, UEC-direct
- Replaces NICs, PCIe switches, Ethernet TOR

800G server I/O

- Composable, modular, production-grade

In Manufacturing
Now Orderable

:: rack-and-stack deployment

BUILD RACK 1

Rackables

ACF-S

- Enfabrica 8T ACF-S III

CPU Server

- 1S EPYC x86 Server III
- 1S AmpereOne ARM Server III
- 2S Xeon x86 Server III

GPU Server

- 4x H100 GPU Server III
- 8x H100 GPU Server III
- GH200 Superchip Server III

Slottable Cards

CDFP Port

- CDFP PCIe Extender Card III

CXL Memory

- CXL.mem 2TB DDR5 III
- CXL.mem 4TB DDR5 III

SSD Storage

- NVMe 16TB SSD III

GH200 Superchip Server X
DELETE

DRAG & DROP COMPONENTS

- Enfabrica 8T ACF-S 8
- GH200 Superchip Server
- GH200 Superchip Server
- GH200 Superchip Server
- Enfabrica 8T ACF-S 8
- GH200 Superchip Server
- GH200 Superchip Server
- GH200 Superchip Server
- Enfabrica 8T ACF-S 8
- GH200 Superchip Server
- GH200 Superchip Server

RACKS

- RACK 1
- RACK 2
- RACK 3
- RACK 4

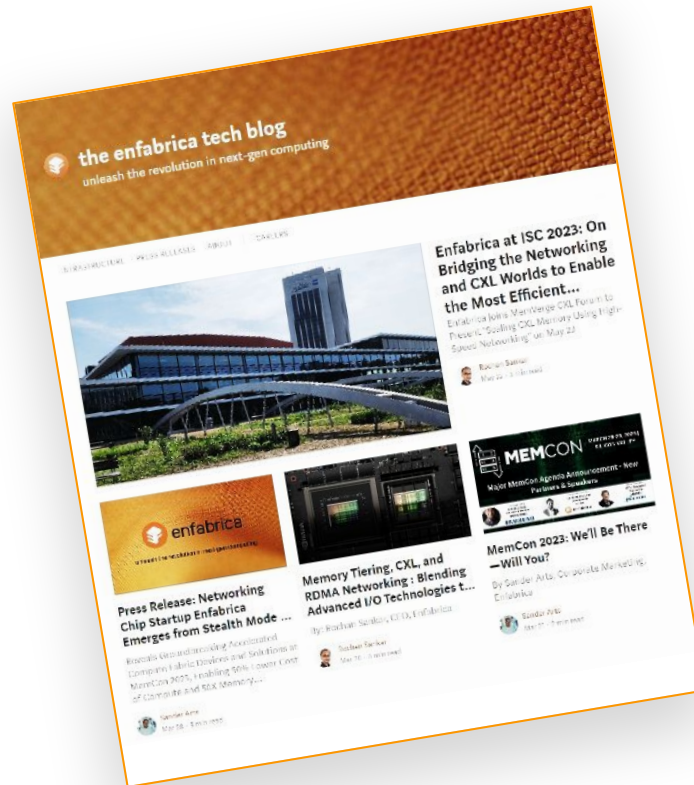
BUILD

:: learn more // engage with us



<https://blog.enfabrica.net/>

<https://enfabrica.net/>



Thank You.

SHRIJEET MUKHERJEE

SHRIJEET@ENFABRICA.NET