Demystifying AI/ML Infrastructure for a Network Engineer

Model counts

Model	Training Token Count	Parameter Count
GPT-2 (OpenAI)	~40 billion tokens	117M, 345M, 762M, 1.5B
GPT-3 (OpenAI)	~300 billion tokens	175 billion
GPT-3.5 (OpenAI)	Estimated ~500 billion+ tokens	~175 billion
GPT-4 (OpenAI)	Estimated 1+ trillion tokens	1.5T model rumored; 220B served
LLaMA 1 (Meta)	1.4 trillion tokens	7B, 13B, 30B, 65B
LLaMA 2 (Meta)	2 trillion tokens	7B, 13B, 70B
Mistral 7B (Mistral AI)	1.5 trillion tokens	7 billion
Mixtral (Mistral AI)	2.6 trillion tokens	12.9B active params (mixture)
Claude 2 (Anthropic)	Estimated 1-2 trillion tokens	~52 billion
Claude 3 (Anthropic)	Unknown (likely >2T tokens)	Unknown (~70-90B rumored)
PaLM 2 (Google DeepMind)	Unknown (PaLM 1 trained on 780B)	340 billion
Gemini 1.5 (Google)	10T+ tokens estimated	Unknown

cisco

Token Count = Knowledge Scope

- Tokens represent the amount of textual (or otherwise tokenized) data the model is trained on. If you train on *more* tokens:
- The model has access to *more facts*, *contexts*, and *edge cases*—i.e., its **breadth of knowledge** increases.
- But this doesn't necessarily mean it *understands* the content better. Think of it as reading more books quickly vs. deeply studying fewer.
- Analogy: Token count is like reading *everything* in a library, but perhaps not always absorbing it all with nuance.

Parameter Count = Cognitive Capacity

- Parameters define the model’s architecture—its “mental horsepower.”
- Increase the model’s ability to represent subtle relationships, abstract patterns, and nuanced reasoning.
- After a certain point, simply adding more parameters yields smaller improvements unless accompanied by more and better data (tokens).
- Analogy: Parameters determine how **deeply and abstractly** you can think about what you’ve read.

Why the Balance Matters

Parameters	Tokens	Effect
Low	High	Quick recall of broad info, but shallow reasoning
High	Low	Sophisticated reasoning, but only on limited knowledge
High	High	Broad knowledge and deep understanding — ideal but resource-intensive
Low	Low	Underpowered and limited — unlikely to perform well

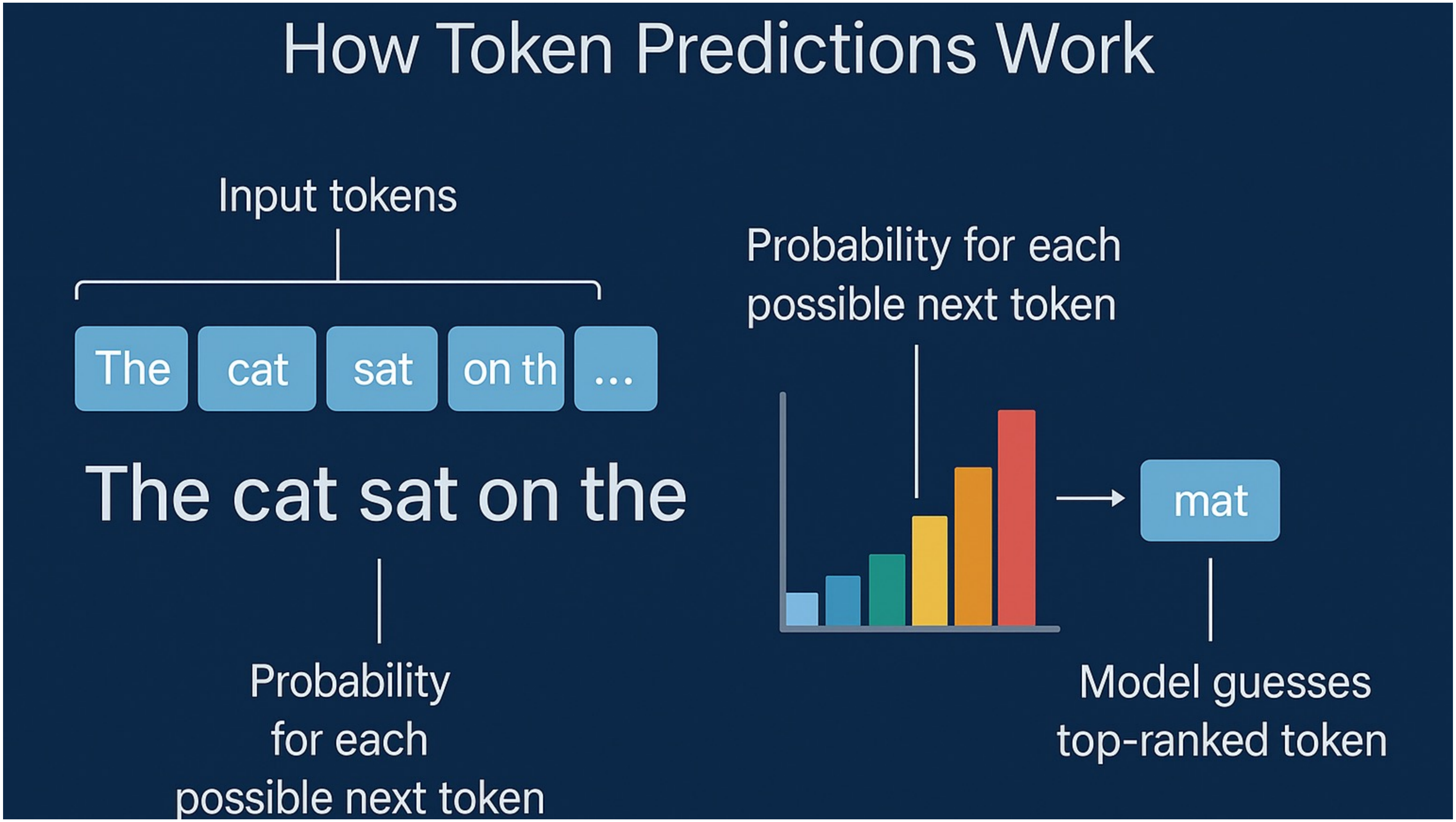
Language Training

Where did all this training data come from???

Dataset	Quantity (tokens)	Weight in training mix
Common Crawl (filtered)	410 billion	60%
WebText2	19 billion	22%
Books1	12 billion	8%
Books2	55 billion	8%
Wikipedia	3 billion	3%

If you train it on billions of documents, it sees tons of examples of:

- Conversations
- Articles
- Code
- Stories
- Instructions
- Explanations
- Questions → Answers
- Comments on Reddit, StackOverflow, books, websites, chat logs, etc.


How Token Predictions Work
Input tokens
The
cat
sat
on th
...
The cat sat on the
Probability
for each
possible next token
Probability for each
possible next token
mat
Model guesses
top-ranked token

But how does it know???

- Now you enter: “How do I configure OSPF on a Cisco router?”

It sees:

- Pattern match: “How do I ___?”
- Cisco + OSPF → has seen that together before

So it generates:

“To configure OSPF on a Cisco router, start by entering global config mode…”

Not because it *understands OSPF* like a network engineer — but because **statistically**, this is what tends to follow that kind of question in real-world documents.

Vectors start out unorganized like these books.

What is a vector?

A **vector** is just a list of numbers representing data in a way AI model can understand:

dog → [5.1, 2.7, 8.3, 0.4, -1.2, ... (300 more numbers)]

cat → [5.0, 2.6, 8.2, 0.5, -1.3, ...]

car → [-0.88, 1.76, -0.05, 0.12, 2.34, ...]

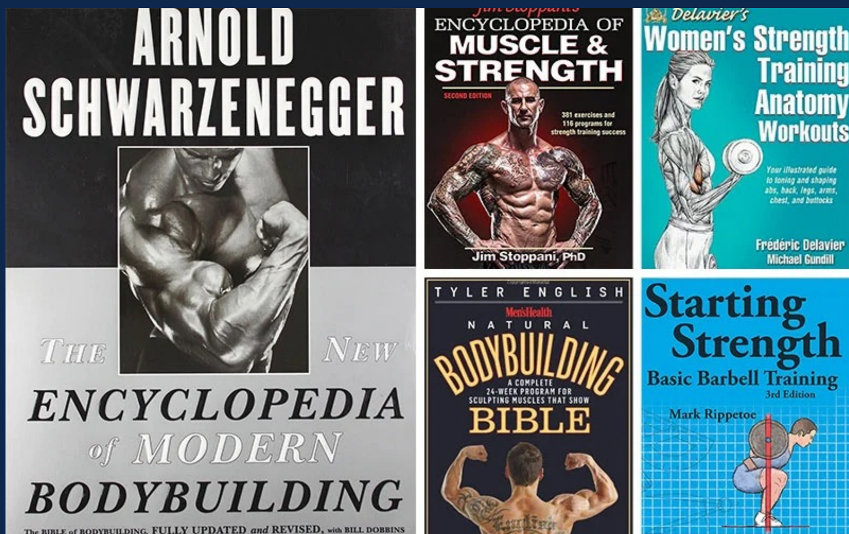
- AI learns that "cat" and "dog" are similar because their vectors are close in value.

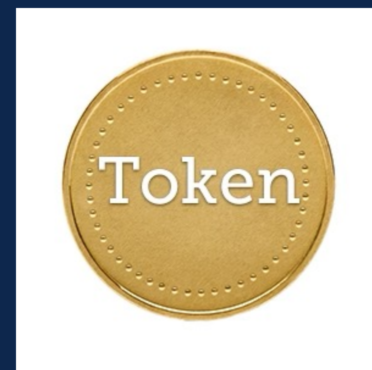
Think of a vector like a digital fingerprint—each piece of data gets its own unique set of numbers!

Dimensions

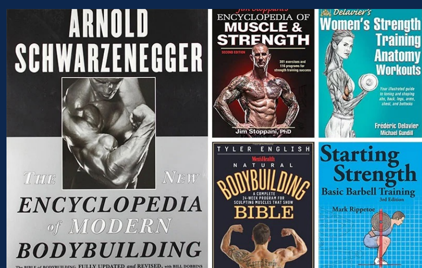
Model	Embedding Size (Dimensions)	Notes
BERT Base	768	Common for Transformer-based models
GPT-2 Small	768	Smaller model, lower resource usage
GPT-2 Medium	1024	More expressive embeddings
GPT-3 (175B)	12,288	Very high-dimensional space
LLaMA 3 (8B)	4096	Efficient scaling for open-source LLMs
LLaMA 3 (70B)	8192	Higher expressiveness, larger model

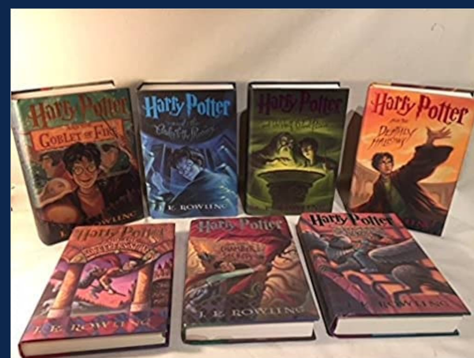
Each word in each book gets assigned a random number.





How are these books similar?????

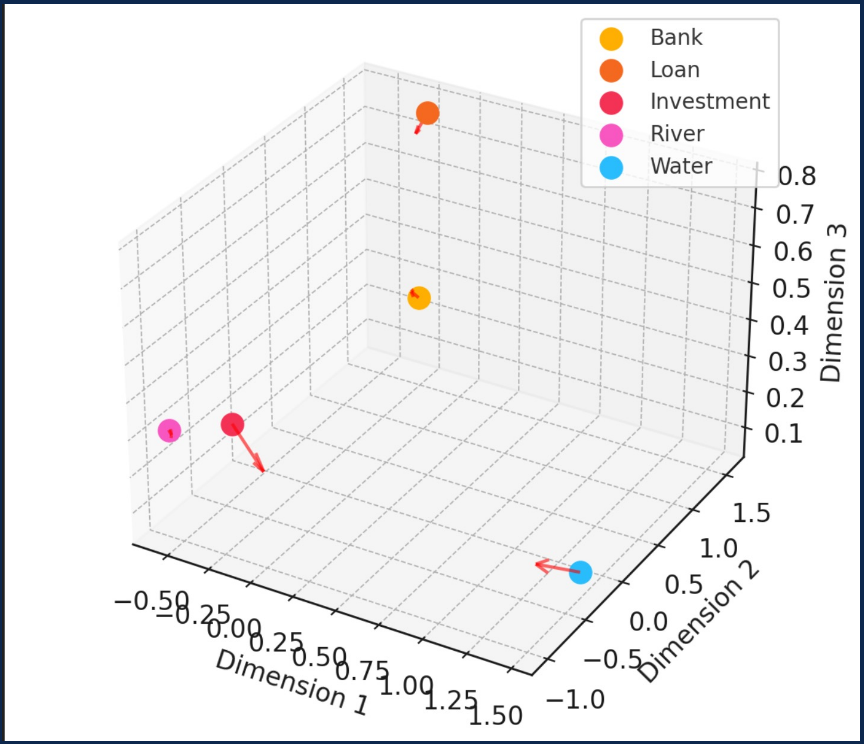




Finding similarities

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5	Dim 6
The Shining	3.275926802538670	10.049703107685100	6.856521725301590	5.5208550884001100	1.8021486759904000	-0.3533350412957710
It	2.837377112952410	8.094839362763870	8.785588187035610	5.76080854148383	1.627714609112290	0.1351970171485700
Carrie	3.1447624985548200	9.215449314305880	6.718232805884650	7.838863026479300	1.5466891796864300	0.5022342744061260
Misery	2.4172151395751900	9.704004299968280	8.05840599810946	6.157953123160340	1.4445381220361200	1.2588415077727400
The Stand	4.089019478042090	7.744102908736430	7.644023387508850	5.6015025615540500	0.8832644041184070	2.1716214922028900
HP Sorcerer's Stone	8.99267028380357	5.587861513670460	3.670375854034910	2.1736200693262100	3.0739971780466700	4.8164768760814800
HP Chamber of Secrets	8.722795035291440	4.245412353674690	3.08930836068898	0.9926960743719830	4.587717261898140	2.733563001993820
HP Prisoner of Azkaban	8.101274054988900	6.746248701388940	3.4060577978745300	1.6042284933931300	2.4077701232792700	2.924171073669700
The Great Gatsby	0.3270716960346710	5.03302711000949	2.2553388161887500	2.194564229258090	1.2588838832778500	5.839393826521570
A Brief History of Time	8.865968215975320	1.8542549855122300	6.821006839146430	7.112720499170980	0.825712008857864	7.350726460924620
The Selfish Gene	9.07325124876977	2.0984911989093300	7.534872691428460	6.38533183685322	2.603835374528130	8.589306240401020
Cosmos	8.619553830227930	1.100096102030730	7.243297039675680	7.913180826520360	2.626868290269870	7.628611695615570
The Elegant Universe	11.722279052250800	2.894385636026790	5.530051976856140	8.272749761921010	1.4016937696966800	8.069248089861050
How Not to Die	6.58913081488134	7.136620110512030	0.6720908450429690	6.397327197183950	3.2776910245822600	1.5827388773745500
The China Study	7.0493665475769800	5.610257120697840	0.08995396021251980	2.162310805908470	2.205508724750230	3.296295228114310
The Obesity Code	5.6922714868023000	5.836811139300240	0.9949563967269250	2.534187732763710	3.0098110371655400	3.00068096015907
The Blue Zones	7.091164190212090	6.196871984266930	1.4517953645869700	7.529889521249060	3.2229141875104200	3.5754620739714300

Visualizing 3D vectors and gradients being bumped



Examples of Data Sets

```
[
  {"image": "image1.jpg", "caption": "A dog running in the park."},
  {"image": "image2.jpg", "caption": "A red car parked on the street."}
]
```

```
text,label
"This product is amazing! I love it!",1
"Terrible experience. The quality is awful.",0
"Best purchase ever! Highly recommend.",1
"Not worth the money. Waste of time.",0
```

```
Patient_ID,Age,Gender,Blood_Pressure,Cholesterol,Diabetes,Heart_Disease,Diagnosis
1001,45,Male,130/85,High,Yes,No,Type 2 Diabetes
1002,62,Female,140/90,Normal,No,Yes,Hypertension
1003,29,Male,120/80,Normal,No,No,Healthy
1004,51,Female,150/95,High,Yes,Yes,Coronary Artery Disease
1005,37,Male,125/85,Borderline,No,No,Healthy
```

