

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



Building a High-Performance AI Data Center

Chris Gascoigne, Principal Solutions Engineer
@chriscascoigne@hachyderm.io
BRKDCN-2613

Cisco Webex App

Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until November 15, 2024.



<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKDCN-2613>





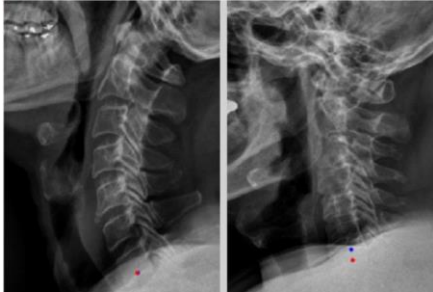
Agenda

- AI/ML Applications
- What Are Their Requirements?
- Building a High-Performance AI DC
- Conclusion

AI is everywhere!



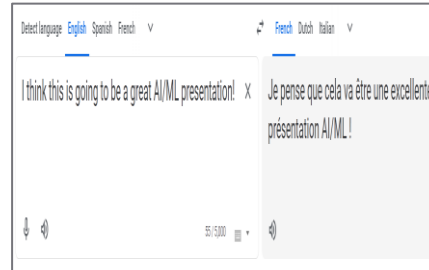
Face recognition



Analysis of medical images



Self-driving vehicles



Machine translation



Conversational agents

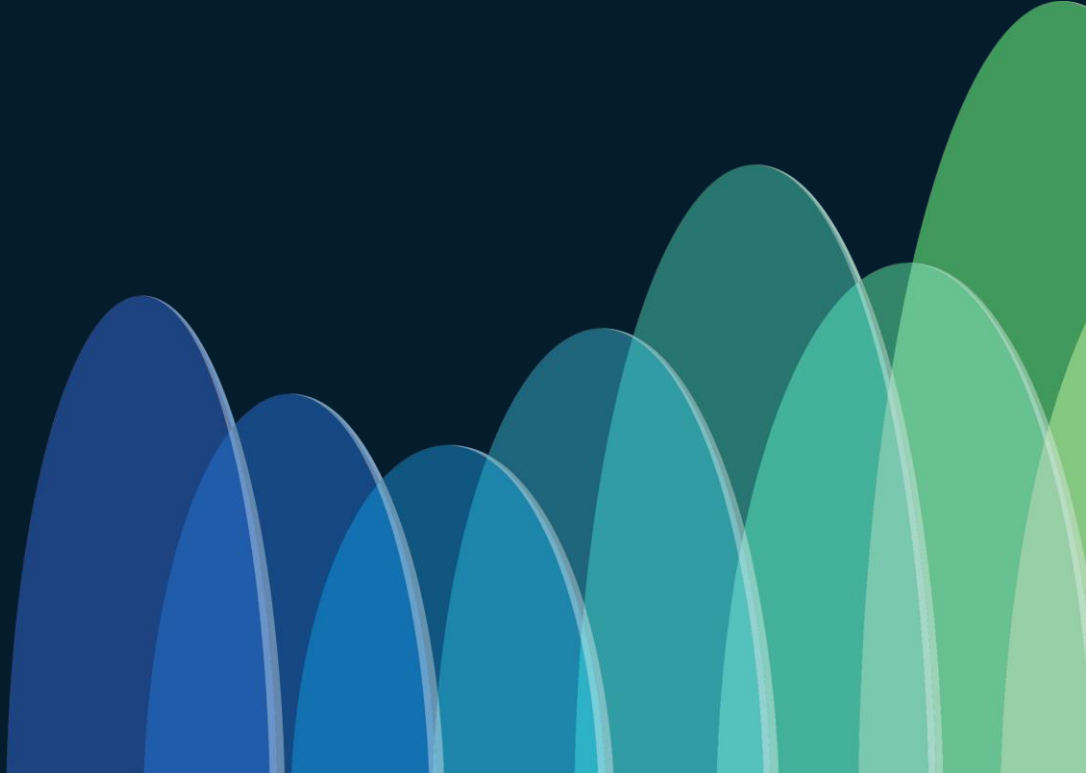


Image generation



Recommender systems

Let's Talk about AI Infrastructure





FULL DATA CENTER WITH 32,000 GPUs

AI FACTORY FOR THE NEW INDUSTRIAL REVOLUTION

645 exaFLOPS of AI performance

13PB of fast memory

58PB/s of aggregate NVLink bandwidth

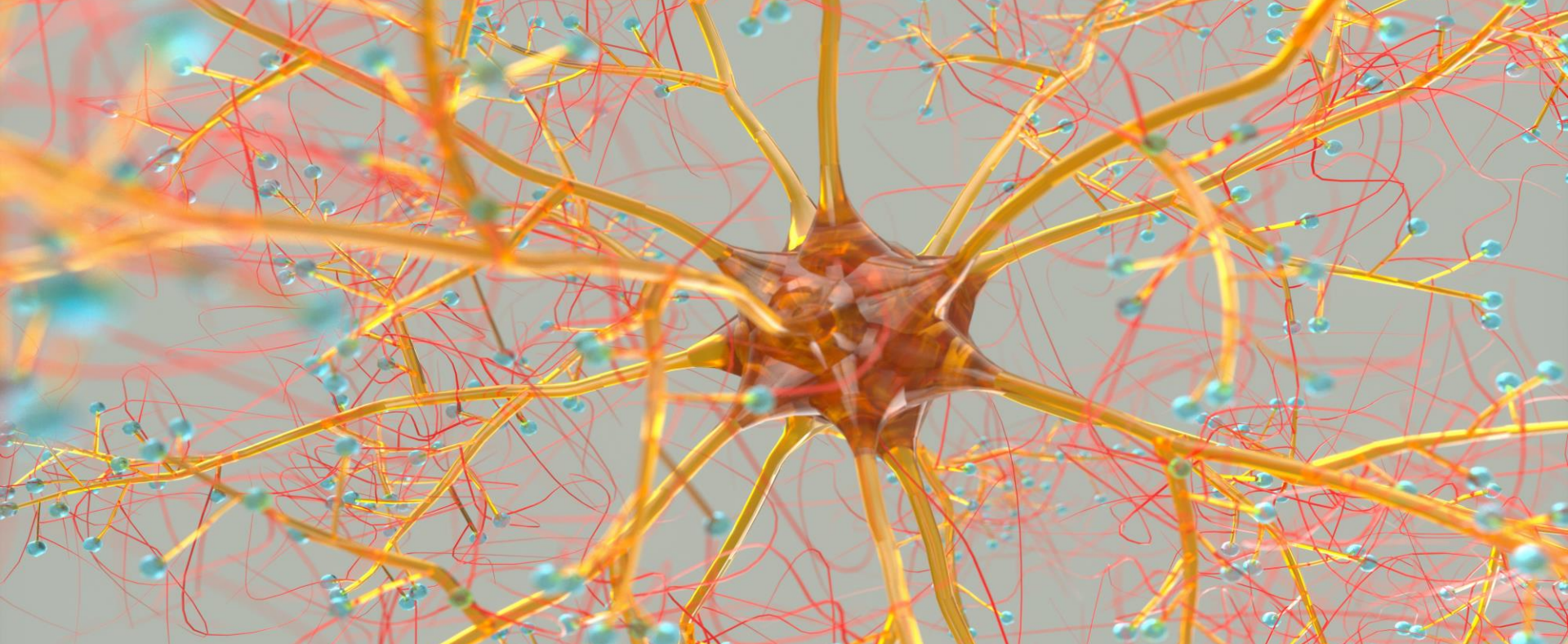
16.4 petaFLOPS of In-Network Computing





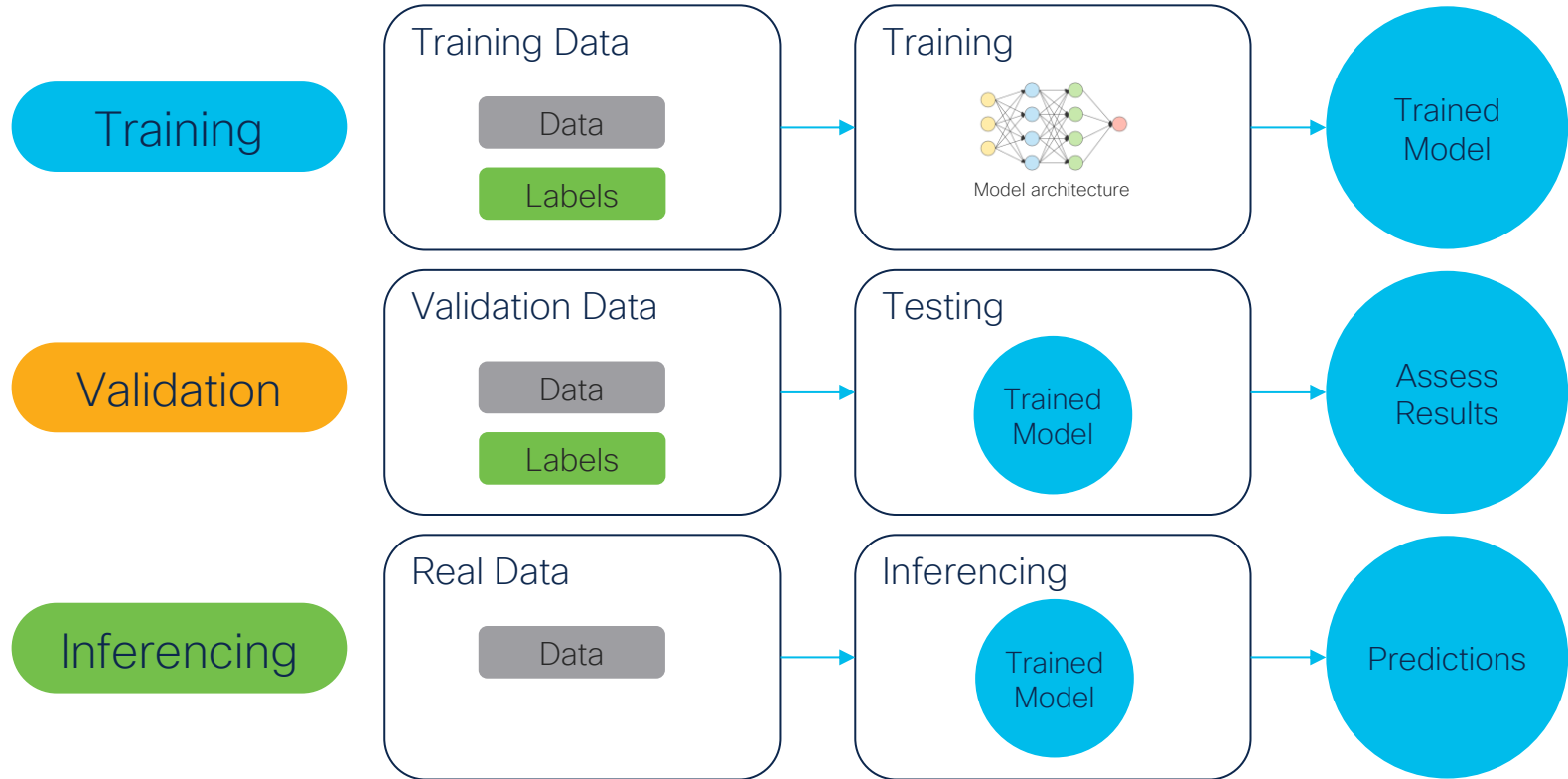
What do I actually need?