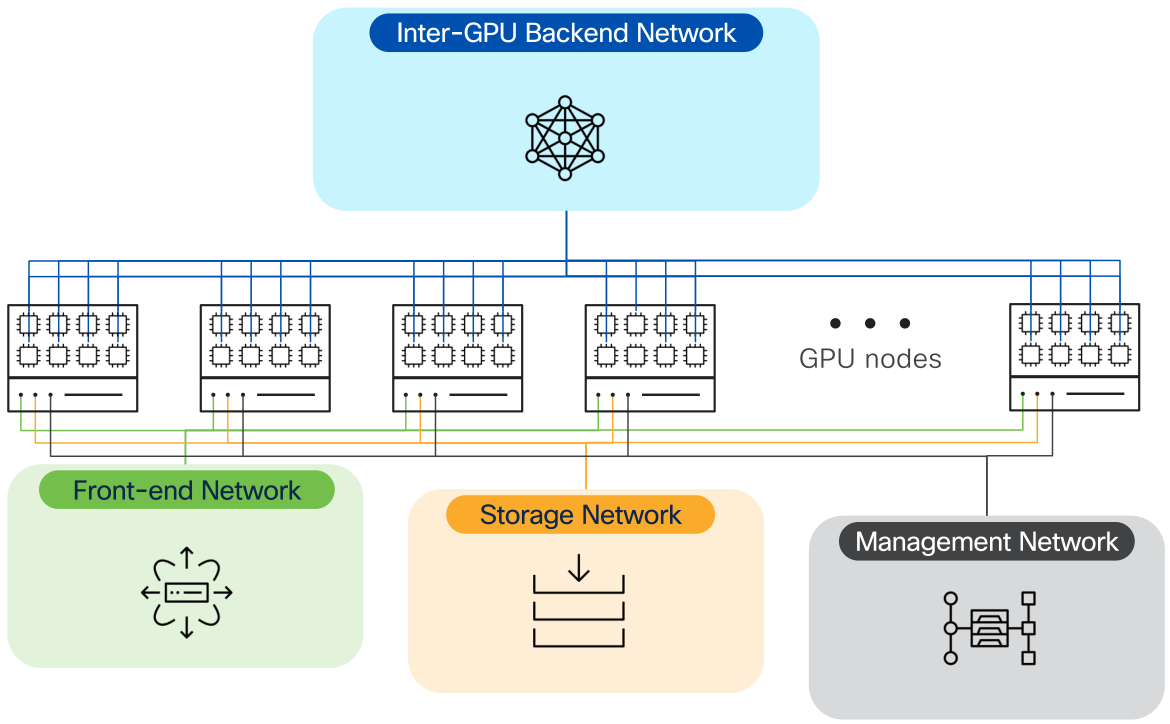
Types of AI Language Models

Model Type	What It Is	Example	Use Case
Base Model	Trained from scratch on raw text	GPT-2, BERT	Generic language modeling
Foundation Model	Large, general-purpose base model	GPT-3, LLaMA	Starting point for many applications
Fine-Tuned Model	Adapted to a specific task	Codex, Alpaca	Code generation, chatbots, QA
Instruction-Tuned	Trained to follow natural language instructions	ChatGPT, Claude	Assistants, tutoring, reasoning

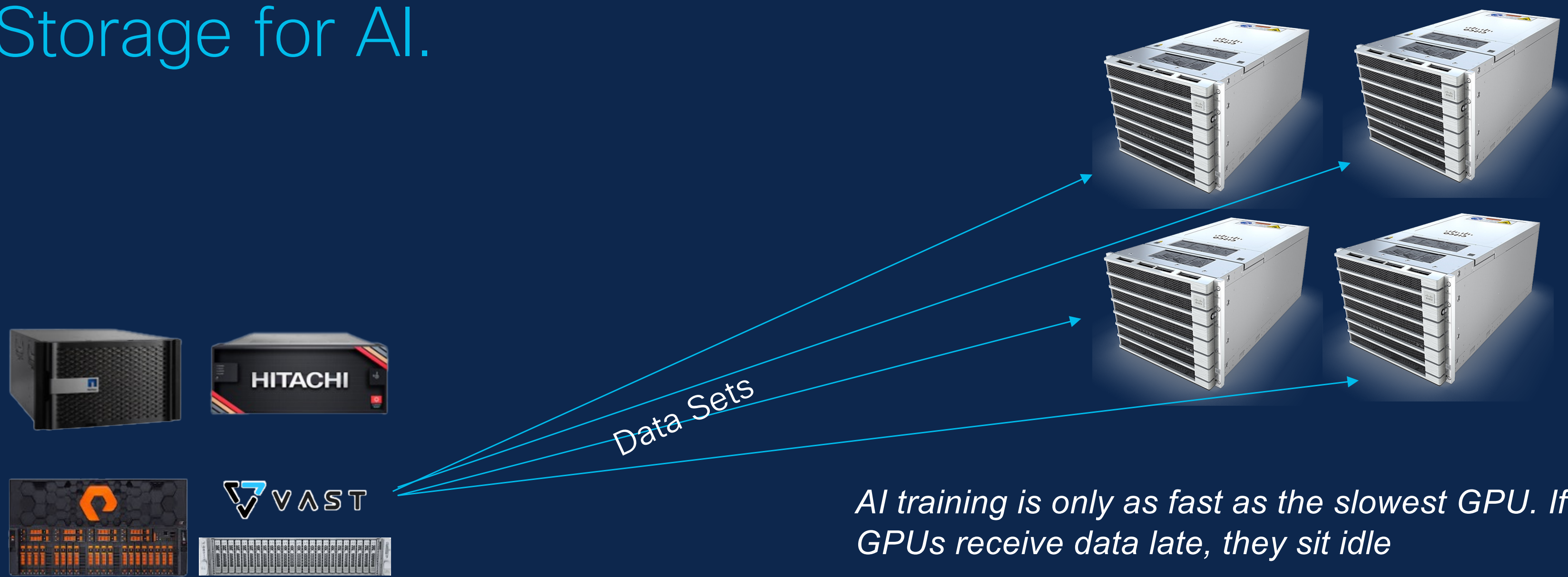
base → foundation → fine-tuned → instruction-tuned
mirrors a real-world development pipeline.


and now our favorite topic plumbing
DATA
NVIDIA
CISCO

Multiple Networks for AI/ML Infrastructure



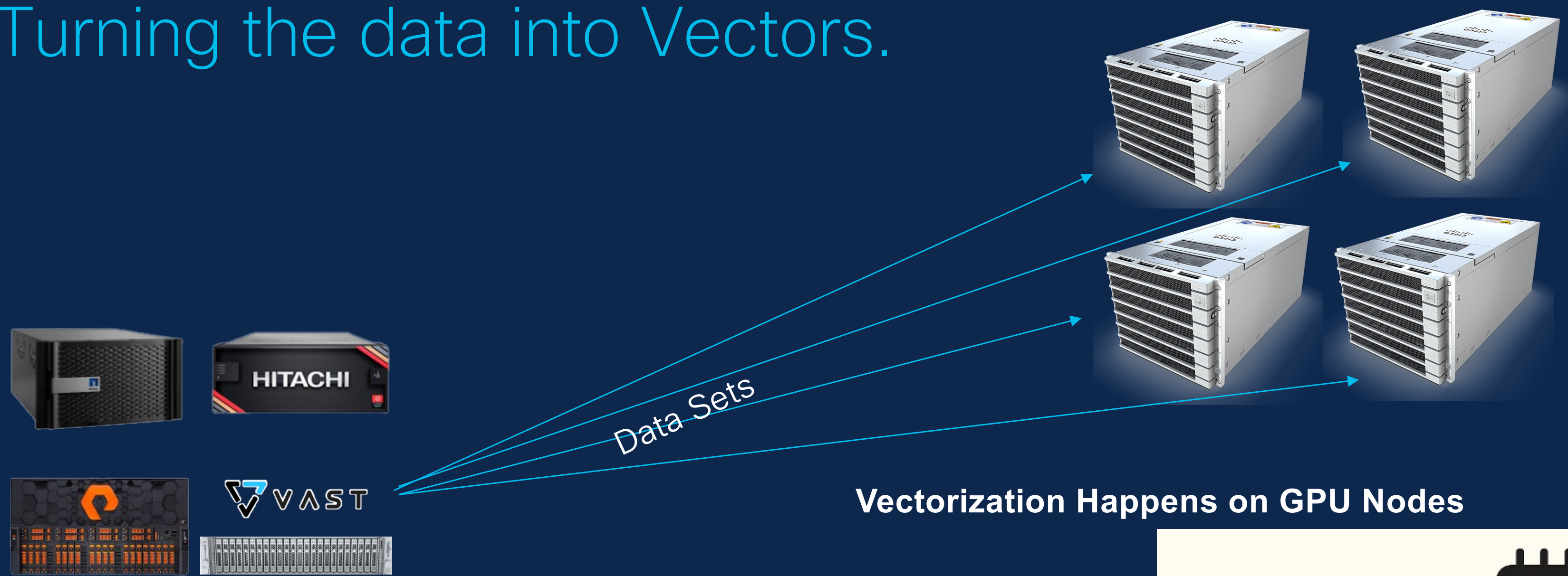
Storage for AI.



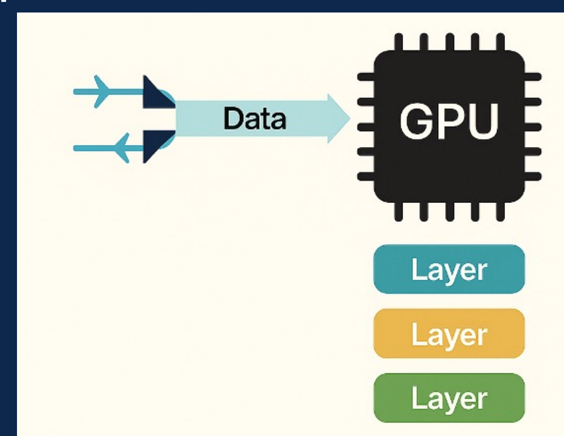
AI training is only as fast as the slowest GPU. If some GPUs receive data late, they sit idle

**Parallel Access:
NO GPU Starvation
Due to Slow Data
Feeds**

Turning the data into Vectors.



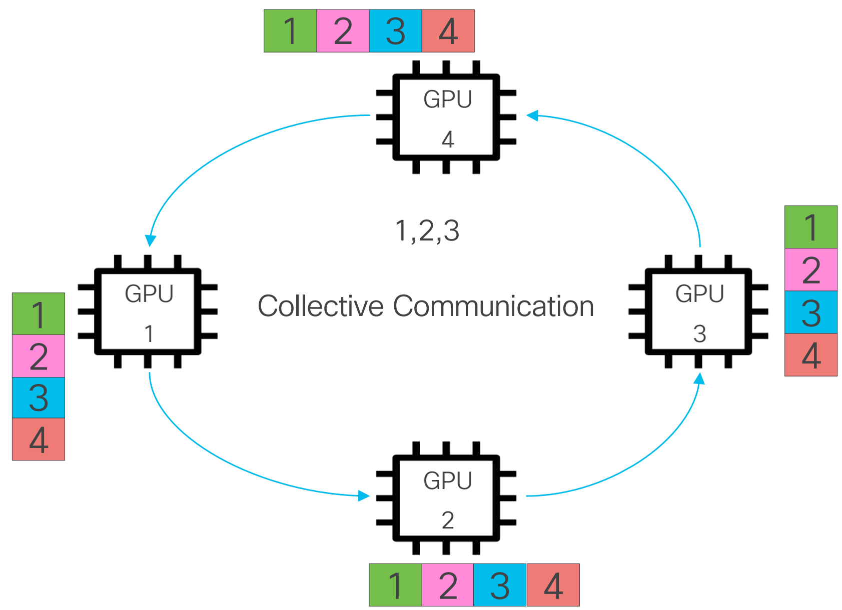
Vectorization Happens on GPU Nodes



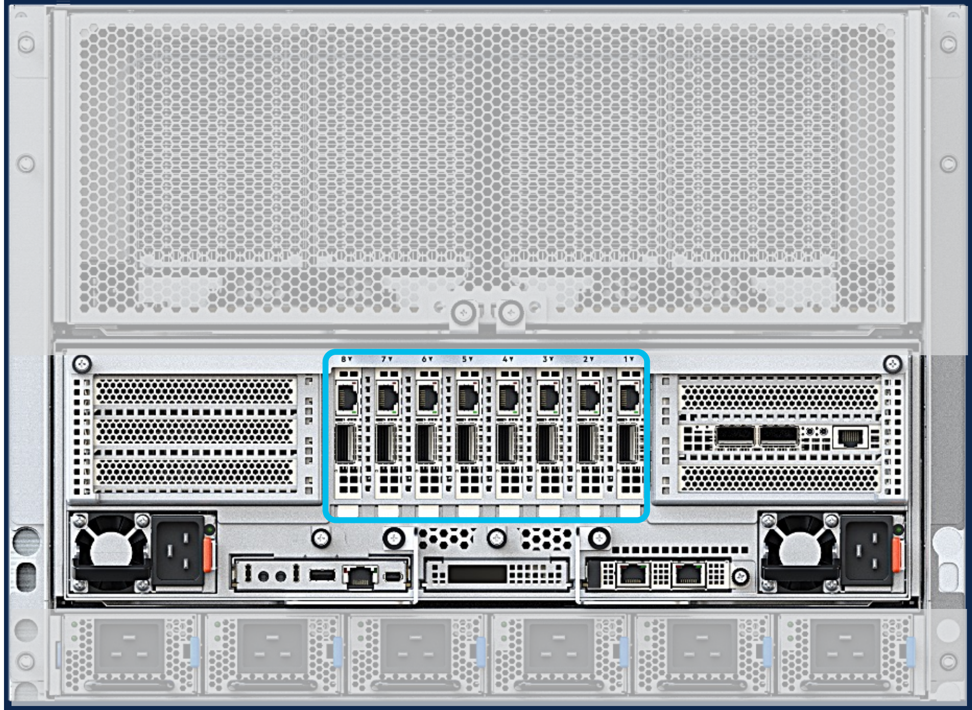
- **Data is retrieved from parallel storage.**
- **GPUs receive data at the same time to avoid idle time.**
- **Each GPU server processes its batch independently, converting data into vectors.**
- **Model training begins immediately.**

All-Reduce: How GPUs Share and Aggregate Gradients

This is basically a group chat where everyone's yelling their gradients until they all agree who's right.