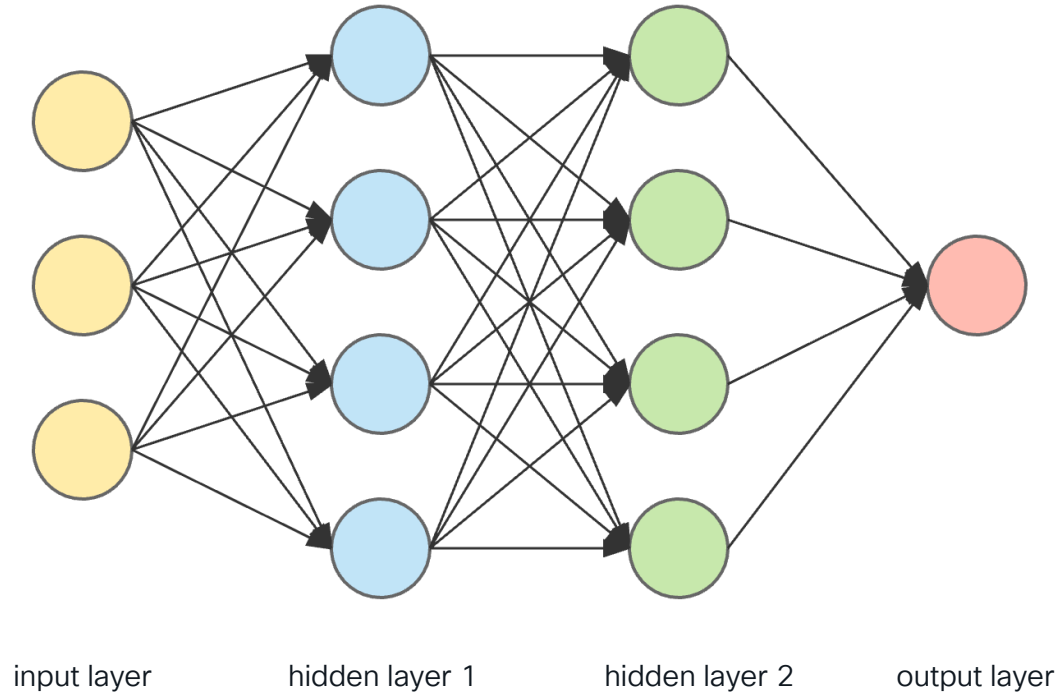


Back to first principles

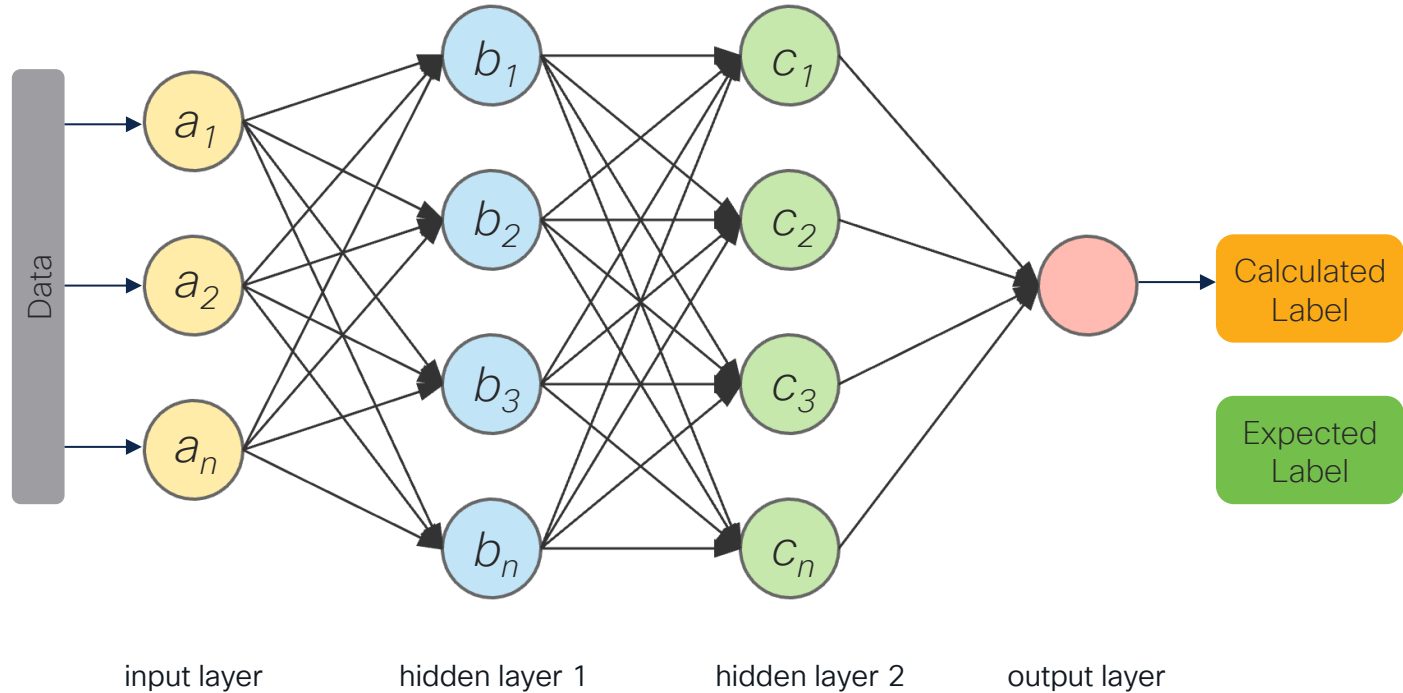
Machine Learning Overview



Model Architecture

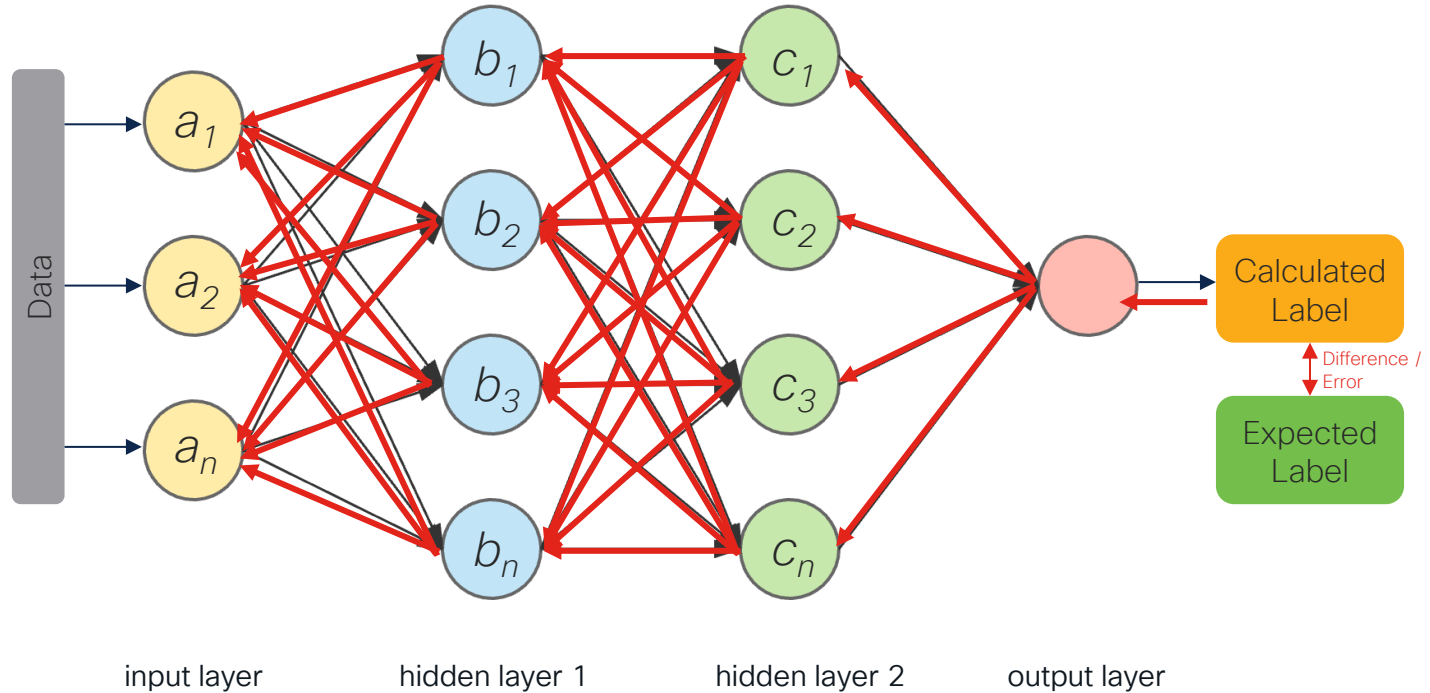


Model Training



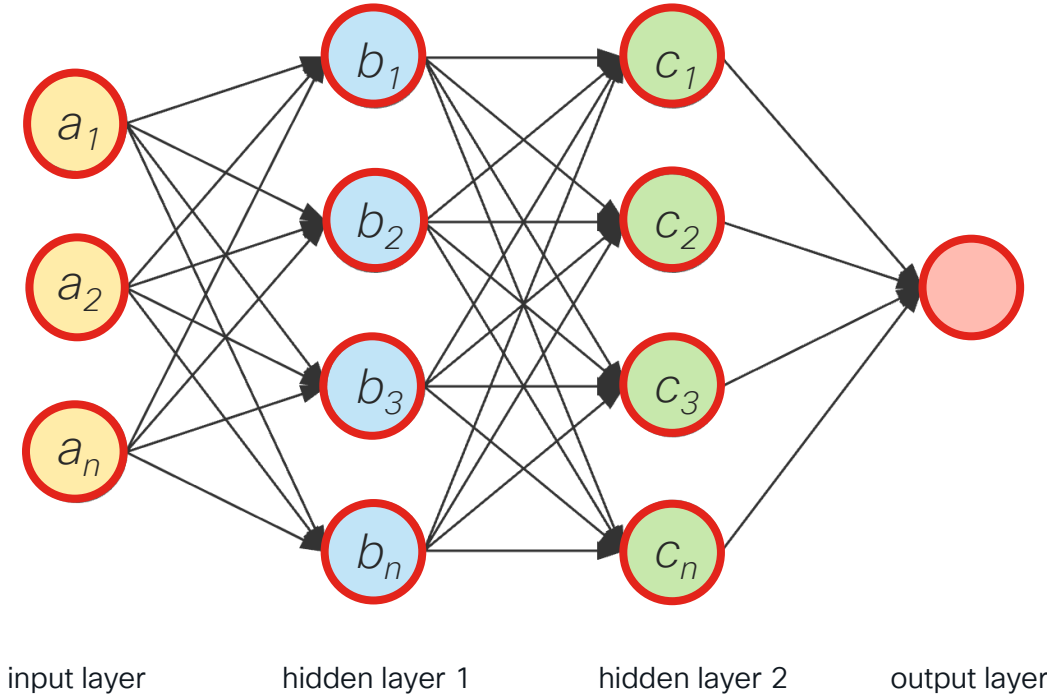
Feed forward phase

Model Training



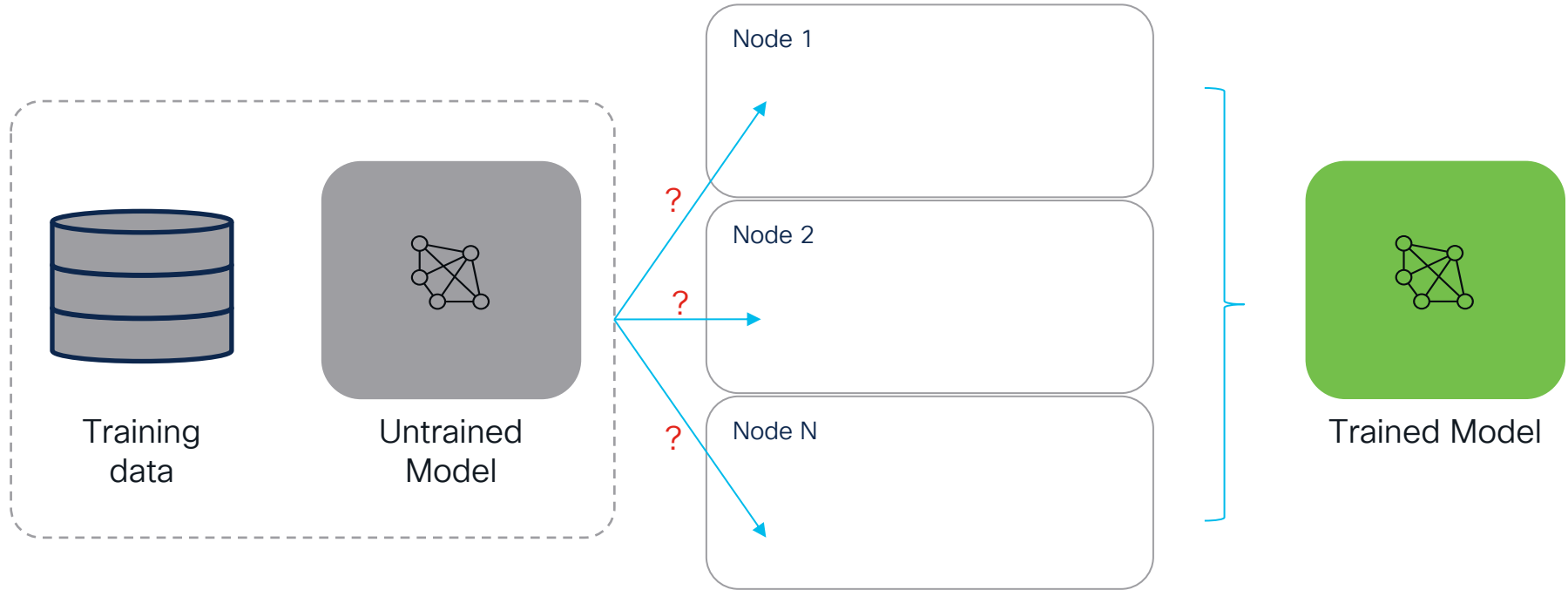
Back propagation phase

Model Training

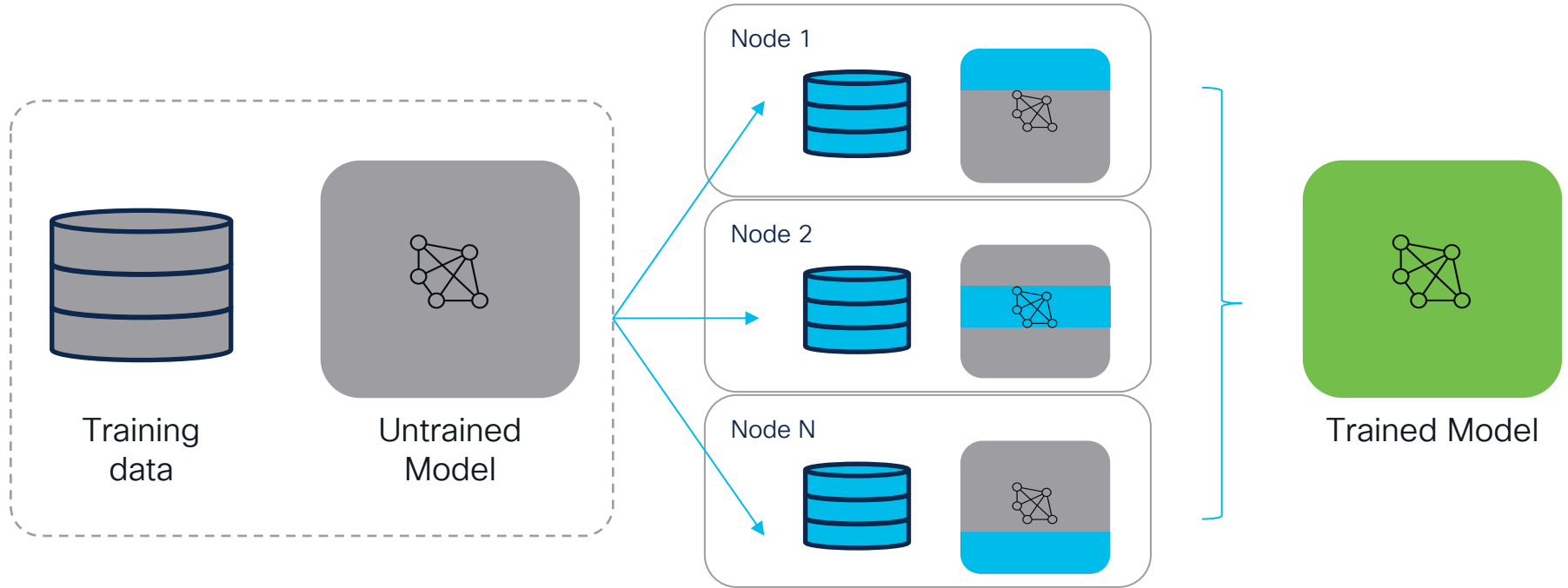


Optimiser phase

Putting the “Distributed” in Distributed ML Training

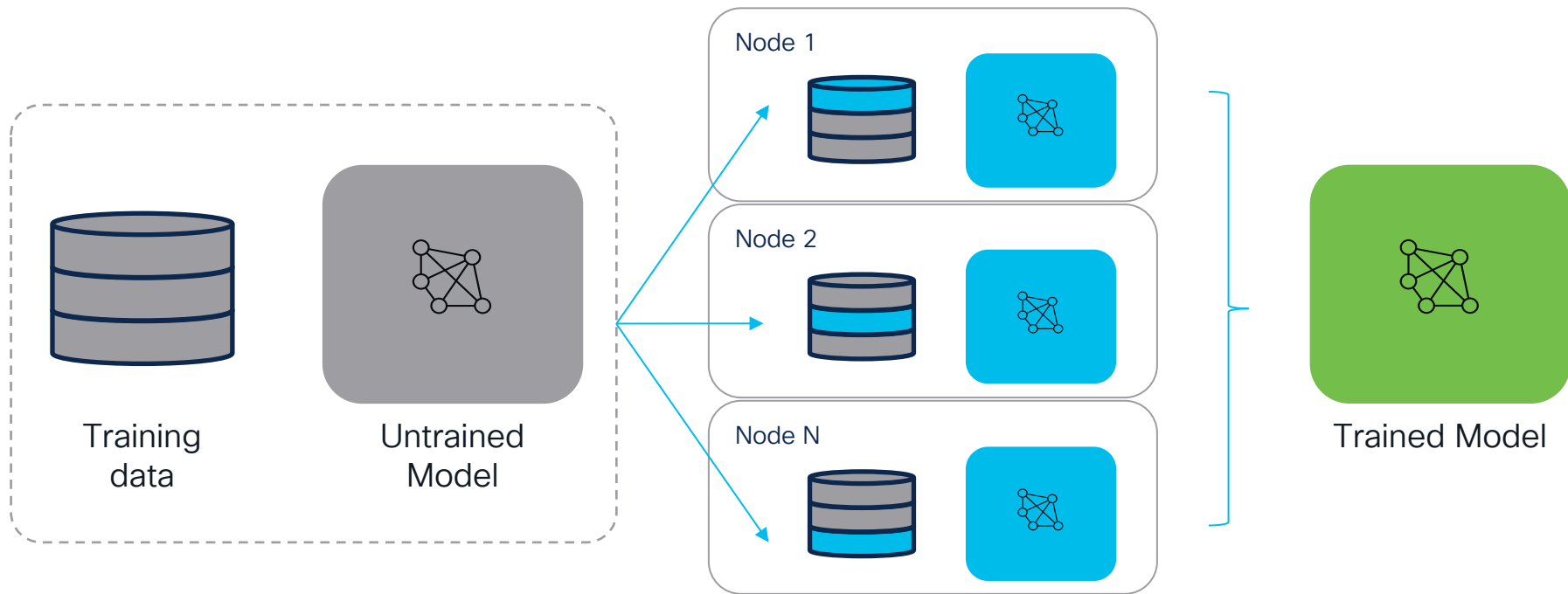


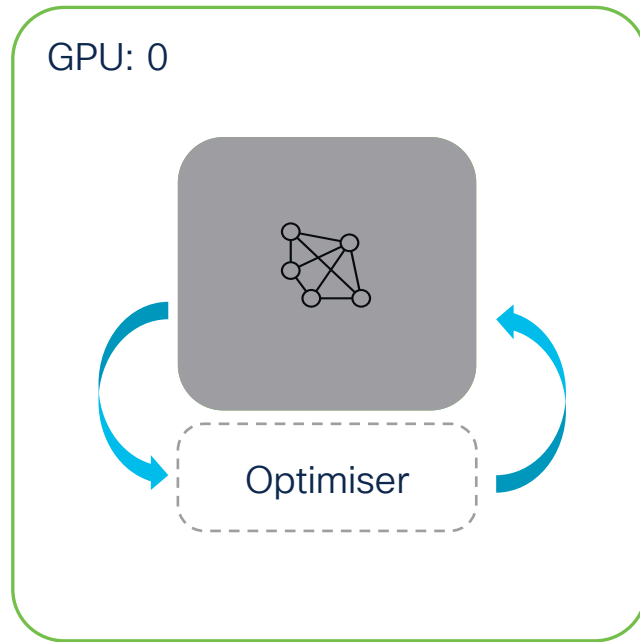
Putting the “Distributed” in Distributed ML Training Model Parallelism



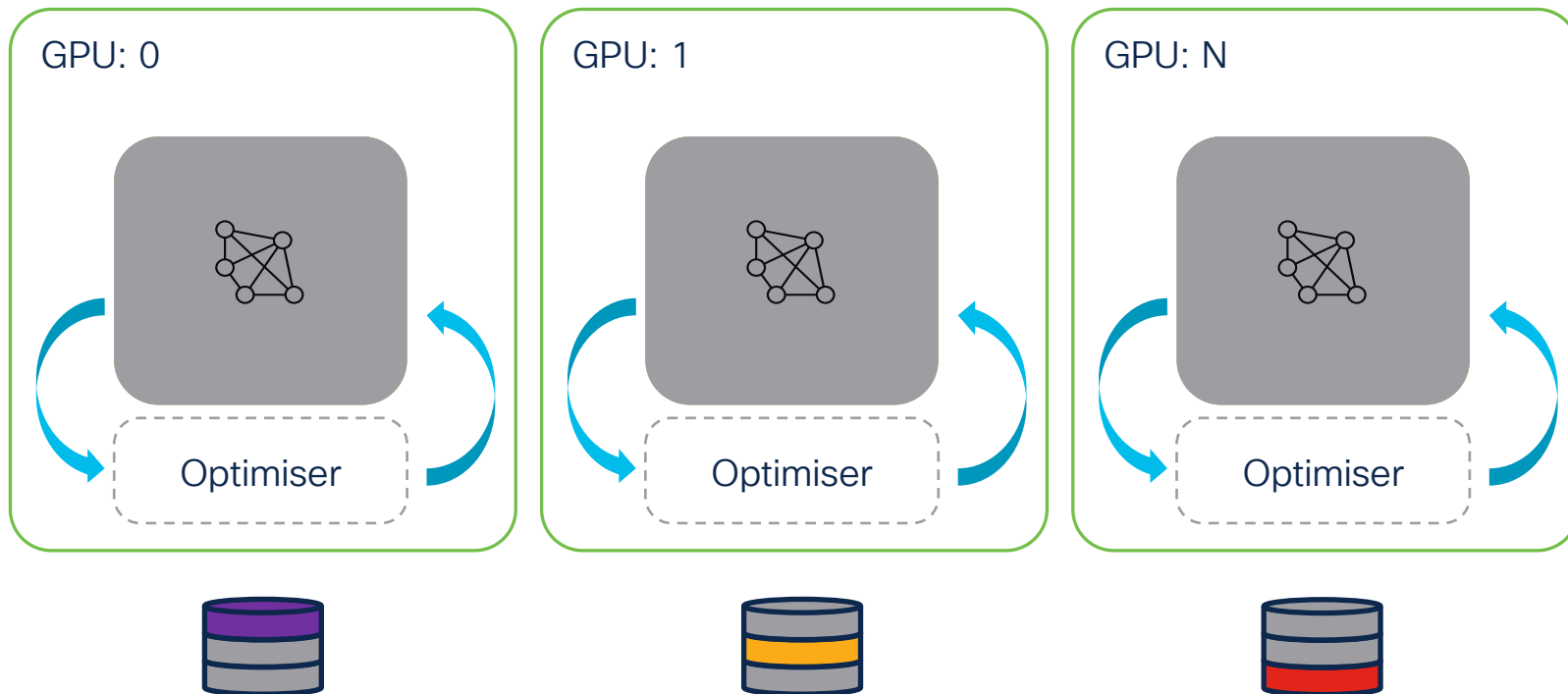
Putting the “Distributed” in Distributed ML Training