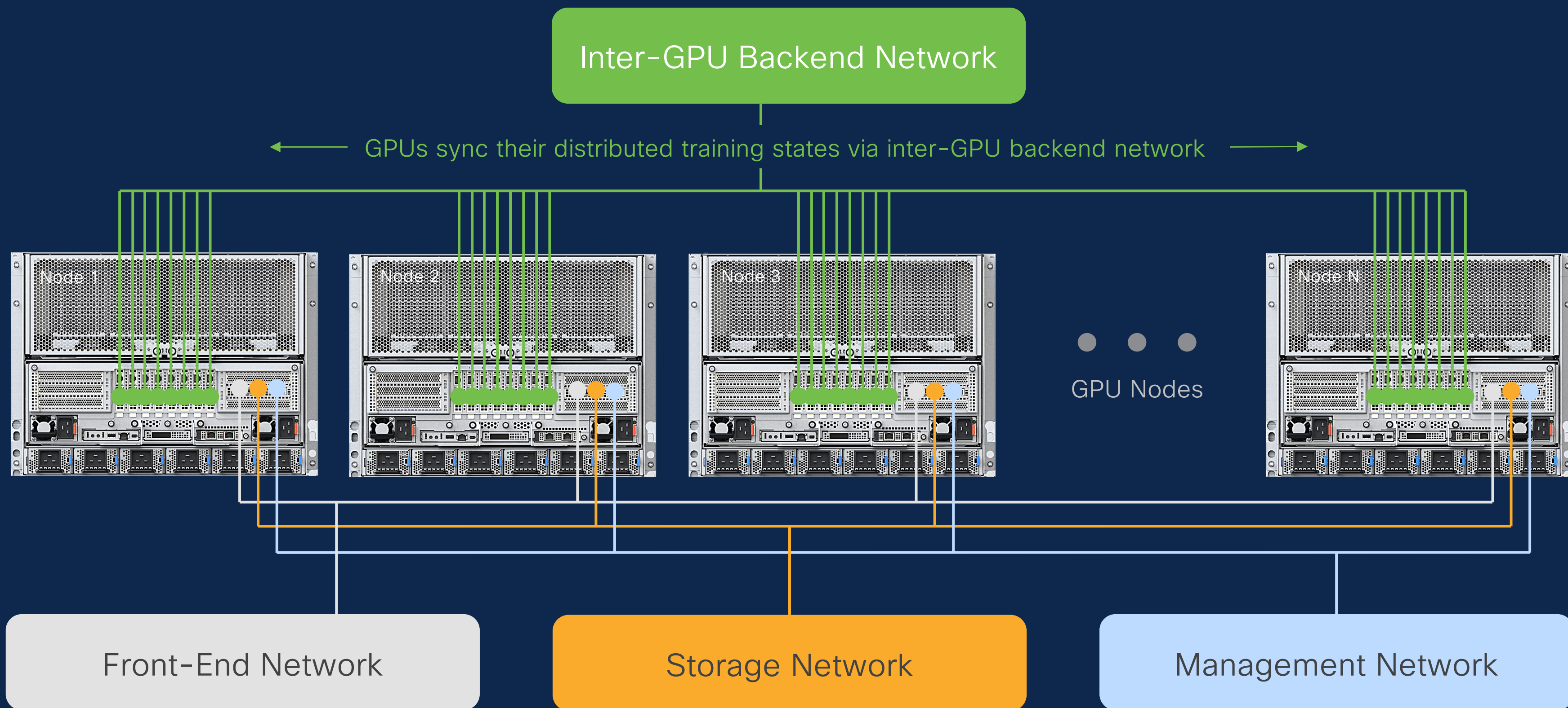


Where GPUs Talk Behind the CPU’s Back

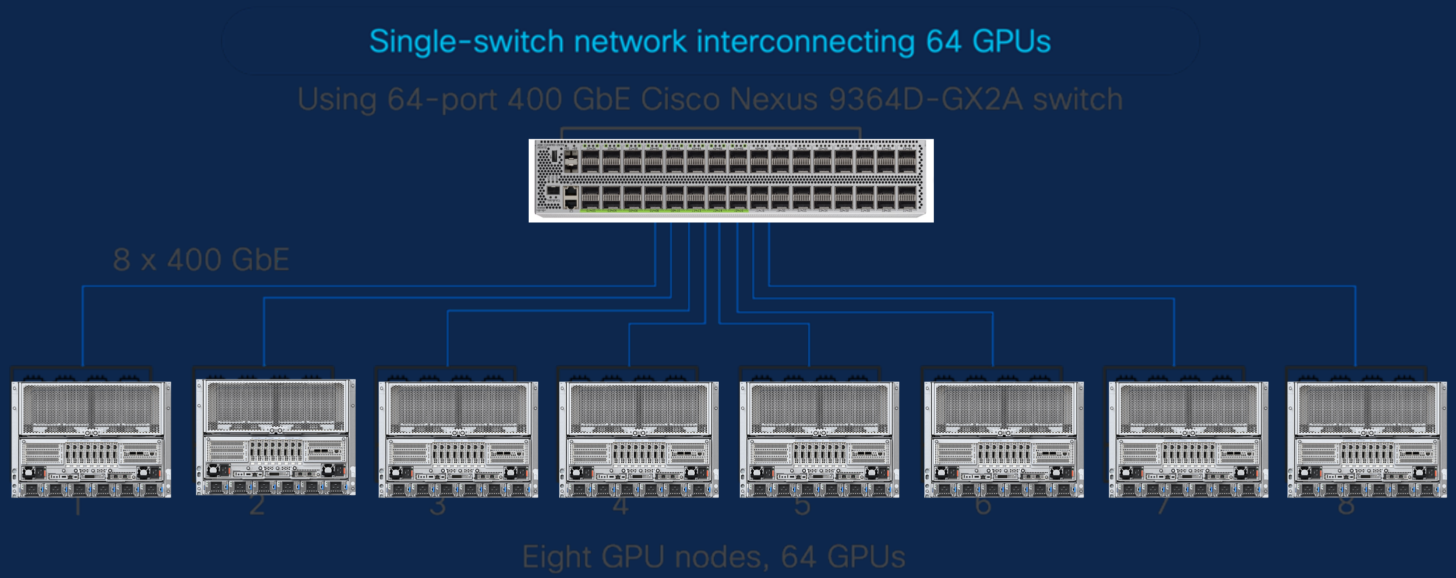


- **Purpose:** Move gradients and model updates *at warp speed* between GPUs during training.
- **Low Latency:** Every microsecond counts — syncs faster than your cat on a laser pointer.
- **High Bandwidth:** 400G links keep data flowing non-stop, no traffic jams allowed.
- **Dedicated Fabric:** NVLink, NVSwitch, and RoCE — the “express lanes” for GPU-to-GPU chatter.

New Data Center design requirements

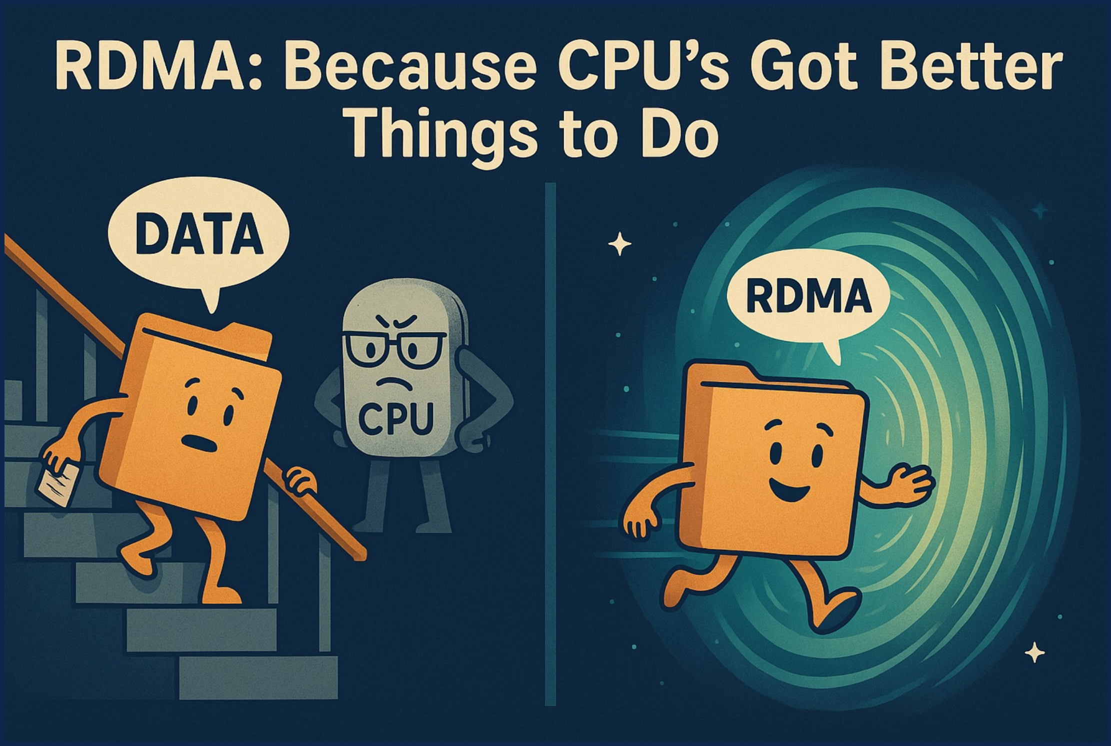


Designing a Smaller Inter-GPU Backend Network



Smaller GPU clusters can use a single-switch network. For example, up to 64 GPUs can be interconnected using the 2 RU, 64-port 400 GbE, Cisco Nexus 9364D-GX2A switch.

Zero Copy Networking

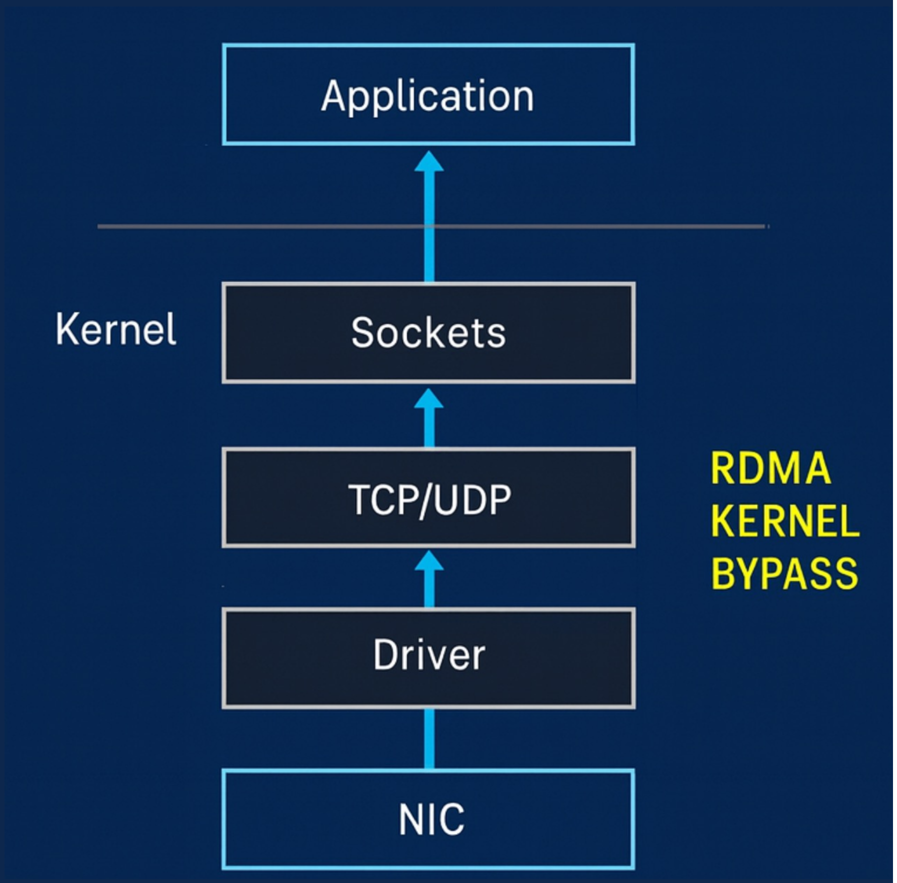


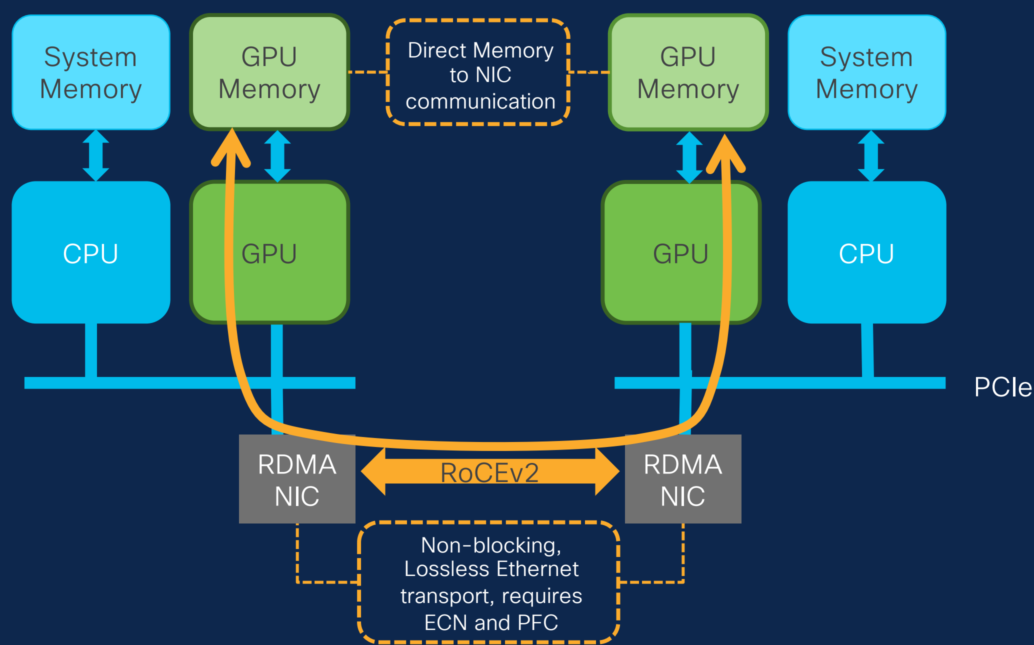
Traditional Data Transfer

- Application to Kernel Space: Data is copied from the application's memory space to the kernel's memory space.
- Kernel Space to Network Interface Card (NIC): Data is copied from the kernel's memory space to the NIC's memory.
- NIC to Network: Data is transmitted over the network.
- Network to NIC: Data is received by the destination NIC.
- NIC to Kernel Space: Data is copied from the NIC's memory to the kernel's memory.
- Kernel Space to Application: Data is copied from the kernel's memory to the application's memory.

How Does RDMA Work

- Direct Memory Access: Bypassing the operating system, RDMA directly moves data between network and application memory.
- Zero-Copy Networking: Minimizes redundant data copies to improve efficiency and throughput.
- Asynchronous Operations: CPU is freed from overseeing data transfer, the CPU can tackle other tasks.



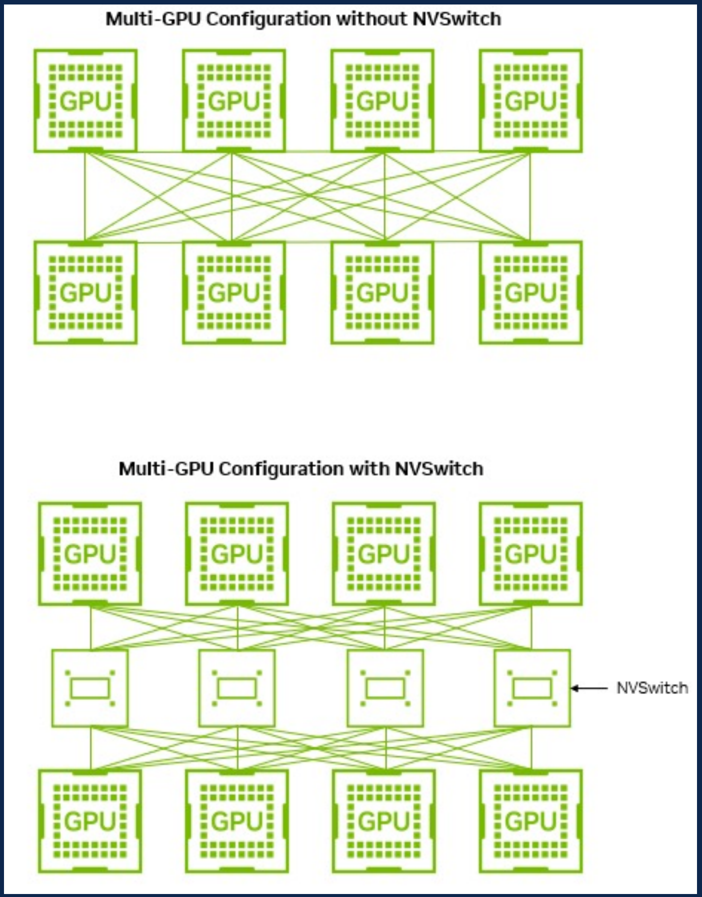


System Memory
GPU Memory
Direct Memory to NIC communication
GPU Memory
System Memory
CPU
GPU
GPU
CPU
PCIe
RDMA NIC
RoCEv2
RDMA NIC
Non-blocking, Lossless Ethernet transport, requires ECN and PFC

NVLink / NVSwitch

NVLink: A high-bandwidth, low-latency interconnect technology designed to directly connect NVIDIA GPUs, enabling them to share memory and data at very high speeds, effectively acting as a single large GPU

NVSwitch: A dedicated switch chip that facilitates communication between multiple GPUs connected via NVLink, acting as a central hub to manage data flow and prevent bottlenecks when numerous GPUs are interacting with each other.



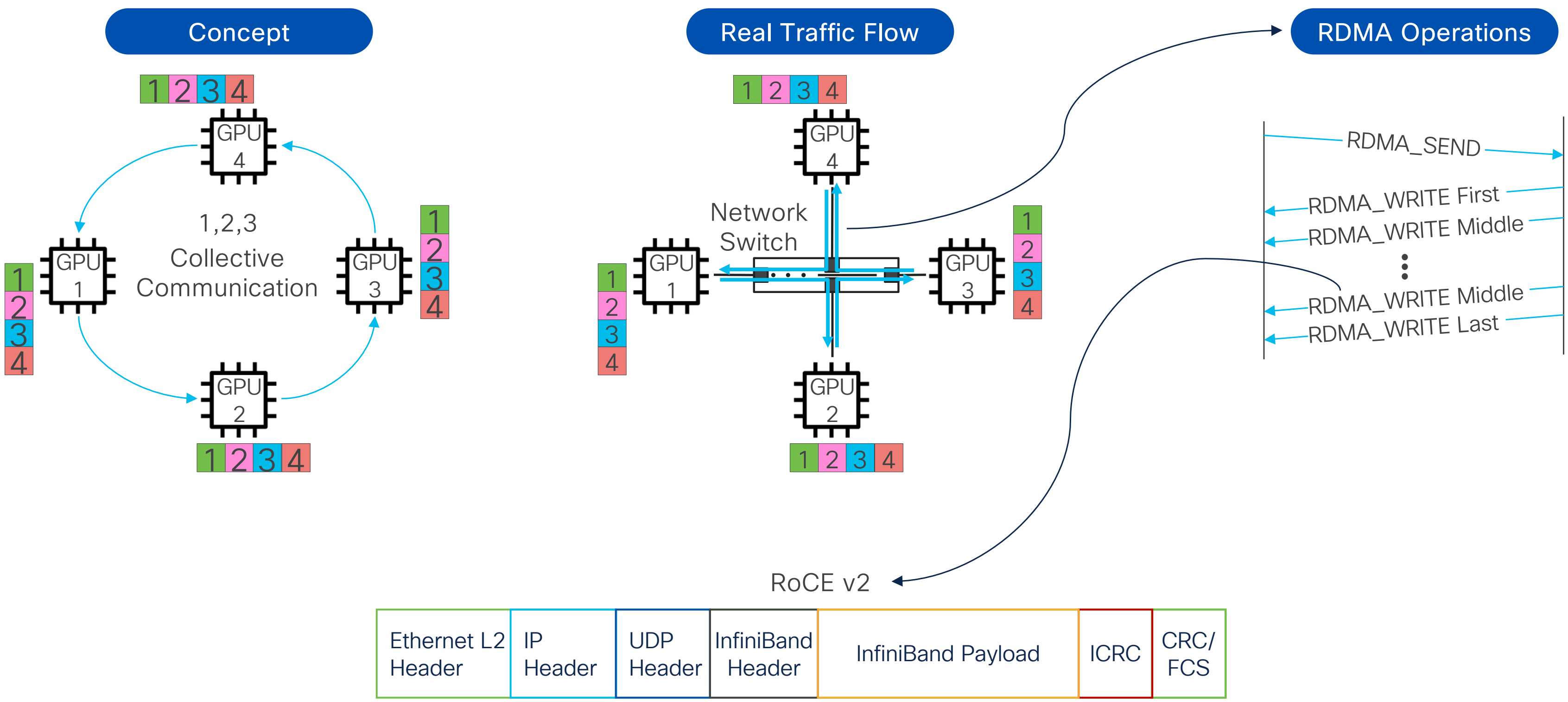
RoCE v1 vs. RoCE v2: Evolution of RDMA over Ethernet

Feature	RoCE v1 (RoCE 1)	RoCE v2 (RoCE 2)
Layer	Layer 2 (Ethernet)	Layer 3 (IP/UDP)
Routing	Cannot be routed; limited to a local Layer 2 network	Can be routed across Layer 3 networks
Encapsulation	Ethernet frames	UDP over IP
Congestion Management	PFC (Priority Flow Control)	ECN (Explicit Congestion Notification) +-PFC
Use Case	Local clusters; single subnet	Larger, routed networks across subnets
Scalability	Limited to a broadcast domain (Layer 2 segment)	Scaiable across multiple subnets, data centers

What does that RDMA packet look like on the wire.

Ethernet Frame
→ **QoS Tagging** (CoS=3, DSCP=26)
→ IPv4 (ECN bits = 00)
→ UDP (Dst Port=4791)
→ Infiniband BTH (Opcode=RDMA_WRITE)
→ RETH (Remote address + length)
→ Payload (Tensor data)
→ ICRC (Checksum)

Collective Communication via RDMA Operations



Distributed training results in UDP/IP Traffic on the inter-GPU network

Silicon One

Cisco 8800/8600/8200 Series (w/ deep buffers) running IOS-XR

19.2 Tbps

12.8 Tbps

10.8 Tbps

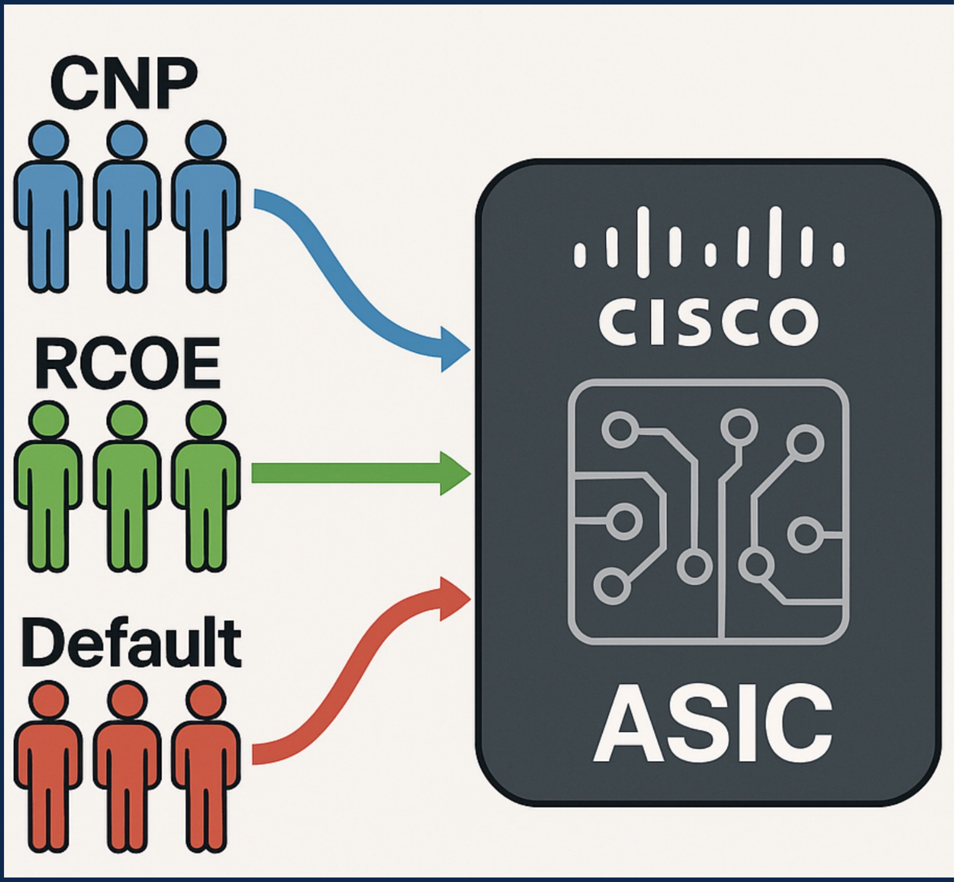
Cisco 8100 Series (w/o deep buffers) running SONiC

51.2 Tbps

25.6 Tbps

12.8 Tbps

Buffers



The Costco Conundrum

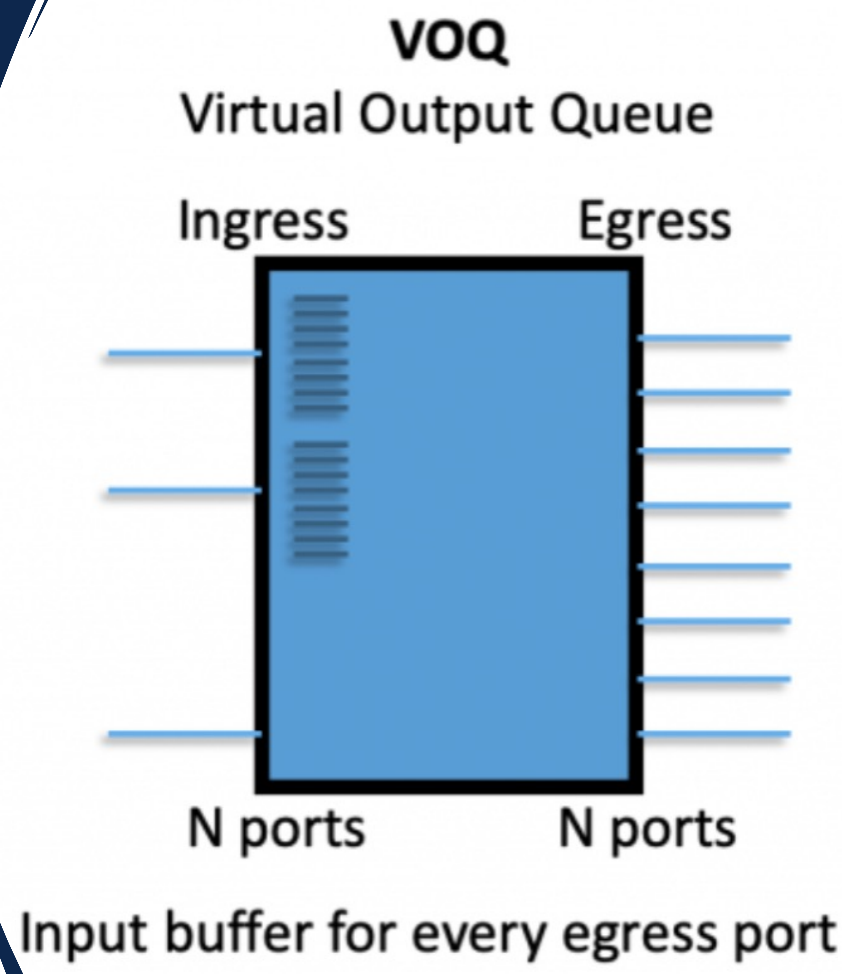
Elephant flows and mice flows

Control data

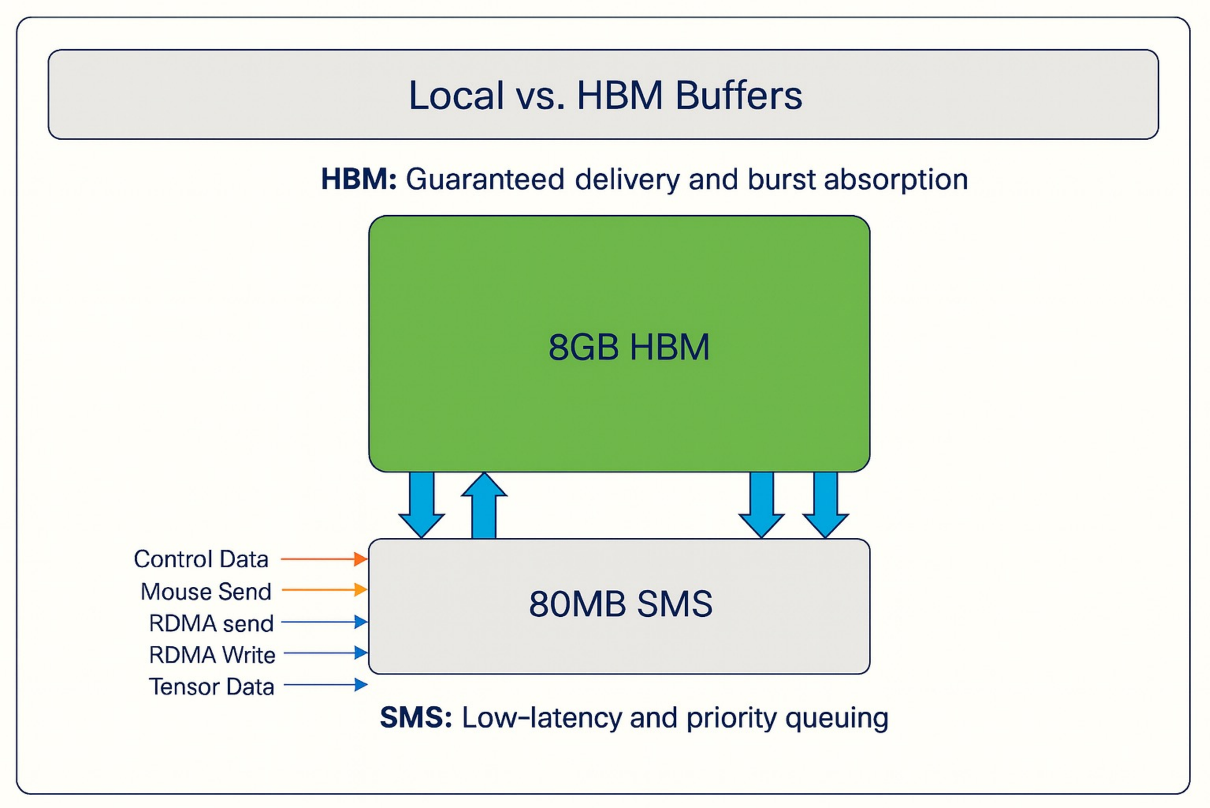
RDMA

Virtual output queues

- Each ingress port has multiple VOQs, one for each egress port.
- Egress ports issue credits to VOQs, allowing the ingress port to send packets only when the egress port has available buffer space.



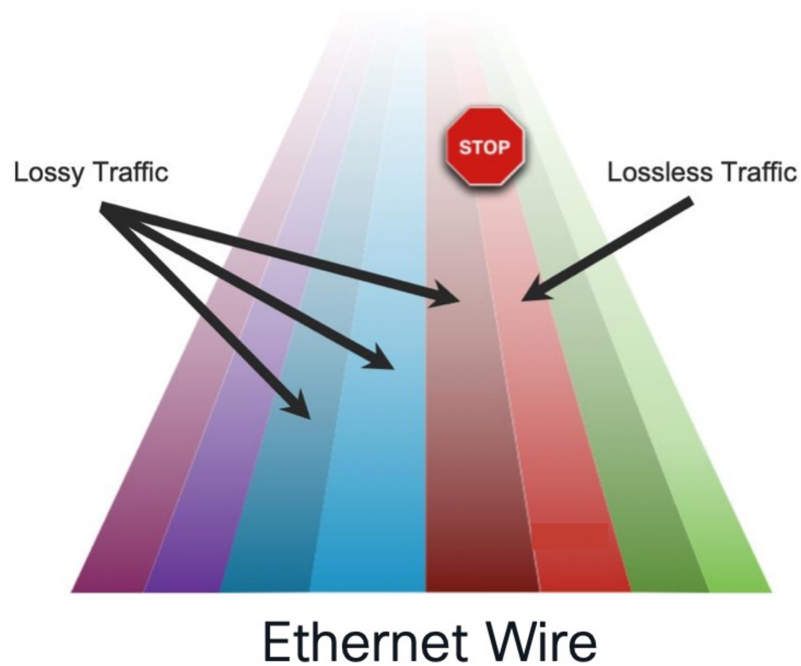
Buffering system
On-Chip vs. Off-Chip Buffers in QoS



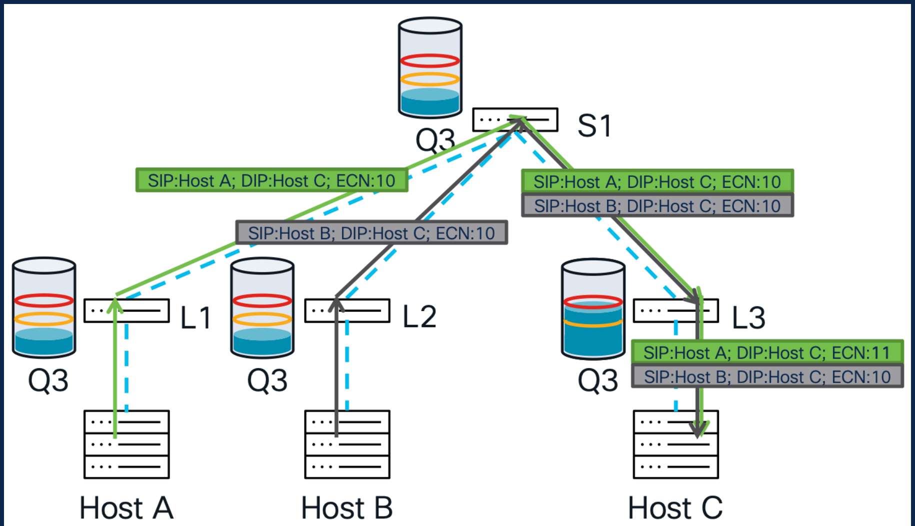
Priority Flow Control

- Lossless Ethernet
- Flow control on a per priority basis
- Supports Per-Priority-Pause
- Other traffic can rely on upper layers for retransmissions.

```
configure terminal
interface ethernet1/1
  priority-flow-control mode auto
```



End to End ECN “Explicit Congestion Notification”



ECN	ECN Behavior
00	Non ECN Capable
10	ECN Capable Transport (0)
01	ECN Capable Transport (1)
11	Congestion Encountered

Nexus QoS Profiles

```
system qos qos-profile ?
  ai-ml      # AI/ML workloads (large buffers)
  default    # General data center (balanced)
  storage    # Storage traffic (lossless, low-latency)
  dcn        # DC networking (traditional mix)
  custom     # Custom user-defined (if supported)
```

1. ai-ml – Optimized for AI/ML infrastructure, with larger buffers and queue depths for GPU burst traffic.
2. default – Balanced, general-purpose for most data center workloads.
3. storage – Designed for low-latency, lossless storage transports like NVMe, FCoE, or iSCSI.
4. dcn – For traditional data center networking environments, balancing throughput across mixed workloads.