Data Parallelism



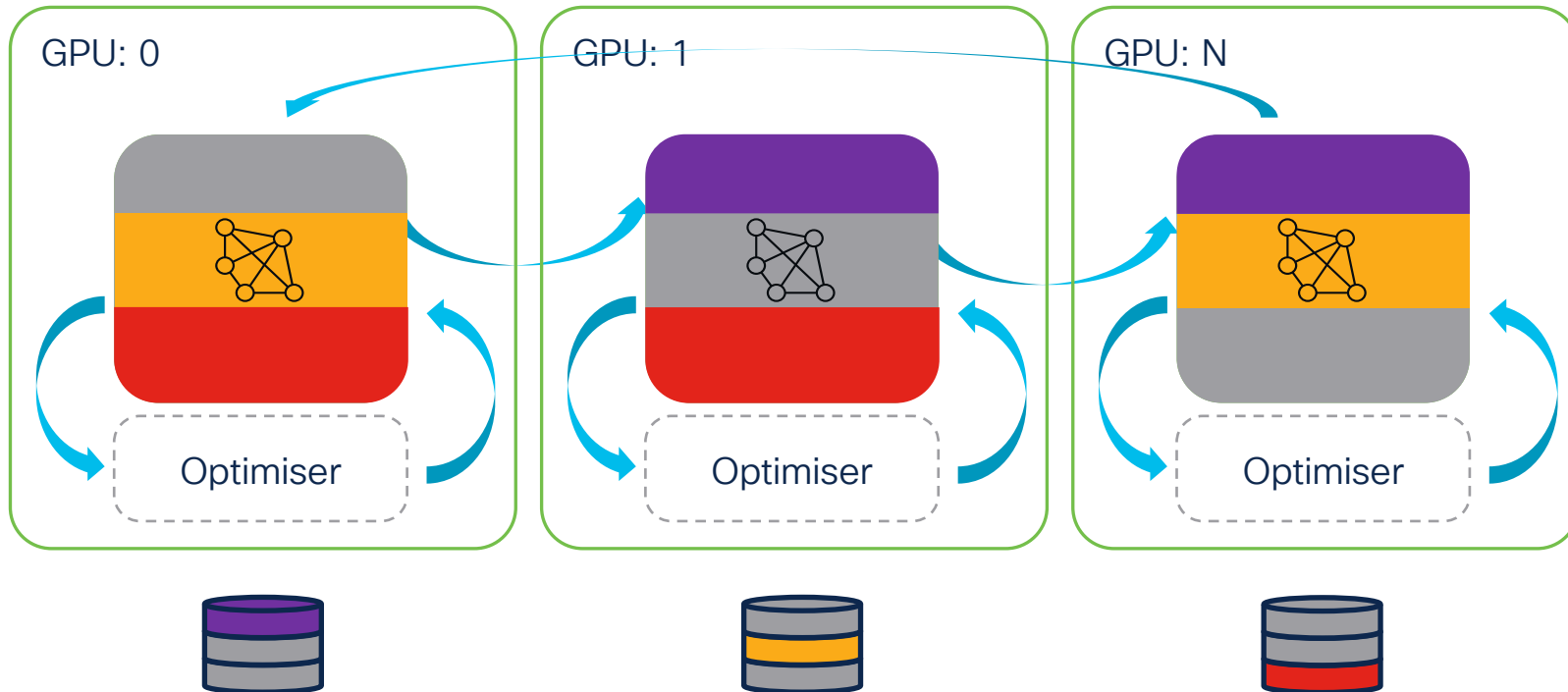


Distributed Training – Naïve approach doesn't work

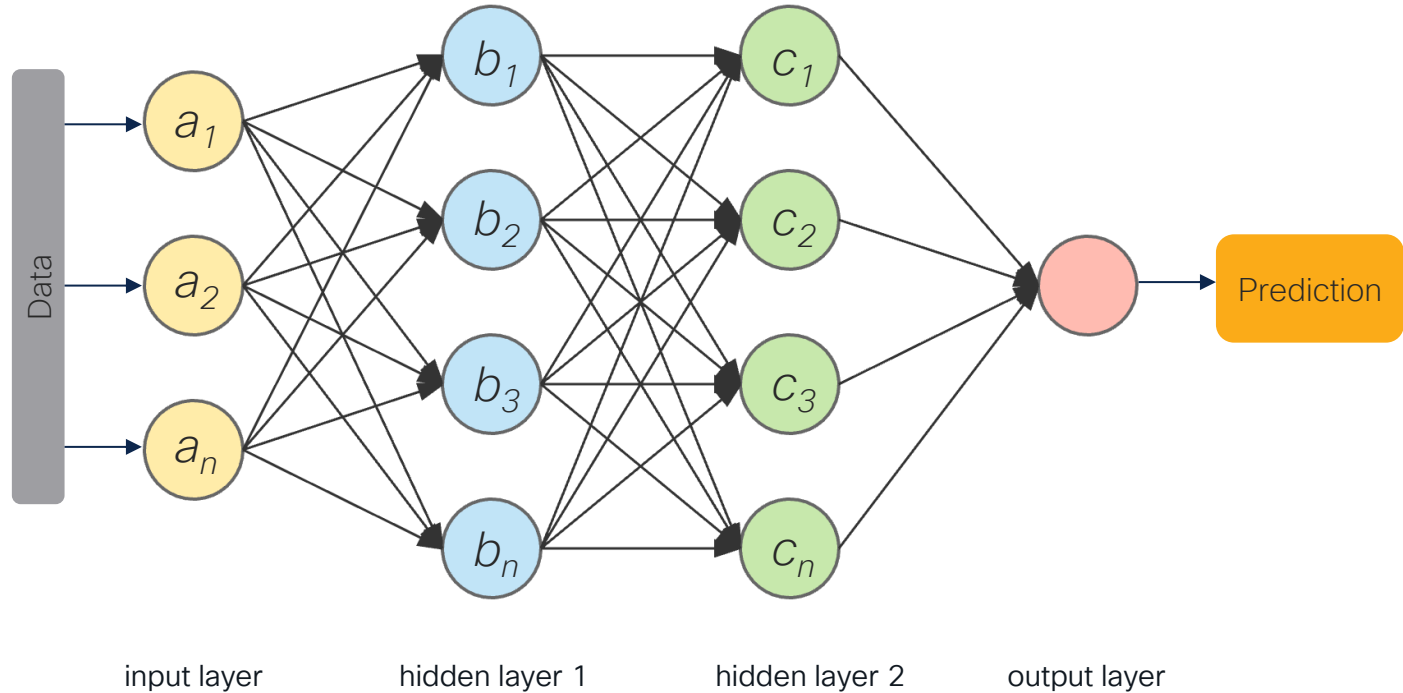


Nvidia Collective Communications Library (NCCL)

AllReduce



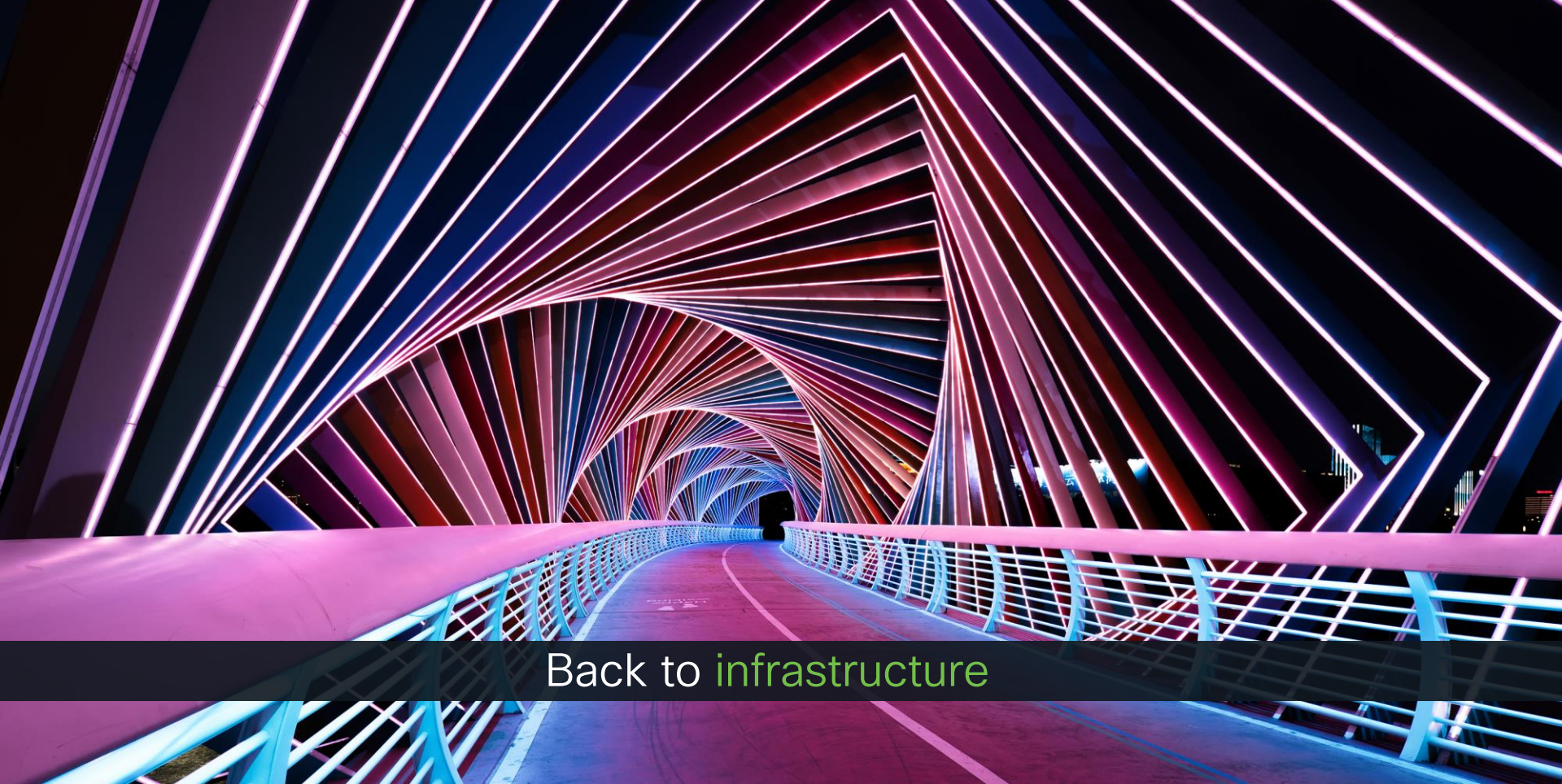
Model Inferencing



Inferencing (Feed forward phase only)

CISCO *Live!*





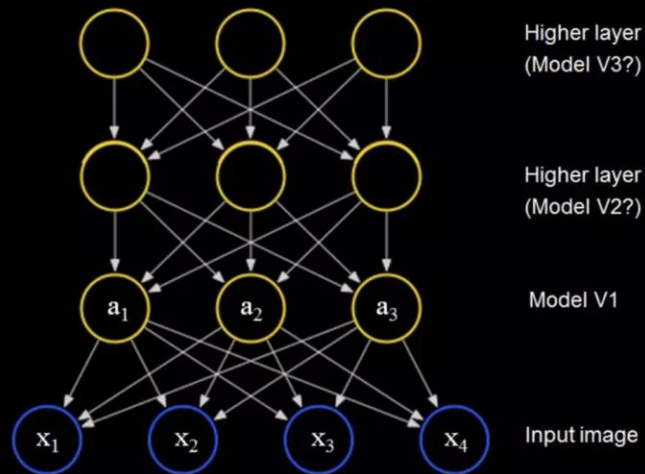
Back to infrastructure

WHY ARE GPUS GREAT FOR DEEP LEARNING?



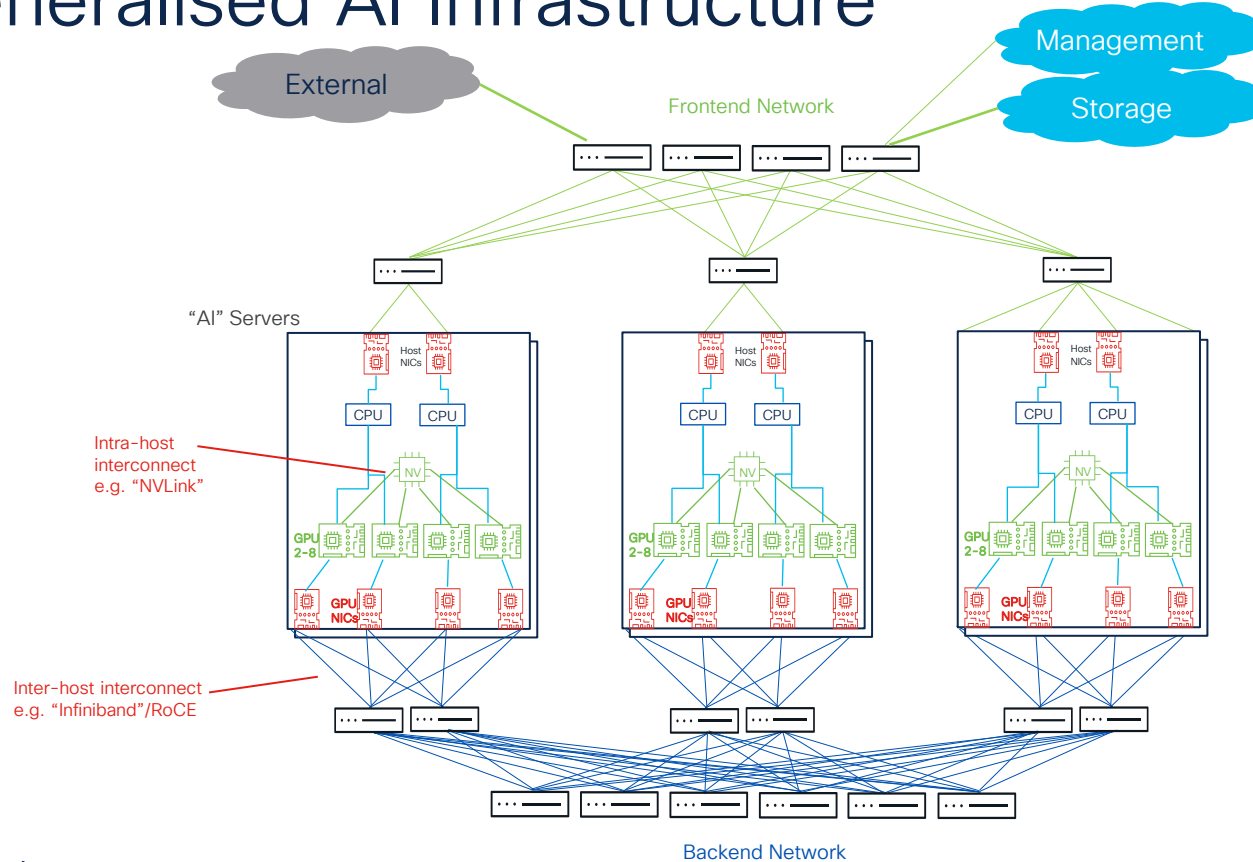
	Neural Networks	GPUs
Inherently Parallel	✓	✓
Matrix Operations	✓	✓
FLOPS	✓	✓

- ▶ GPUs deliver --
 - ▶ same *or better* prediction accuracy
 - ▶ faster results
 - ▶ smaller footprint
 - ▶ lower power

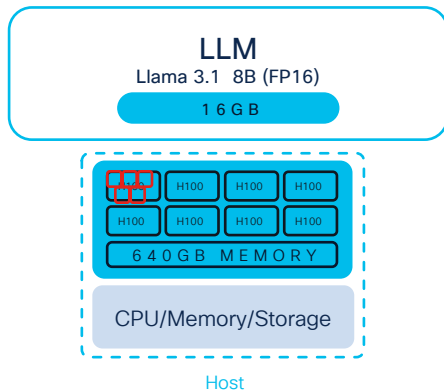


[Lee, Ranganath & Ng, 2007]

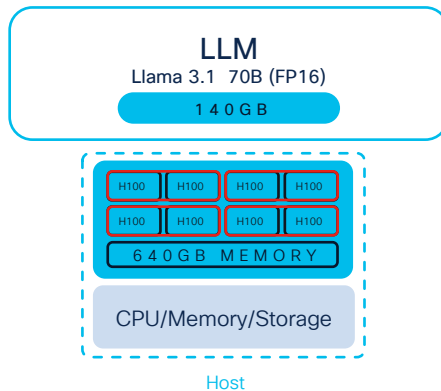
Generalised AI Infrastructure



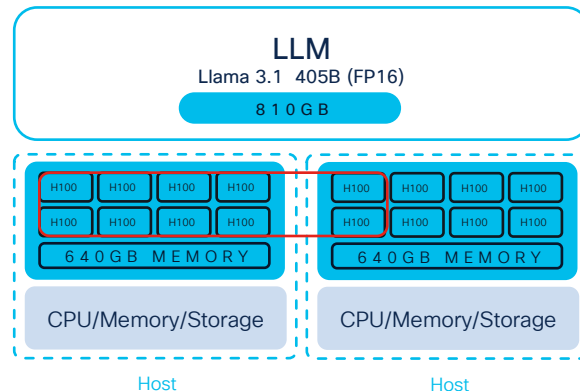
When do GPU interconnects matter for inferencing?



Intra-host and inter-host GPU interconnect not required



Inter-host GPU interconnect not required



Intra-host and inter-host GPU interconnect required

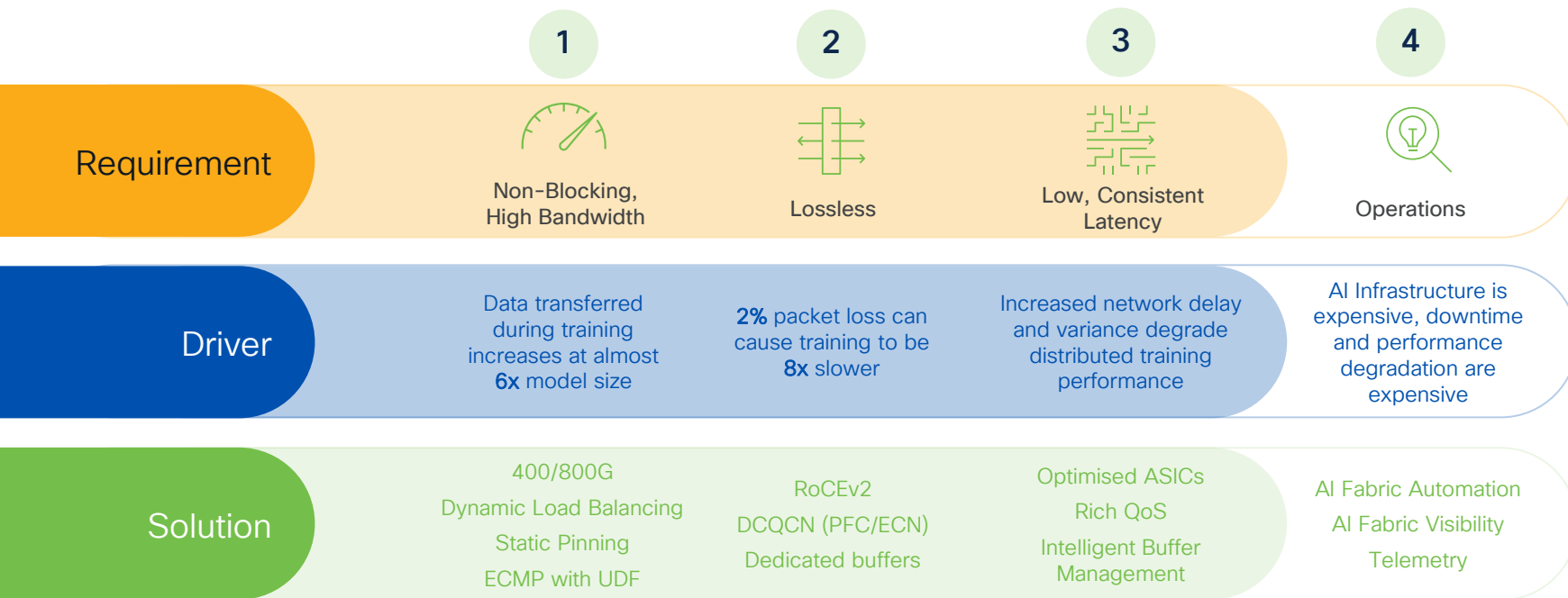
Example Infrastructure Requirements by Workload

Workload type	Small* model inferencing / RAG	Large* model inferencing / RAG	Fine Tuning	Small* model training	Large* model training
AI Accelerator (GPU)	Maybe 0-2 GPU per server	Yes 2-8 GPU per server	Yes 4-8 GPU per server	Yes 2-4 GPU per server	Yes 8 GPU per server
Cluster size	0 – 100's Scales horizontally with request rate	0 – 100's Scales horizontally with request rate	20-80 GPUs ~10's of servers	20-80 GPUs ~10's of servers	>1000 GPUs 100s of servers
Intra-host GPU interconnect (e.g. NVLink)	No	Maybe**	Maybe	Yes	Yes
Inter-host GPU interconnect (“Backend” network)	No	Probably Not**	Maybe 200-400G RoCE	Maybe 200-400G RoCE	Yes 200-400G RoCE/Infiniband
“Frontend” network	“Any”	100-400G	100-400G	100-400G	100-400G

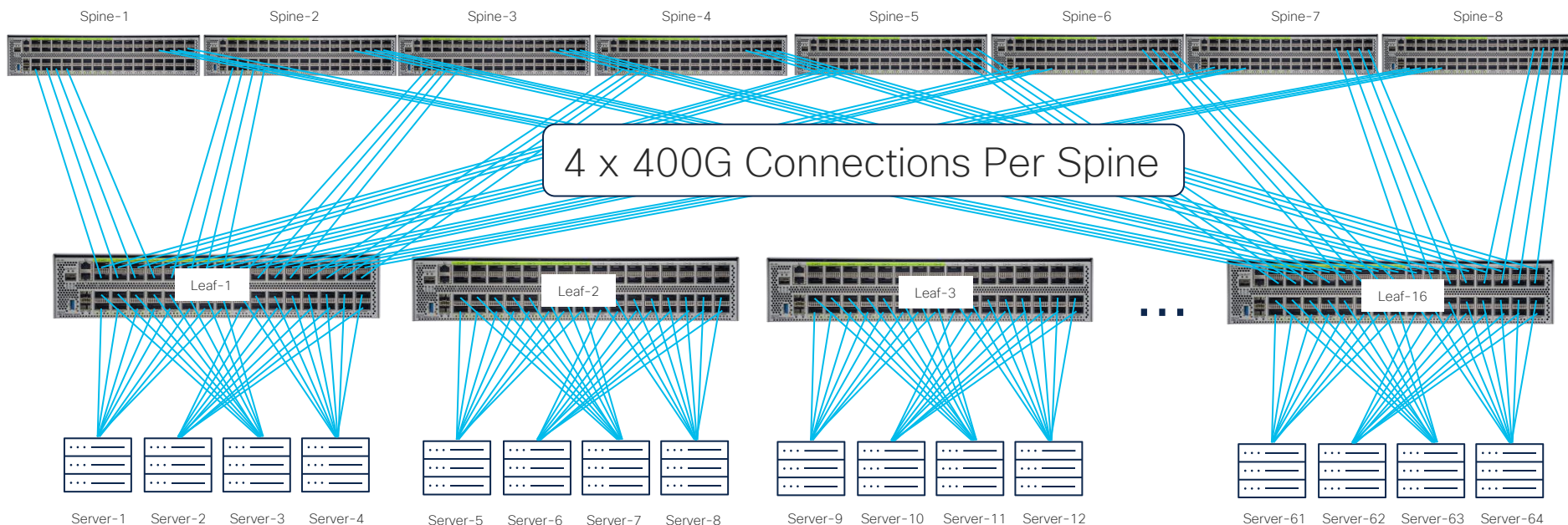
* “Small” and “Large” here are relative to the size of the GPU. i.e. Small can fit in memory of a single GPU.