Turning on the AI-ML Template

system qos
qos-profile ai-ml

QoS Profile: ai-ml
Active Queues: 8 (AI/ML mode)
Dynamic Buffer Sharing: Enabled

It changes *how the ASIC behaves internally*:
- ECN/PFC thresholds
- buffer carve-outs
- DPP/ETRAP logic
- dynamic buffer sharing
- watchdog timers

Default vs AI-ML QoS Profile – What Changes in the ASIC

Behavior	Default	AI-ML Profile
ECN thresholds	Static (wide range)	**Tuned per queue (≈150–3000 KB typical)**
PFC thresholds	Same for all classes	**Tuned per ASIC + MTU (tight for RoCE)**
Buffer model	Static per-queue	**Dynamic buffer sharing (elastic)**
PFC watchdog	Off	**Enabled (≈8 µs detection)**
DPP / ETRAP	Off	**Enabled by default**
ECN behavior	Fixed ramp	**Adaptive ramp (ASIC measures utilization)**

AI/ML fabric

```
class-map type qos match-any ROCEv2
  match dscp 26
class-map type qos match-any CNP
  match dscp 48

policy-map type qos QOS_CLASSIFICATION
  class ROCEv2
    set qos-group 3
  class CNP
    set qos-group 7
  class class-default
    set qos-group 0

policy-map type network-qos qos_network
  class type network-qos c-8q-nq3
    mtu 4500
    pause pfc-cos 3
  class type network-qos c-8q-nq-default
    mtu 9216

policy-map type queuing AI-ML-QUEUING
  class type queuing c-out-8q-q3
    bandwidth remaining percent 90
    random-detect minimum-threshold 150 kbytes maximum-threshold 3000 kbytes drop-probability 7 weight 0 ecn
  class type queuing c-out-8q-q7
    priority level 1
    bandwidth remaining percent 1
  class type queuing c-out-8q-q0
   bandwidth remaining percent 9

system qos
  qos-profile ai-ml
  service-policy type qos input QOS_CLASSIFICATION
  service-policy type network-qos qos_network
  service-policy type queuing output AI-ML-QUEUING
```

How to build a lossless fabric.

- **Classification** (type qos) - Identifying the important traffic
- **Queuing & ECN** (type queuing) - Managing bandwidth and congestion signaling
- **Flow Control** (type network-qos) - Pausing traffic before buffers overflow

Identify the type of traffic.

```
class-map type qos match-any ROCEv2
  match dscp 26
class-map type qos match-any CNP
  match dscp 48
```

Taking that traffic I just matched and assign it to a queue.

```
policy-map type qos QOS_CLASSIFICATION
  class ROCEv2
    set qos-group 3
  class CNP
    set qos-group 7
  class class-default
    set qos-group 0
```

Mapping (What Each Group Means)

qos-group 3 (RoCEv2 data, DSCP 26)
- Lossless class: PFC on, MTU 4200, ECN/WRED enabled
- Goal: no drops for GPU all-reduce; ECN prevents buffer blow-ups

qos-group 7 (CNP, DSCP 48)
- Strict-priority at egress; no PFC, MTU 9216
- Goal: CNP feedback always cuts through—never paused or delayed

qos-group 0 (Default)
- Best-effort, MTU 9216, no PFC
- Goal: Isolation from GPU traffic; drops are acceptable

Key CLI commands for classification

Ingress Classification (type qos)

```
class-map type qos match-any ROCEv2
  match dscp 26
class-map type qos match-any CNP
  match dscp 48

policy-map type qos QOS_CLASSIFICATION
  class ROCEv2
    set qos-group 3
  class CNP
    set qos-group 7
  class class-default
    set qos-group 0

system qos
  service-policy type qos input QOS_CLASSIFICATION
```

Transport Traits (Lossless vs Not)

Network Behavior (type network-qos)

```
policy-map type network-qos qos_network
  class type network-qos c-8q-nq3
    mtu 4200
    pause pfc-cos 3
  class type network-qos c-8q-nq-default
    mtu 9216

system qos
  service-policy type network-qos qos_network
```

Scheduling & Congestion Signals

Egress Queuing (type queuing)

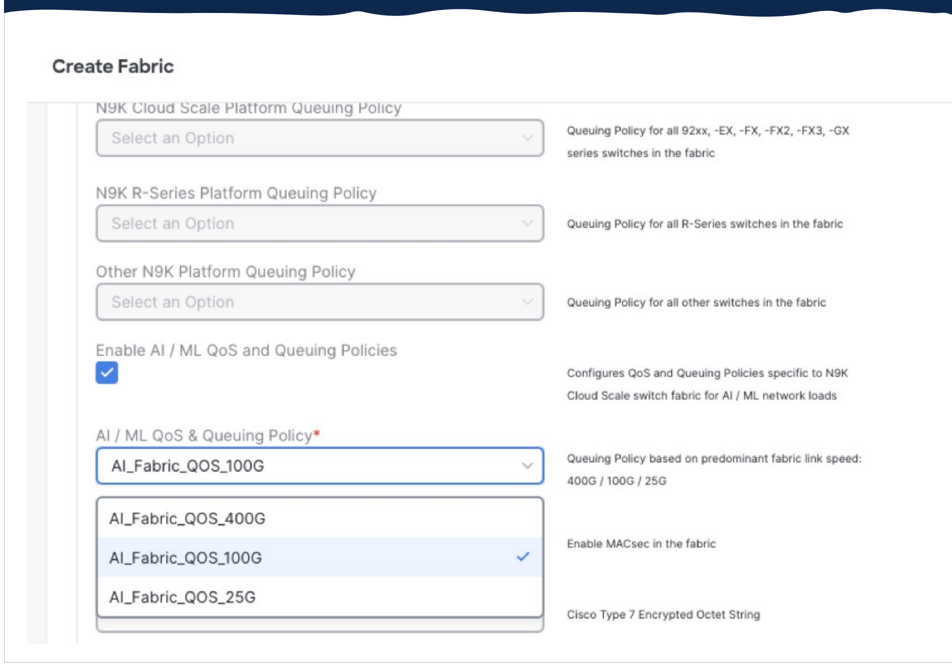
```
policy-map type queuing AI-ML-QUEUING
  class type queuing c-out-q3
    bandwidth percent 90
    random-detect ecn
    random-detect minimum-threshold 3000 kbytes
    random-detect maximum-threshold 6000 kbytes
  class type queuing c-out-q7
    priority level 1
   bandwidth percent 1
  class type queuing c-out-q0
    bandwidth percent 9

system qos
  service-policy type queuing output AI-ML-QUEUING
```

Service Policy commands attach the policy maps

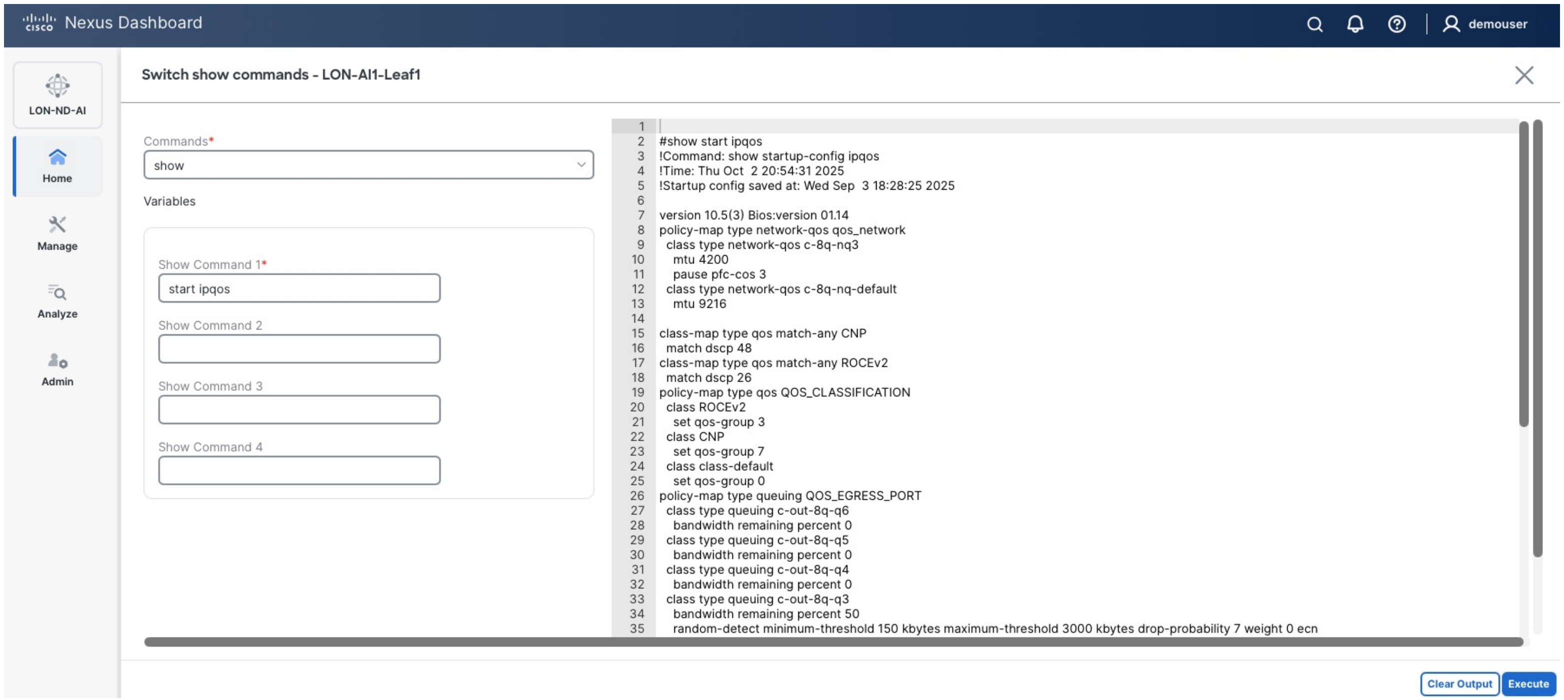
```
system qos
  qos-profile ai-ml
  service-policy type qos input QOS_CLASSIFICATION
  service-policy type network-qos qos_network
  service-policy type queuing AI-ML-QUEUING
```

Config drift: because who doesn't love an after-hours ghost hunt?

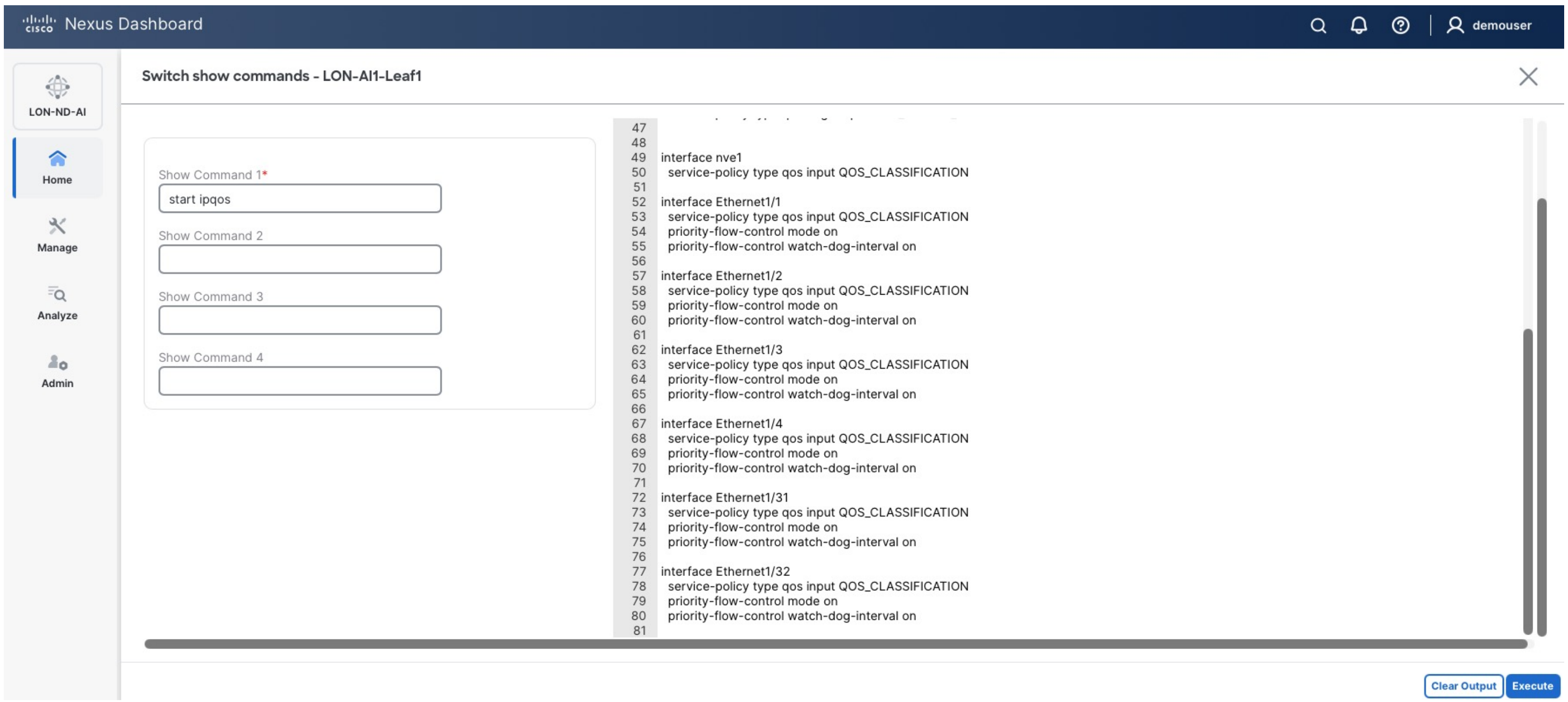
If you’ve set up QoS, you know!!!



NDFC QOS Configuration.



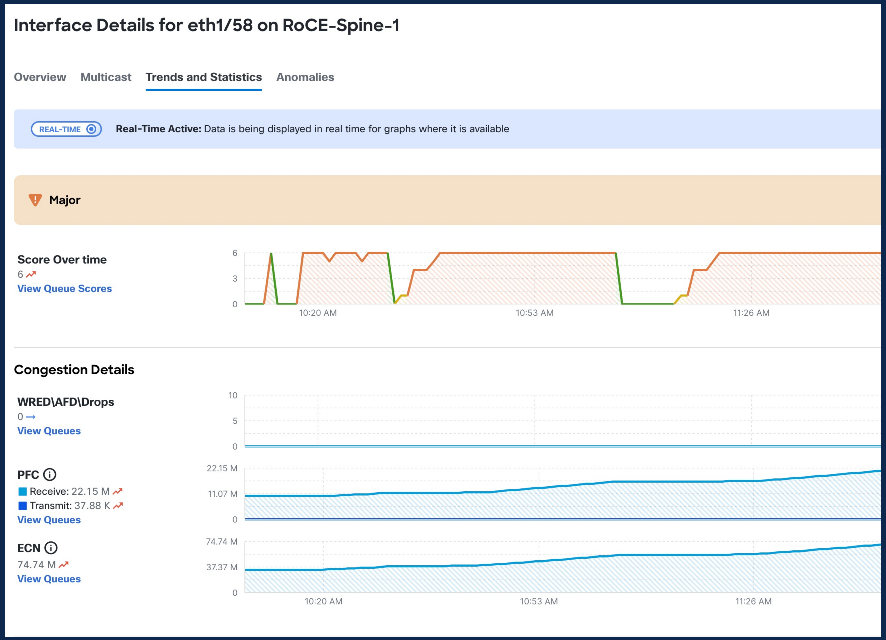
NDFC QOS Configuration.

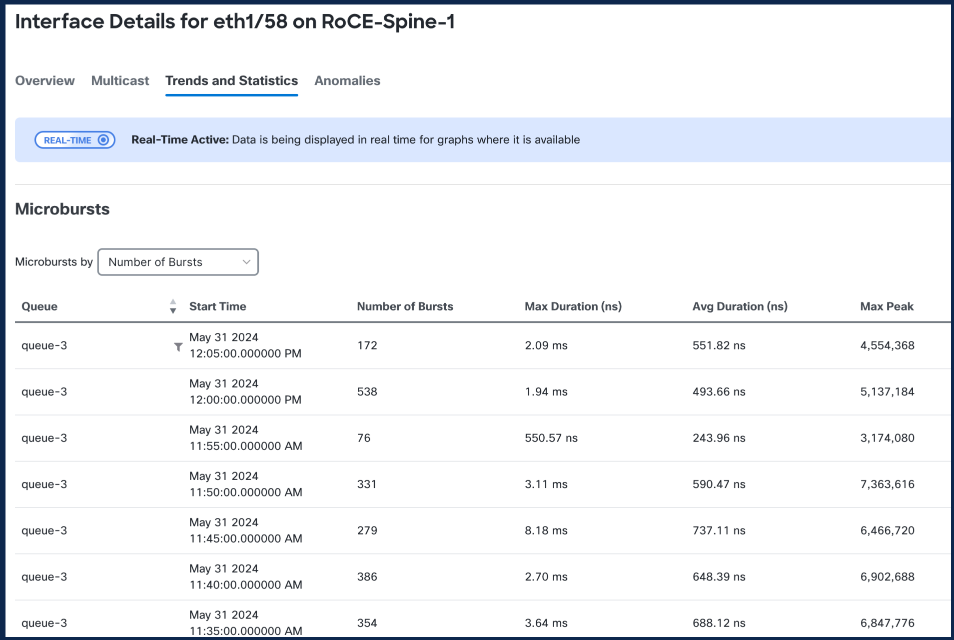



Telemetry. Telemetry.
Telemetry

Visibility matters

Real-time performance metrics





Interface Details for eth1/58 on RoCE-Spine-1

Overview Multicast Trends and Statistics Anomalies

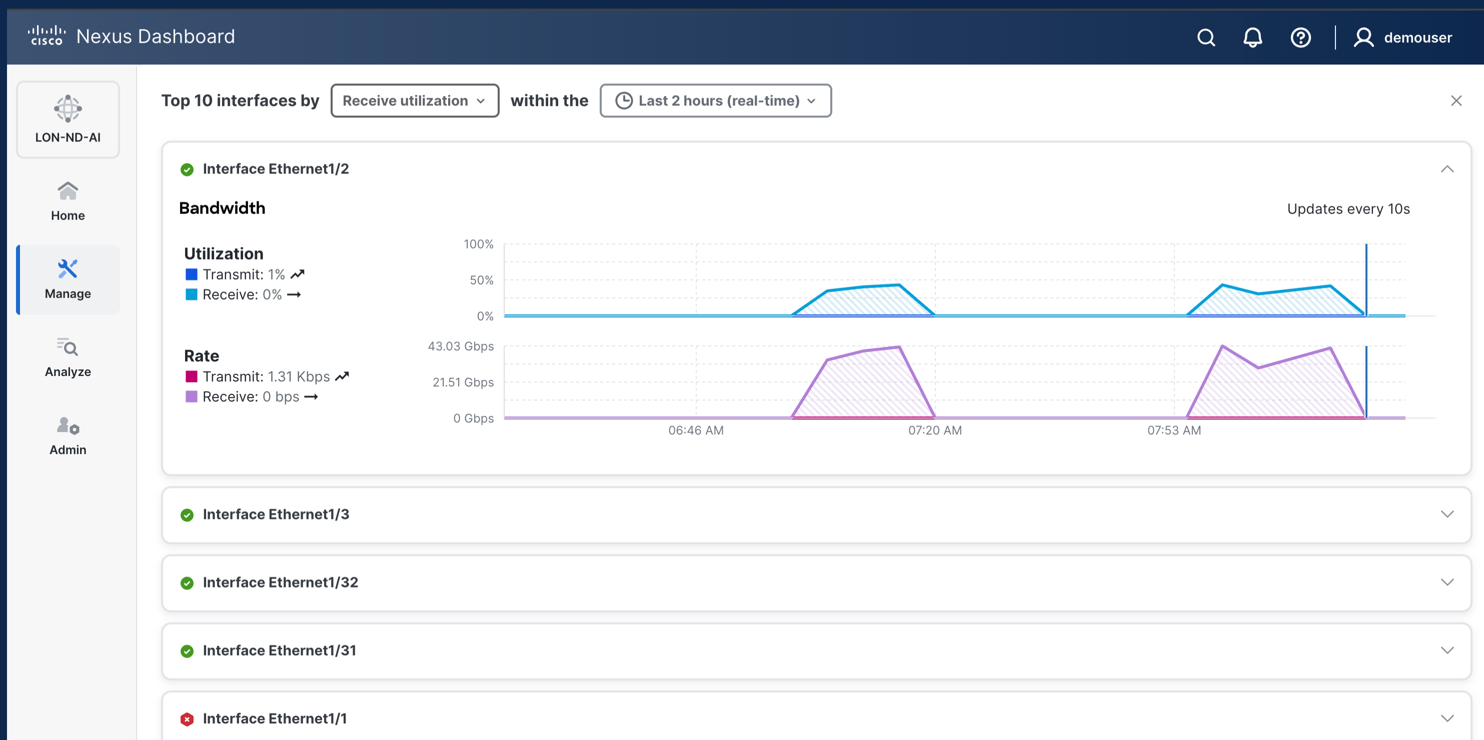
REAL-TIME **Real-Time Active:** Data is being displayed in real time for graphs where it is available

Microbursts

Microbursts by: Number of Bursts

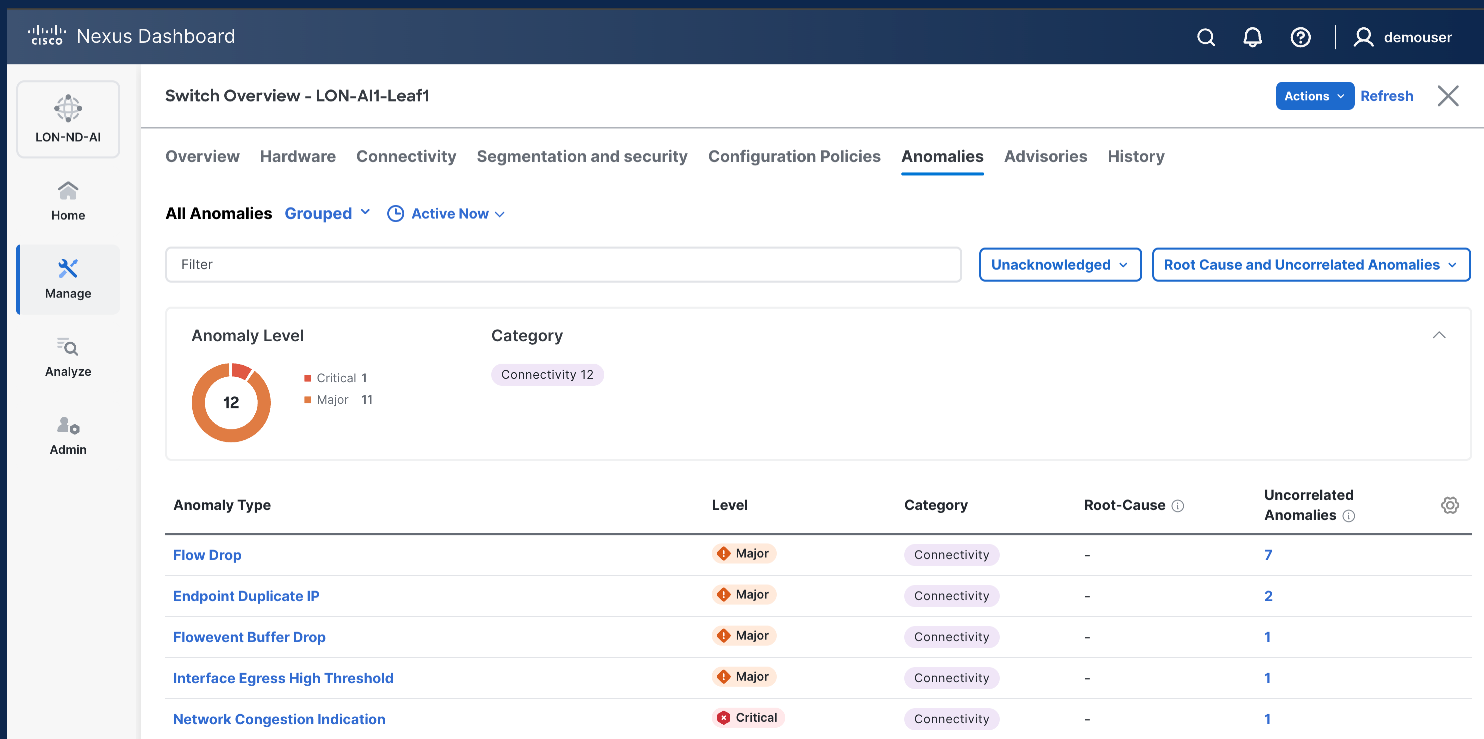
Queue	Start Time	Number of Bursts	Max Duration (ns)	Avg Duration (ns)	Max Peak
queue-3	May 31 2024 12:05:00.000000 PM	172	2.09 ms	551.82 ns	4,554,368
queue-3	May 31 2024 12:00:00.000000 PM	538	1.94 ms	493.66 ns	5,137,184
queue-3	May 31 2024 11:55:00.000000 AM	76	550.57 ns	243.96 ns	3,174,080
queue-3	May 31 2024 11:50:00.000000 AM	331	3.11 ms	590.47 ns	7,363,616
queue-3	May 31 2024 11:45:00.000000 AM	279	8.18 ms	737.11 ns	6,466,720
queue-3	May 31 2024 11:40:00.000000 AM	386	2.70 ms	648.39 ns	6,902,688
queue-3	May 31 2024 11:35:00.000000 AM	354	3.64 ms	688.12 ns	6,847,776

Viewing interface statistics.

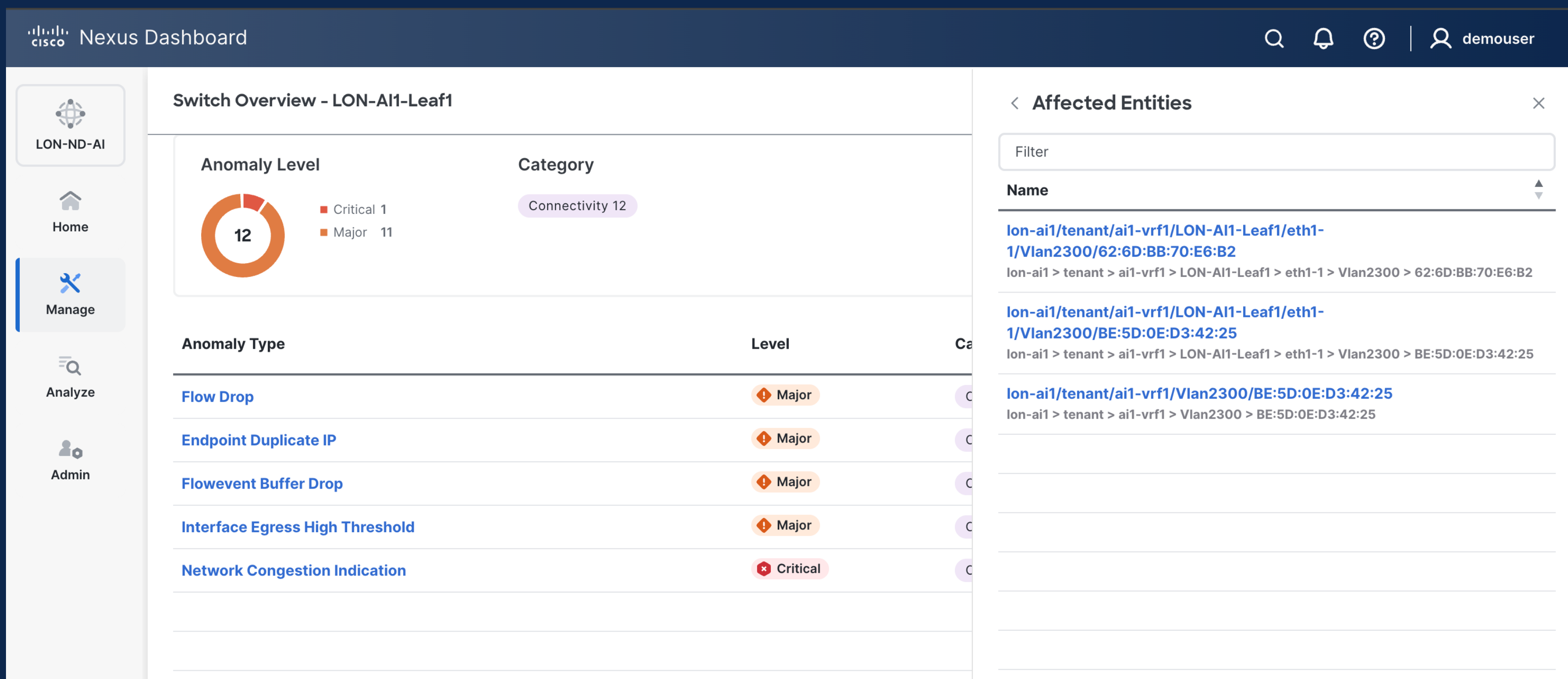


Checking on interface congestion.