** Depends on parallelism strategy, e.g. Tensor Parallel, Pipeline Parallel, etc.

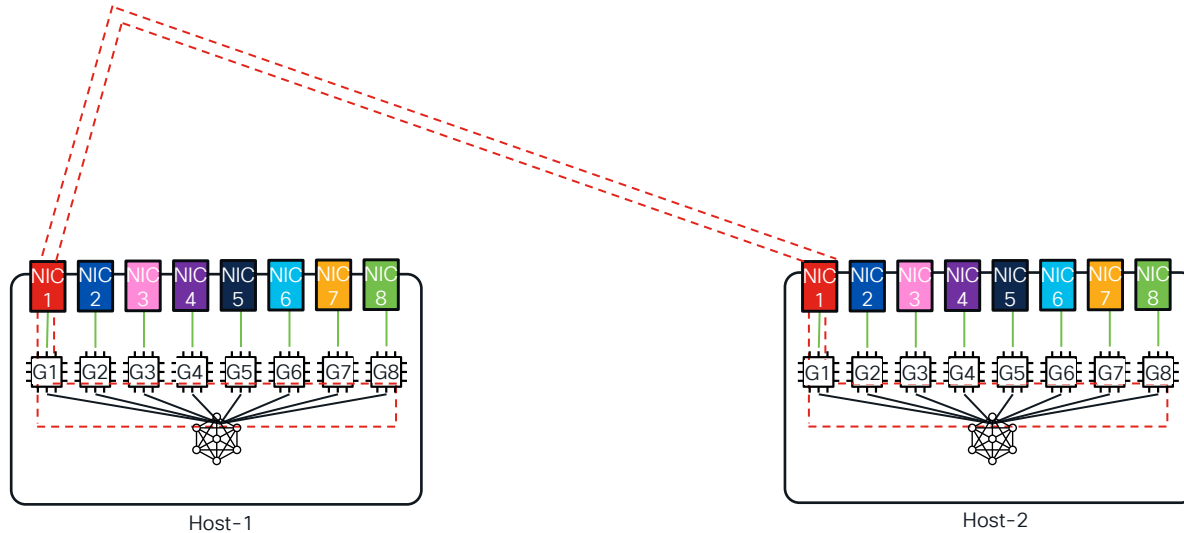
AI/ML Backend Network Requirements



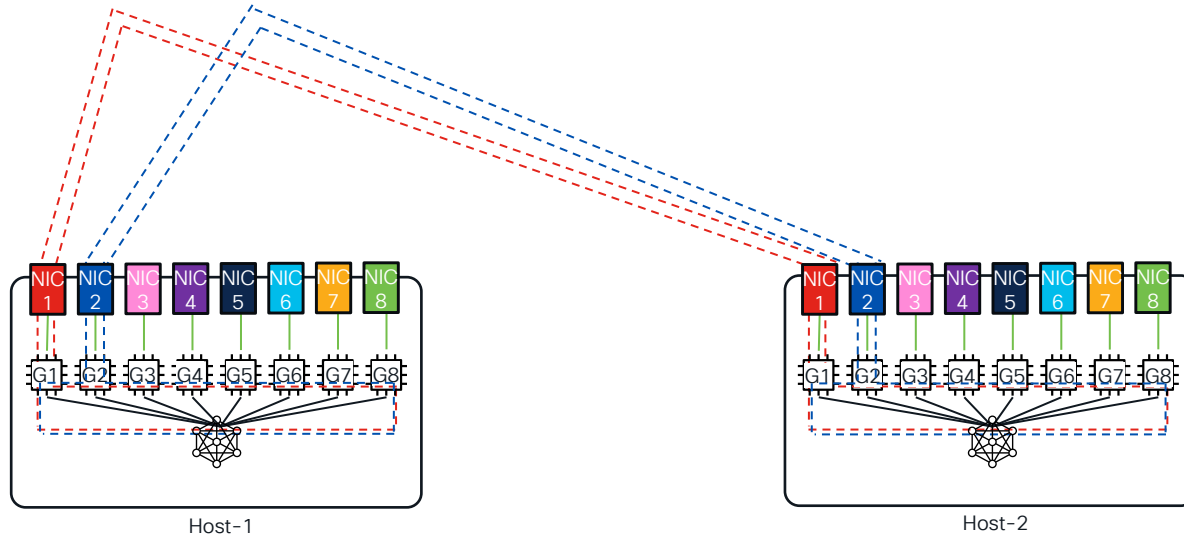
512 GPU Cluster Connected at 400G per GPU



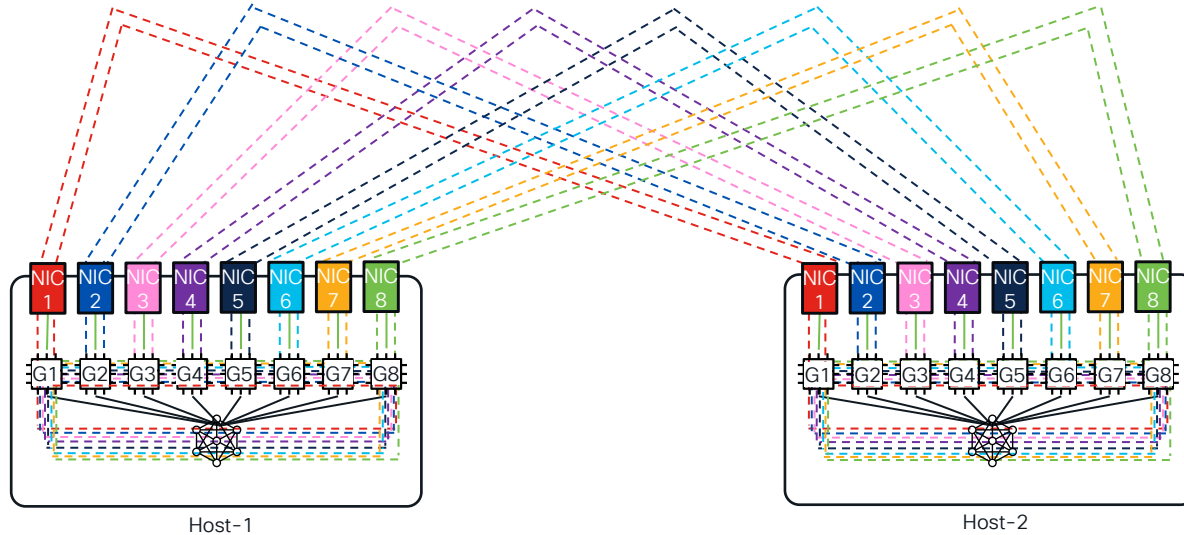
Collective Communication between GPU Servers



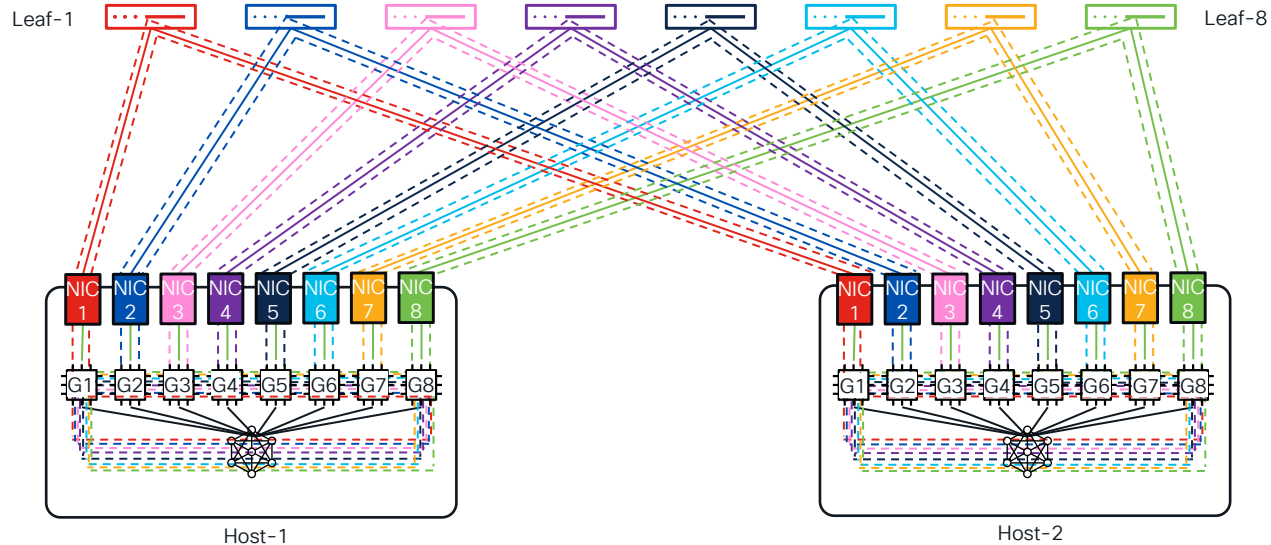
Collective Communication between GPU Servers



Collective Communication between GPU Servers



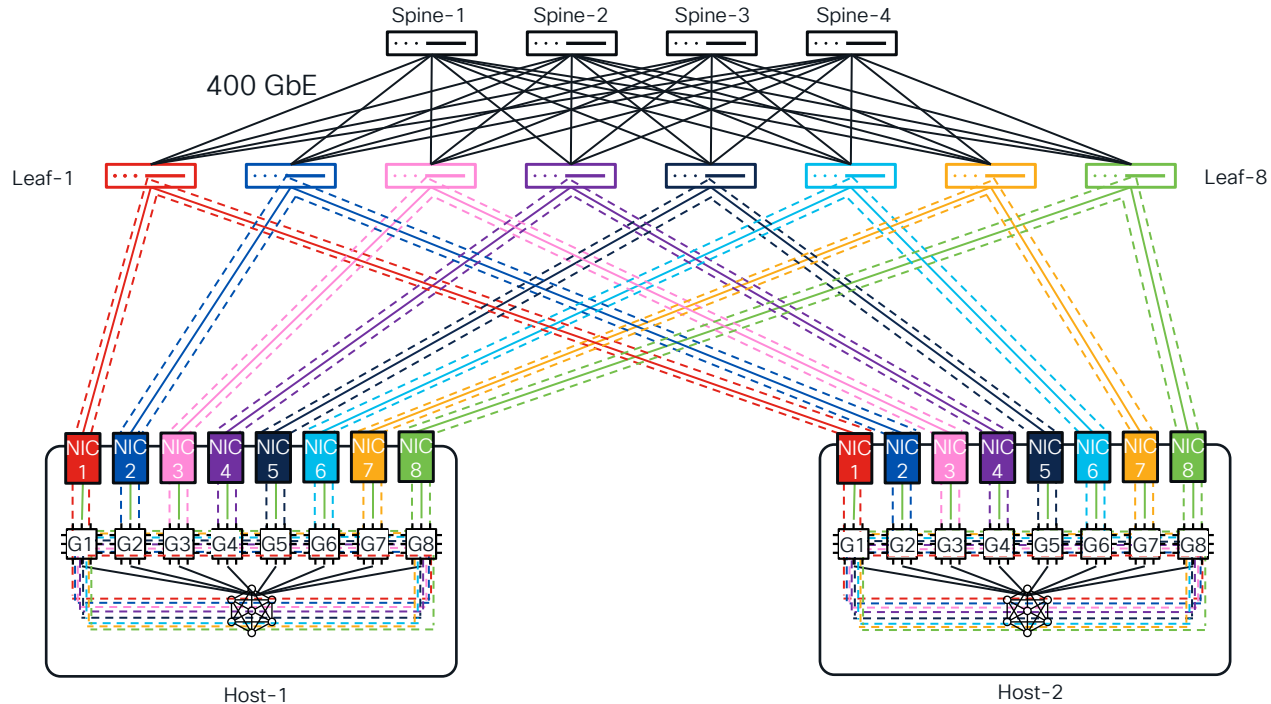
Collective Communication between GPU Servers



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

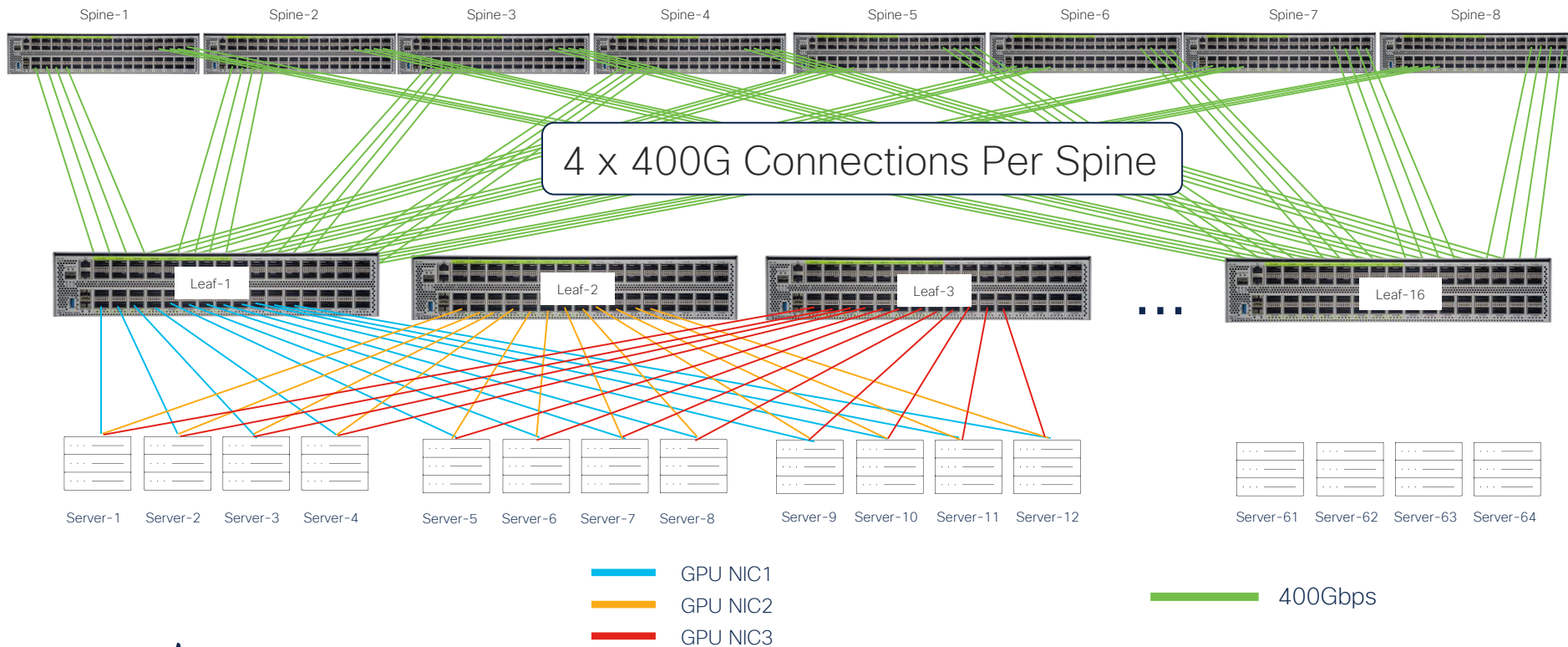
Rails Design for Inter-GPU Network

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



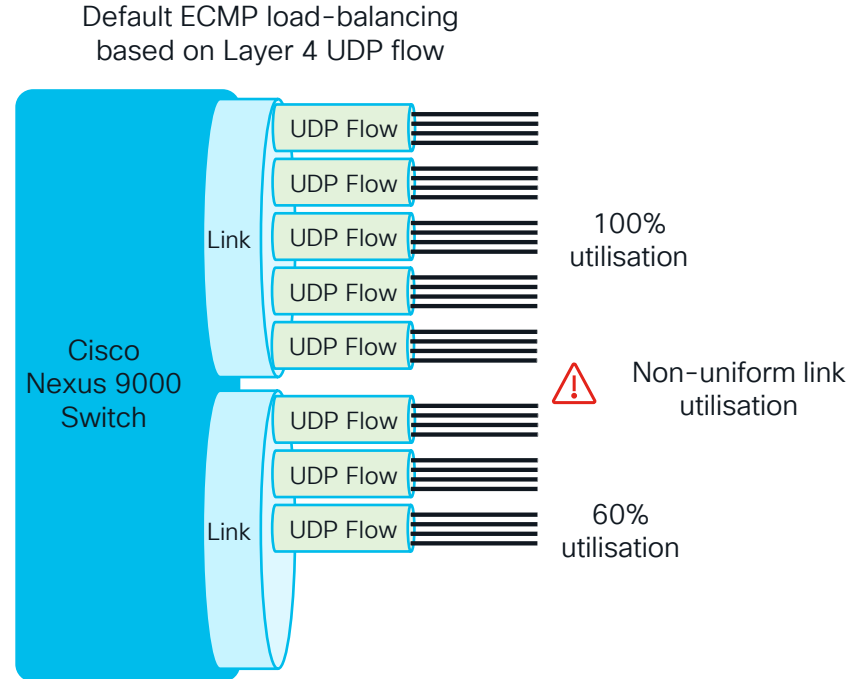
Port 1 on all hosts connects to Leaf-1, Port 2 on all hosts connects to Leaf-2, and so on.

512 GPU Cluster Connected at 400G per GPU Rails Design



Fabric Load Balancing – The challenge

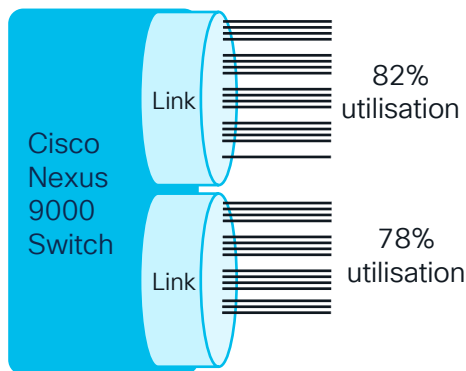
AI/ML backend network traffic (RoCEv2) has very little entropy



Fabric Load Balancing – The solutions

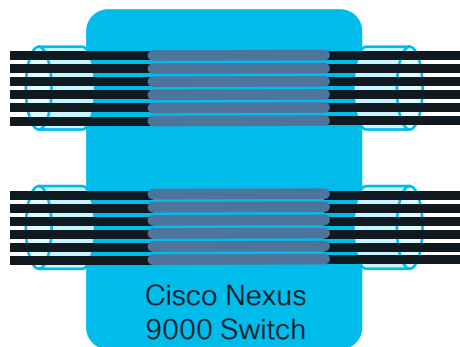
ECMP with UDF

Load-balance based on IP destination queue pair



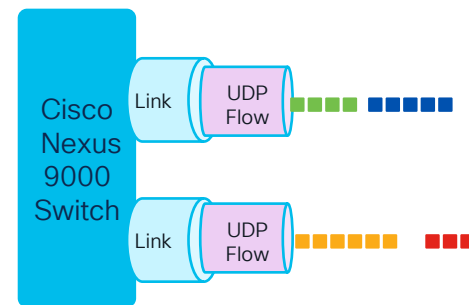
Static Pinning

Statically pin all traffic on an ingress port to an egress port and vice-versa



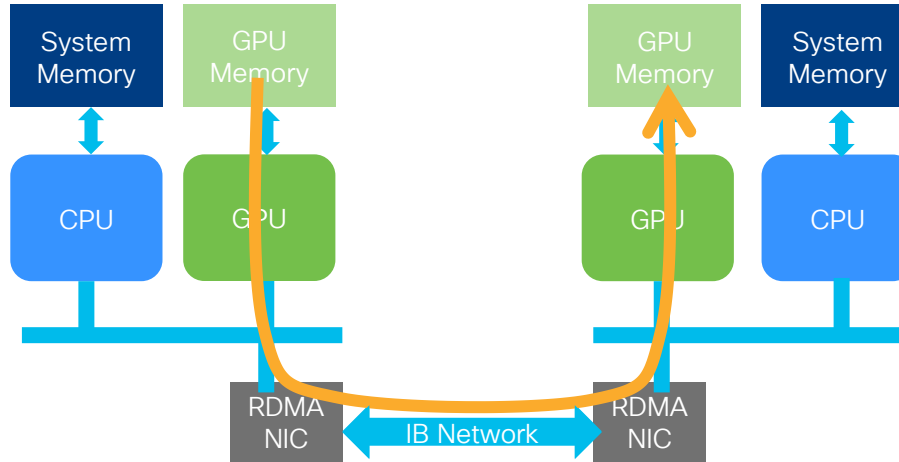
Dynamic Load Balancing (DLB)

Dynamically load balance **flowlets** based on **real-time** link utilisation



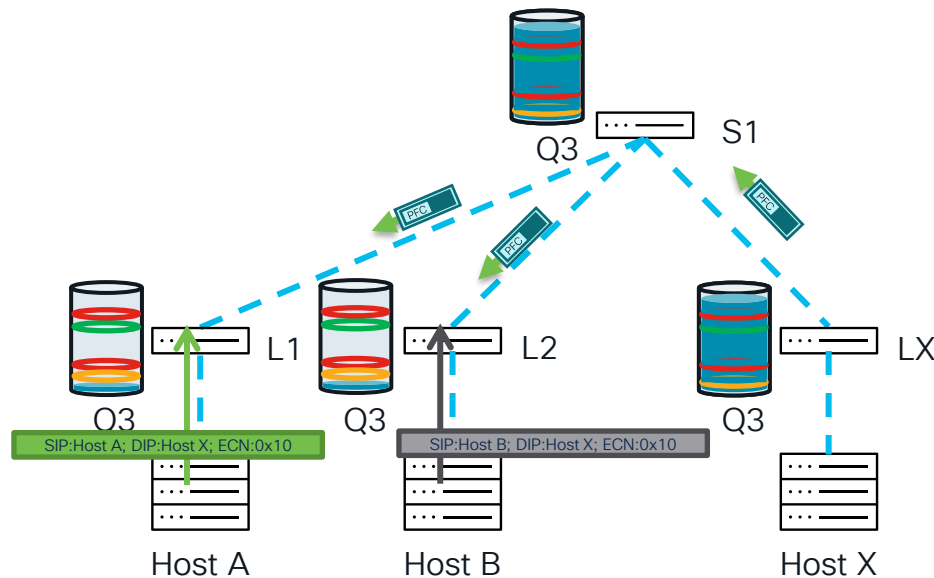
✓ Nearly uniform link utilisation

RDMA Over Converged Ethernet – RoCE/RoCEv2



Congestion Control

- Priority Flow Control (PFC)
- Explicit Congestion Notification (ECN)
- Data Centre Quantised Congestion Notification (DCQCN)
 - PFC + ECN
- See BRKDCN-2921



Nexus Dashboard Fabric Controller



Create Fabric

N9K Cloud Scale Platform Queuing Policy

Select an Option

Queuing Policy for all 92xx, -EX, -FX, -FX2, -FX3, -GX series switches in the fabric

N9K R-Series Platform Queuing Policy

Select an Option

Queuing Policy for all R-Series switches in the fabric

Other N9K Platform Queuing Policy

Select an Option

Queuing Policy for all other switches in the fabric

Enable AI / ML QoS and Queuing Policies



Configures QoS and Queuing Policies specific to N9K Cloud Scale switch fabric for AI / ML network loads