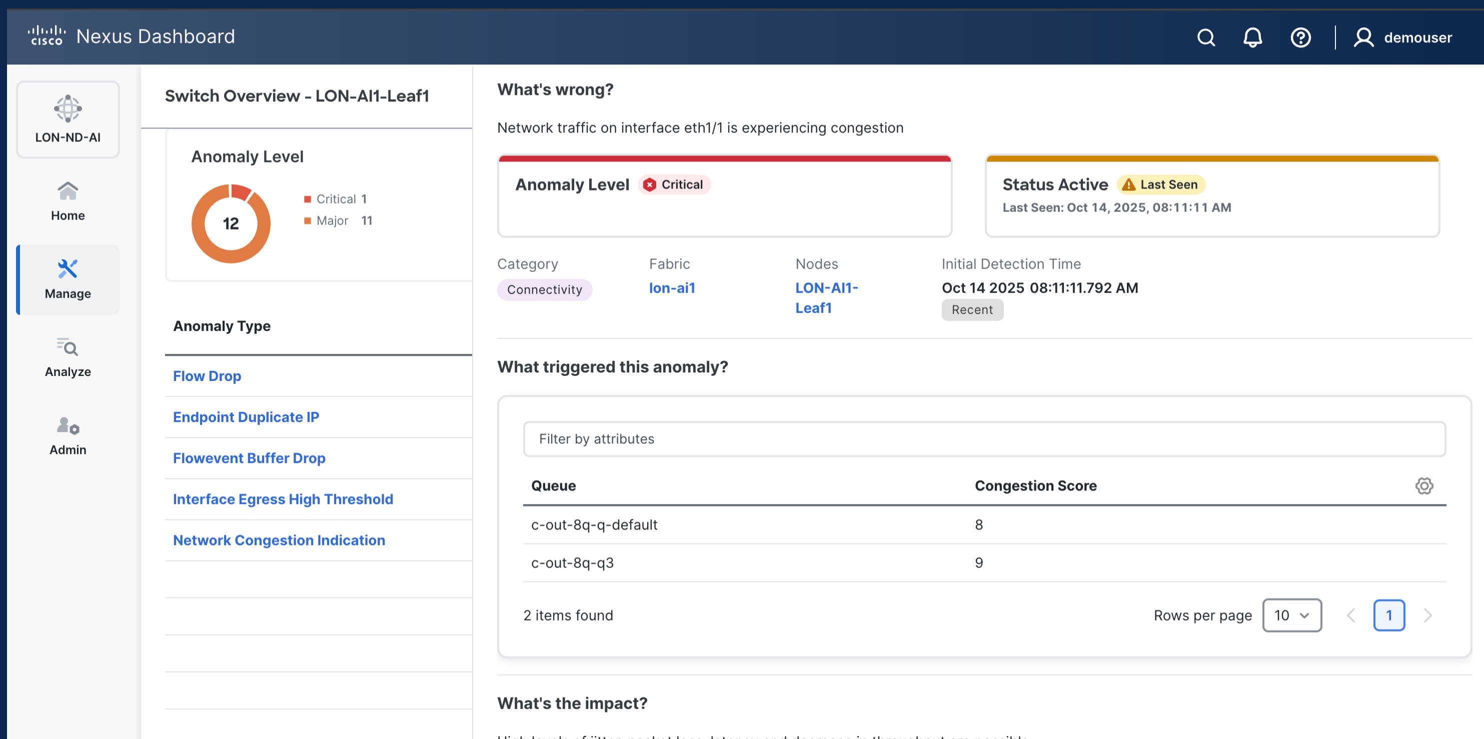
Checking in on Anomalies



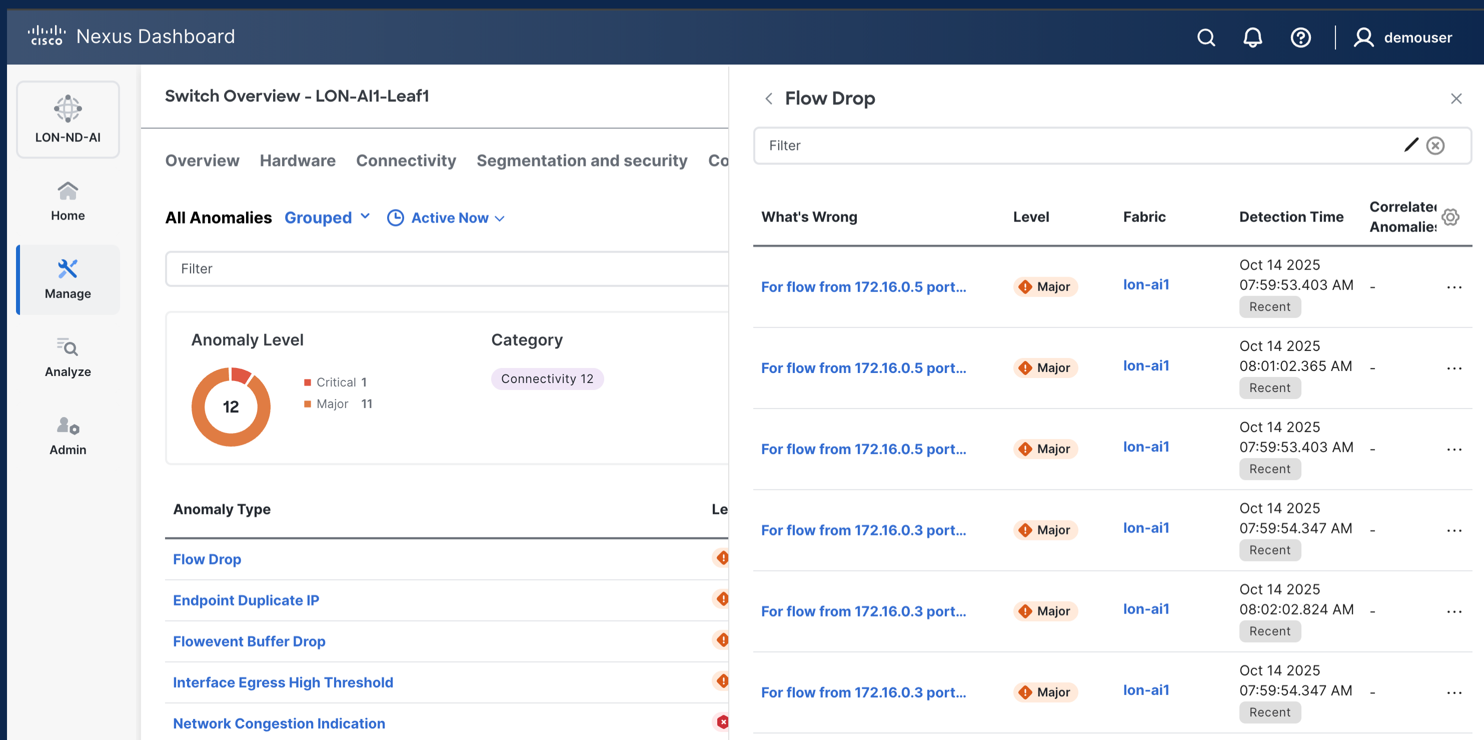
Hosts experiencing congestion



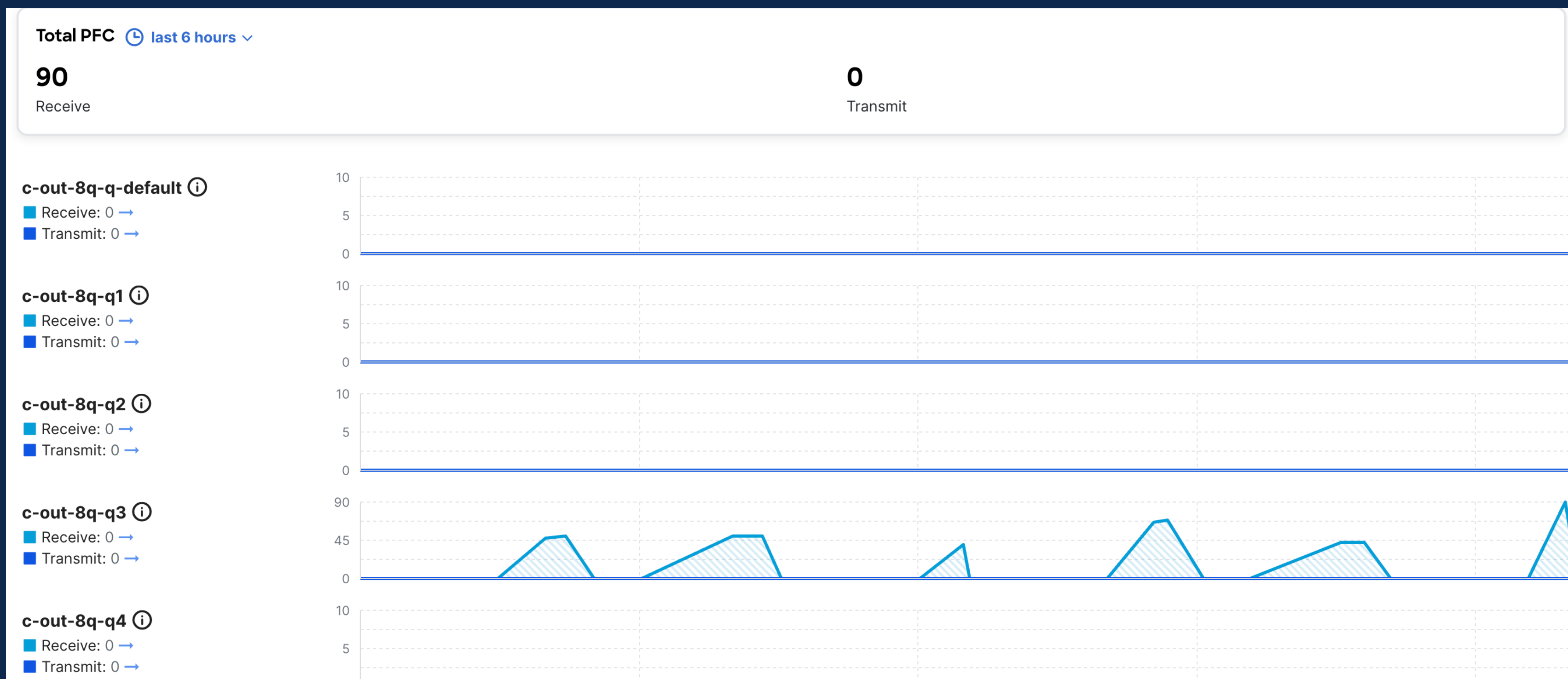
Congestion score.



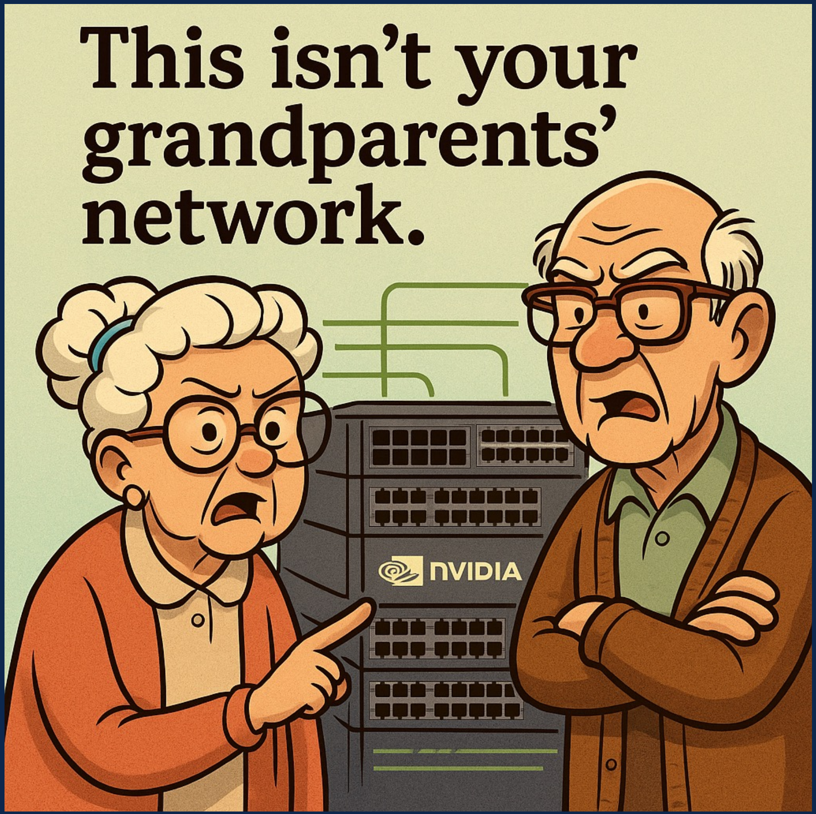
For flow from 172.16.0.5 port 55551 to 172.16.0.4 port 4791, packet drop is detected due to buffer drop, buffer max drop,



Checking interface PFC’s

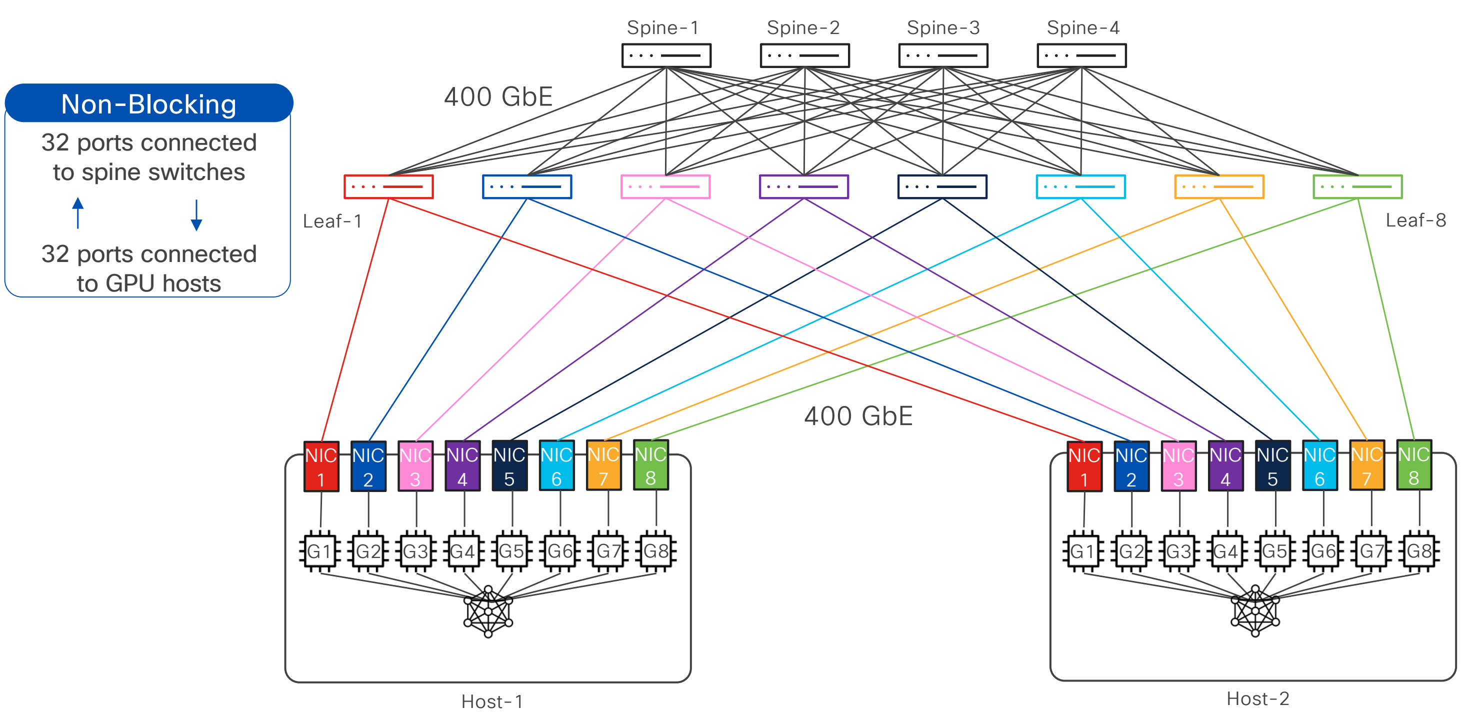


Back in My Day… Networks Had Spanning Tree



Rails-Optimized Design for Inter-GPU Network

Host to Leaf switch connections

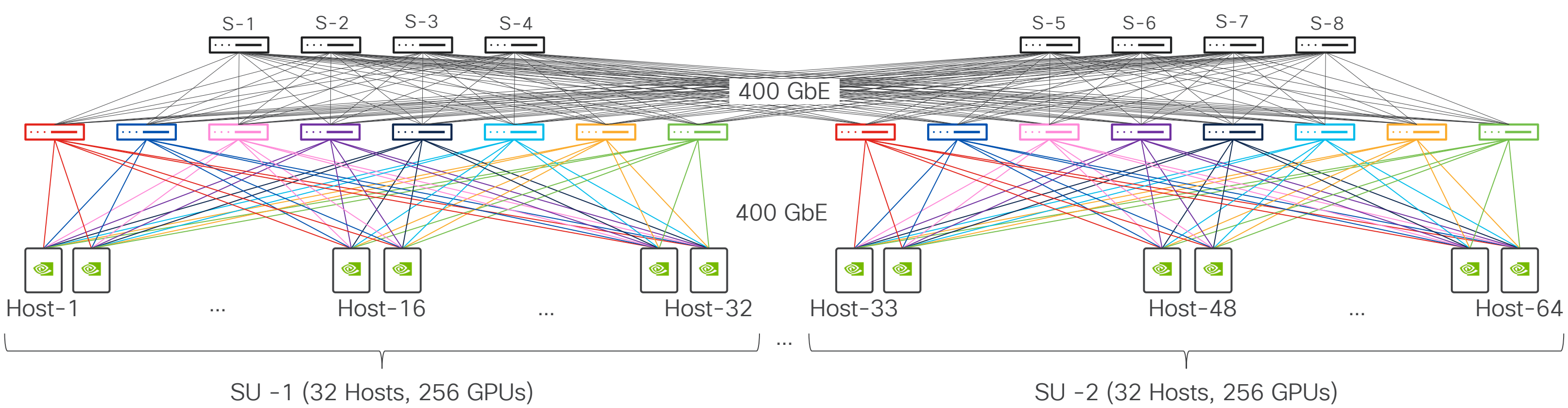


Port 1 on all hosts connects to Leaf-1, Port 2 on all hosts connects to Leaf-2, and so on.

Rails-Optimized Design for Inter-GPU Network

SuperPOD - Up to 1024 GPUs with multiple Scalable Units (SUs) of 256 GPUs

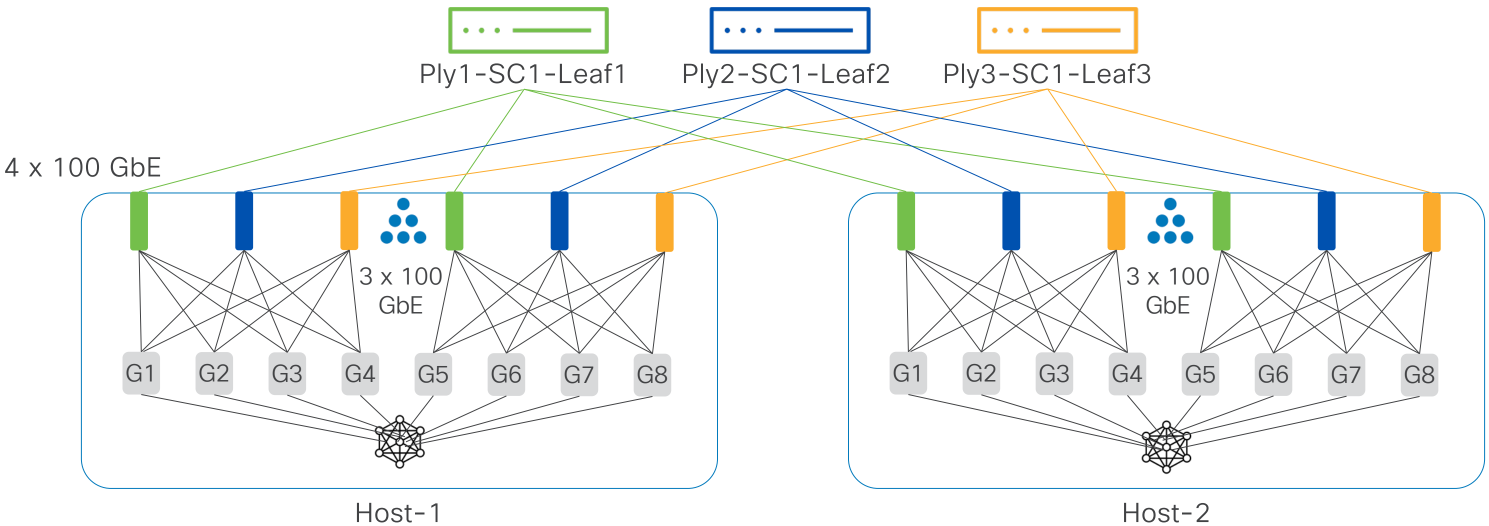
Using Cisco Nexus 9364D-GX2A switch (64 port x 400 GbE)



For larger SuperPOD, add more SUs (each with 256 GPUs) and cables between leaf and spine switches

Intel 3-Ply Design for Inter-Gaudi2 Network

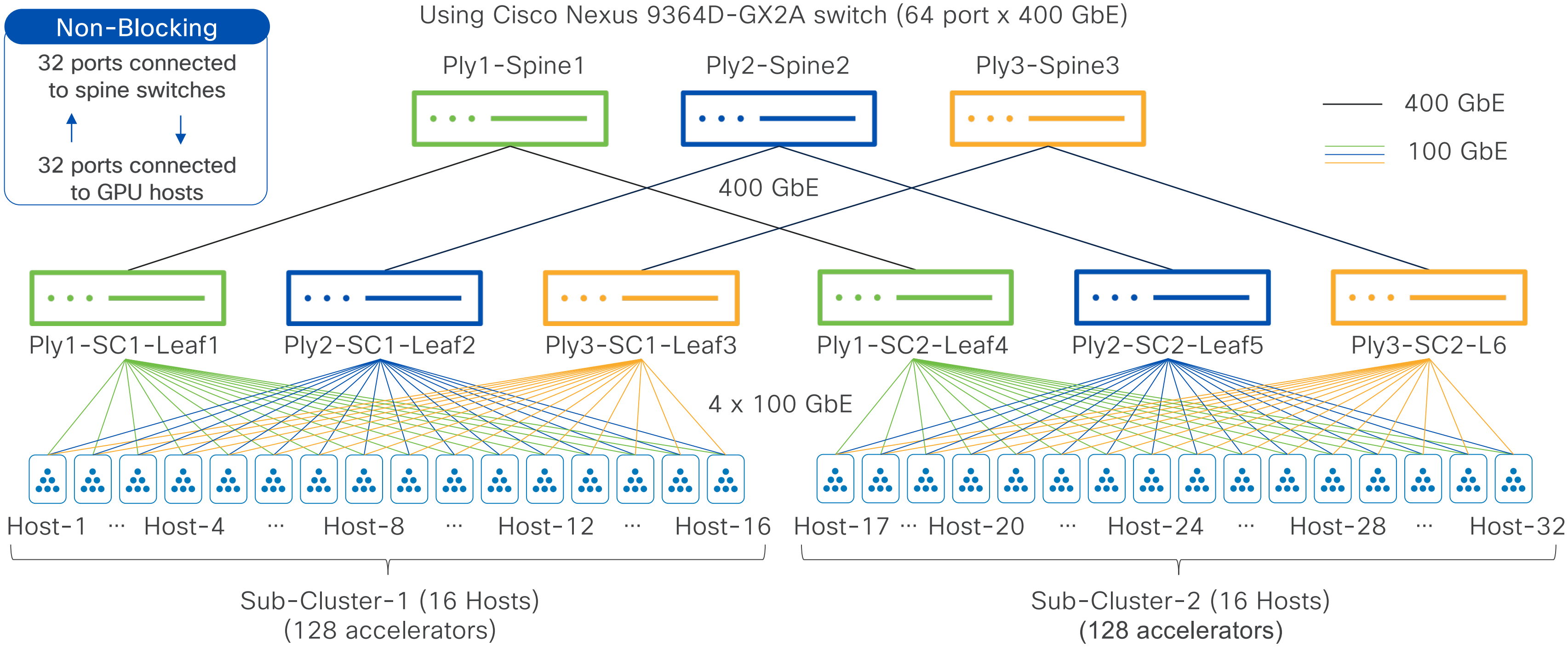
Host to Leaf switch connections



Port 1 and 4 on all hosts connect to Ply-1
Port 2 and 5 on all hosts connect to Ply-2
Port 3 and 6 on all hosts connect to Ply-3

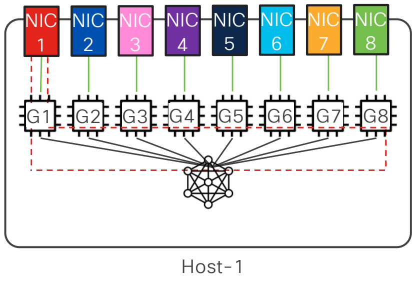
Intel 3-Ply Design for Inter-Gaudi2 Network

Cluster – Multiple Sub-clusters of 128 Gaudi2 accelerators



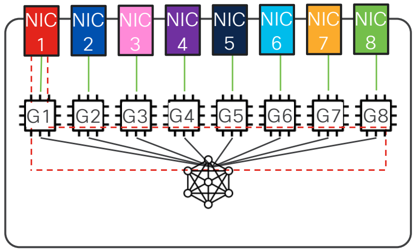
For larger clusters, add more sub-clusters (each with 128 accelerators) and spine switches

Collective Communication between GPU Hosts

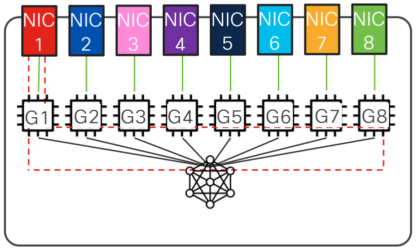


Only a conceptual representation for easier understanding

Collective Communication between GPU Servers



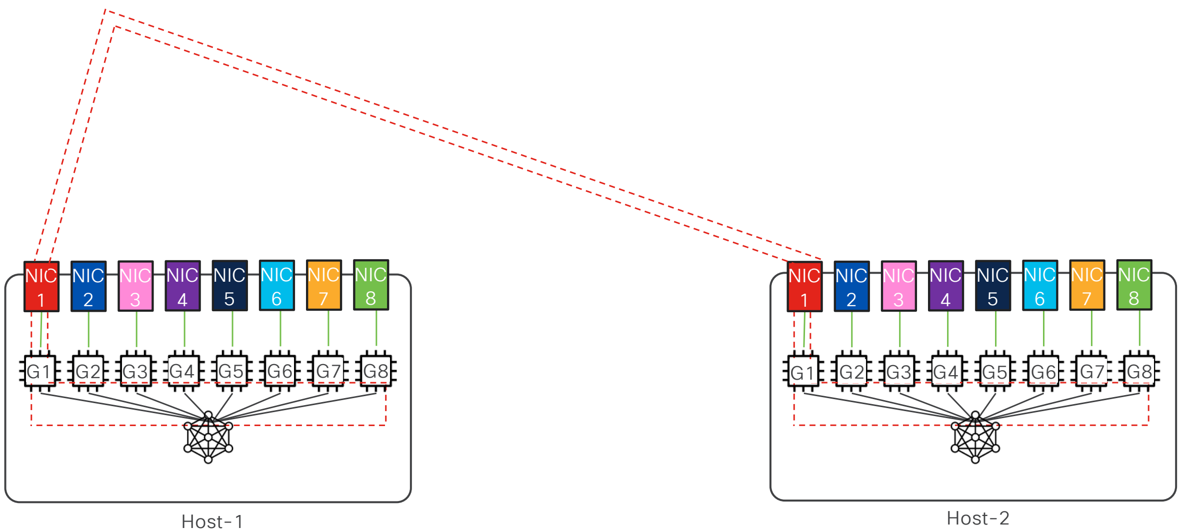
Host-1



Host-2

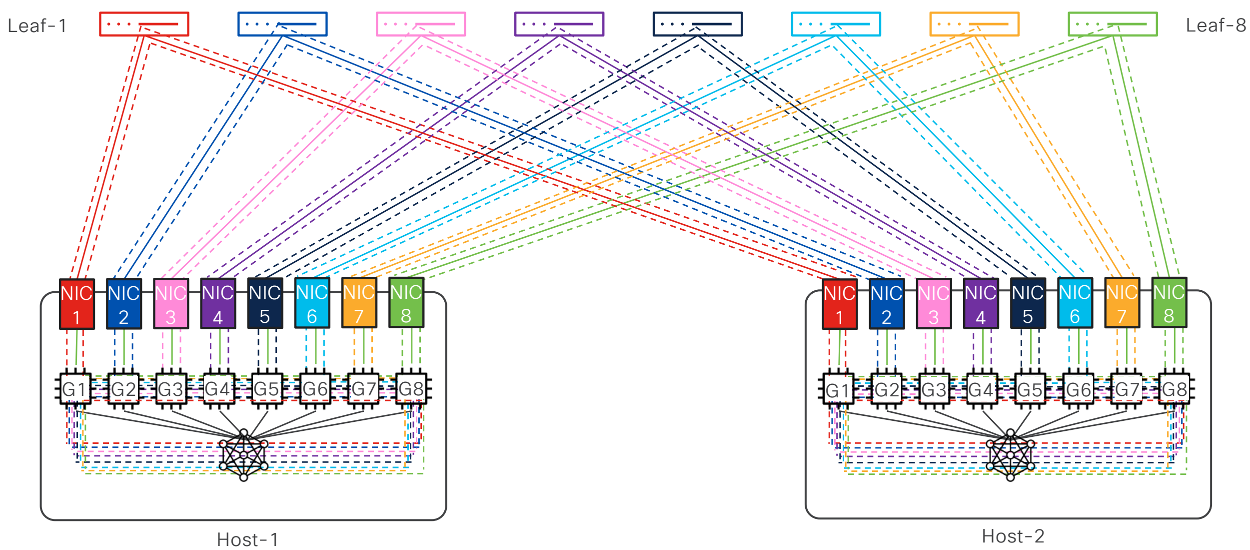
Only a conceptual representation for easier understanding

Collective Communication between GPU Servers



Only a conceptual representation for easier understanding

Collective Communication between GPU Servers

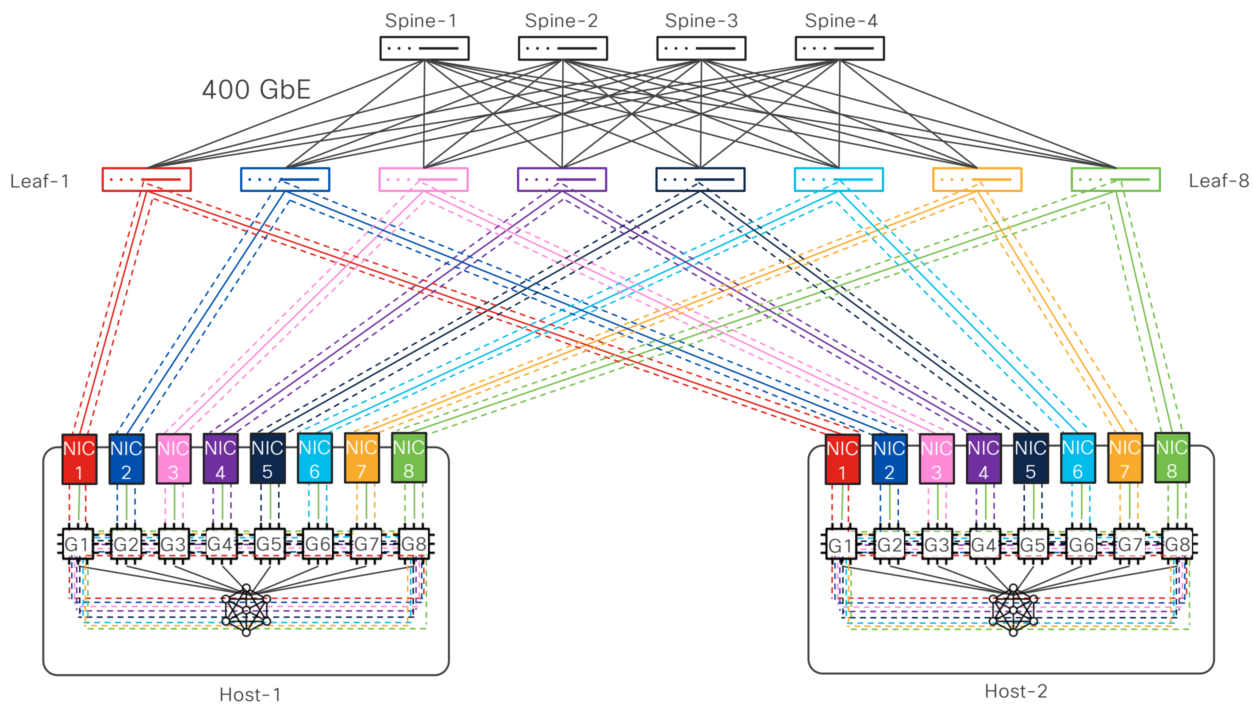


Port 1 on all hosts connects to Leaf-1, Port 2 on all hosts connects to Leaf-2, and so on.

Only a conceptual representation for easier understanding

Rails Design for Inter-GPU Network

Using Cisco Nexus 9364D-GX2A switch (64 port x 400 GbE)



Port 1 on all hosts connects to Leaf-1, Port 2 on all hosts connects to Leaf-2, and so on.

Only a conceptual representation for easier understanding

Questions?

The bridge to possible