AI / ML QoS & Queuing Policy*

AI_Fabric_QOS_100G

Queuing Policy based on predominant fabric link speed: 400G / 100G / 25G

AI_Fabric_QOS_400G

AI_Fabric_QOS_100G

Enable MACsec in the fabric

AI_Fabric_QOS_25G

Cisco Type 7 Encrypted Octet String

Day0 Provisioning for AI/ML Networks

Edit interface(s)

Additional CLI for the interface

Enable Interface*



Uncheck to disable the interface

Enable Netflow



Netflow is supported only if it is enabled on fabric

Netflow Monitor

Provide the Layer 2 Monitor Name

Netflow Sampler

Netflow sampler name, applicable to N7K only

Enable priority flow control



Enable priority flow control

Enable QoS Configuration



Enable to configure a QoS Policy for this interface. If AIML Queuing is enabled on the fabric, will use the QOS_CLASSIFICATION policy. Enter a custom policy below to override

Custom QoS Policy

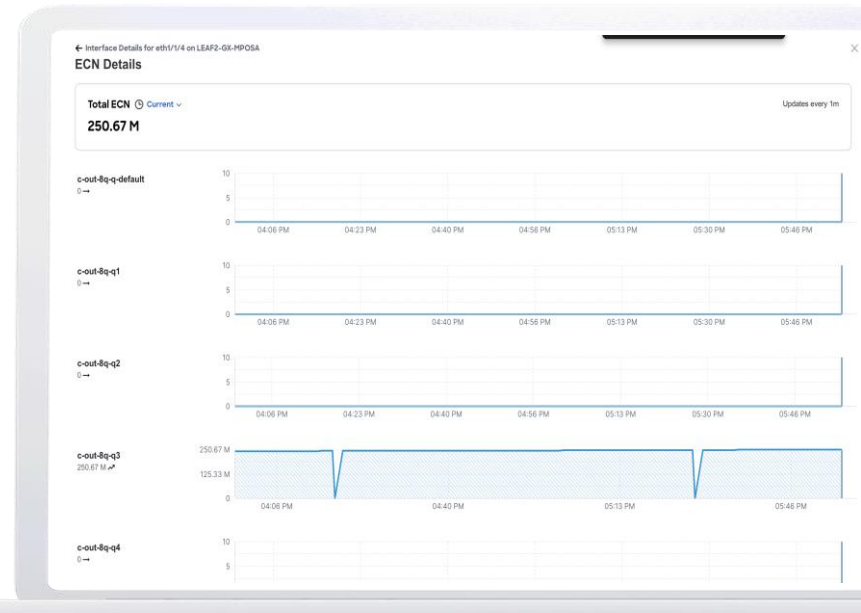
Custom QoS Policy must be defined previously

Custom Queuing Policy

Queuing Policy must be defined previously

Custom QoS Classification/Marking

Nexus Dashboard Insights



Congestion Scores

Network Hotspots

PFC, ECN Statistics

AI/ML Networking Blueprint



Building a 400Gbps Back-End Network

Table of Contents

Introduction

RoCEv2 as Transport for AI Clu...

AI Clusters Require Lossless N... +

How to Manage Congestion E... +

How Visibility into Network Be... +

Network Design to Accommod... -

Building a Shared 100Gbps Network

Building a 400Gbps Back-End Netw...

Using Cisco Nexus Dashboard Fabr...

Conclusion

Related Materials

Building a 400Gbps Back-End Network

For higher performance use cases, where GPU to GPU traffic requires 400Gbps Ethernet connectivity, it may be preferred to build a separate network for these communications. These types of networks are often referred to as a “back-end network.” Back-end networks generally are designed for GPUDirect (GPU to GPU) or GPUDirect Storage (GPU to storage) communications. The requirements for back-end networks are the same as any RoCEv2 network discussed in this document. The main difference for this type of network, assuming that it is connected at 400Gbps, is the recommended type of leaf and spine switches. To build a 400Gbps RoCEv2 network either the Cisco Nexus 9332D-GX2B or the Cisco Nexus 9364D-GX2A should be used as leaf switches. The Cisco Nexus 9364D-GX2A is the recommended spine switch. As with the network described earlier in this document a non-blocking fabric is required.

We will use the example of building a 512 GPU cluster. For this back-end network we will need to connect 512 400Gbps NICs. We will use the Cisco Nexus 9364D-GX2A as a leaf switch which will allow us to connect 32 GPUs/NICs to each leaf, leaving 32 ports for spine connectivity to build a non-blocking fabric. We will require 16 leaf switches to connect all 512 NICs.

To accommodate the amount of bandwidth coming from the leaf switches, 512 x 400Gbps ports are needed to the spines. We will use 8 Cisco Nexus 9364D-GX2A spine switches. Each leaf will connect to every spine using 4 x 400Gbps ports. The network is represented in the following diagram:

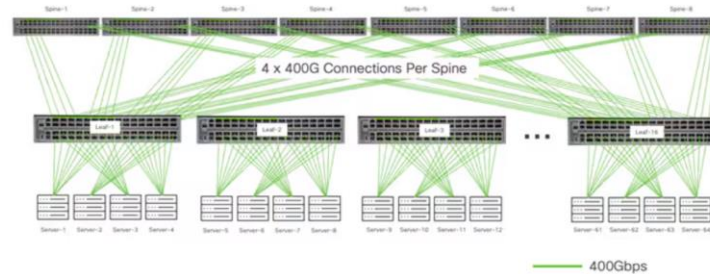
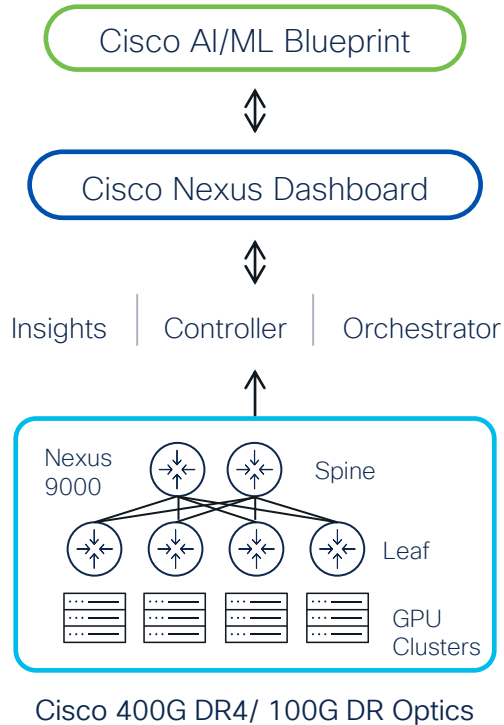


Figure 19. Spine-leaf network design, connecting 512 GPUs in a non-blocking network.

Nexus: Power of Systems



Outcomes



Best performing

AI/ML networks, focus on app performance



Intelligent buffering, low latency, telemetry / visibility, RoCEv2



Dynamic congestion avoidance for various workloads



One IP / Ethernet network vs. dedicated front-end / back-end



Lower TCO



Validated designs for network and ecosystem partners

A Brief Word on Ultra Ethernet



Ultra Ethernet Specification Update

By Ultra Ethernet Consortium | August 29, 2024



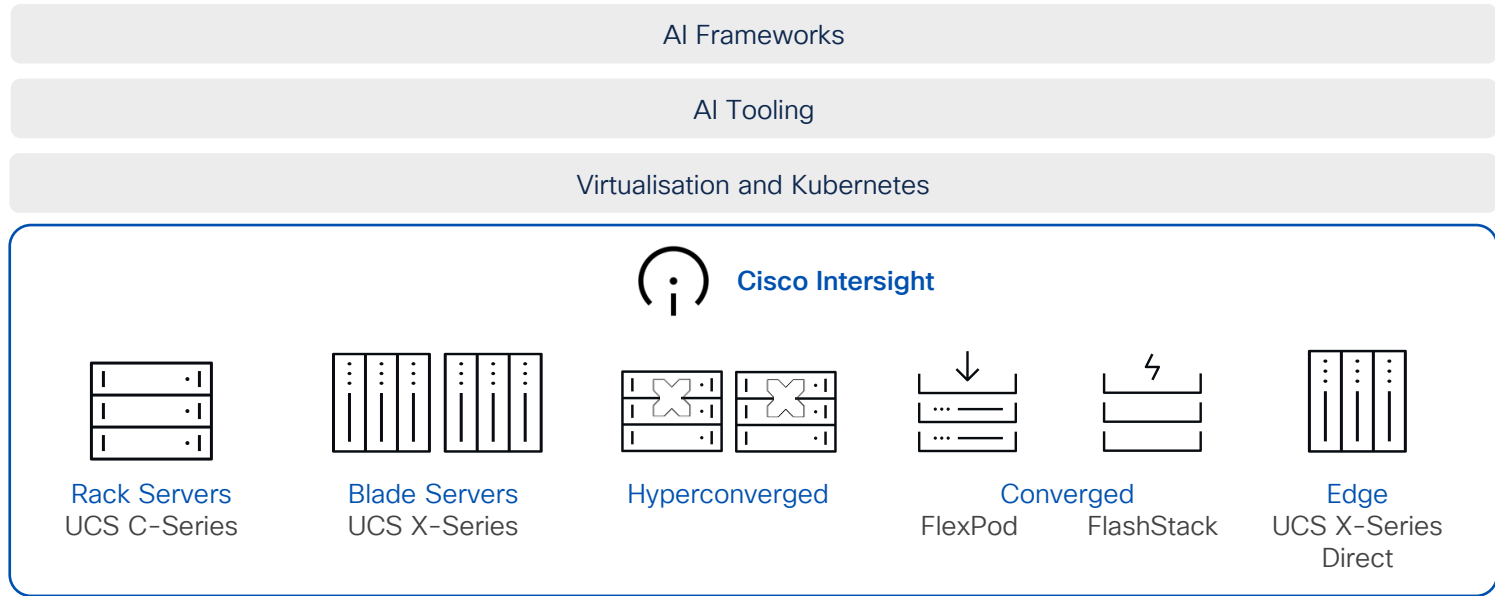
Example Infrastructure Requirements by Workload

Workload type	Small* model inferencing / RAG	Large* model inferencing / RAG	Fine Tuning	Small* model training	Large* model training
AI Accelerator (GPU)	Maybe 0-2 GPU per server	Yes 2-8 GPU per server	Yes 4-8 GPU per server	Yes 2-4 GPU per server	Yes 8 GPU per server
Cluster size	0 - 100's Scales horizontally with request rate	0 - 100's Scales horizontally with request rate	20-80 GPUs ~10's of servers	20-80 GPUs ~10's of servers	>1000 GPUs 100s of servers
Intra-host GPU interconnect (e.g. NVLink)	No	Maybe**	Maybe	Yes	Yes
Inter-host GPU interconnect ("Backend" network)	No	Probably Not**	Maybe 200-400G RoCE	Maybe 200-400G RoCE	Yes 200-400G RoCE/Infiniband
"Frontend" network	"Any"	100-400G	100-400G	100-400G	100-400G

* "Small" and "Large" here are relative to the size of the GPU. i.e. Small can fit in memory of a single GPU.

** Depends on parallelism strategy, e.g. Tensor Parallel, Pipeline Parallel, etc.

Broad Spectrum of Compute Form Factors



GPU-Accelerated Rack Servers

C-Series Rack Servers

C240 M6 INTEL
C245 M6 AMD



5x NVIDIA A10
3x NVIDIA A16
3x NVIDIA A30
3x NVIDIA A40
3x NVIDIA A100-80

C240 M7 INTEL



3x NVIDIA A16
3x NVIDIA A30
3x NVIDIA A40
3x NVIDIA A100-80
2x NVIDIA H100-80
3x NVIDIA L40
8x NVIDIA L4
5X INTEL FLEX140
3X INTEL FLEX170
NVIDIA L40S

C220 M6 INTEL
C225 M6 AMD



3x NVIDIA T4
NVIDIA L4

C220 M7 INTEL



3x NVIDIA L4
3X INTEL FLEX140

C245 M8 AMD
(server not shipping yet)



NVIDIA H100-80
NVIDIA L40S
NVIDIA L40
NVIDIA L4
NVIDIA H100-NVL
NVIDIA A16



UCS X-Fabric Technology and PCIe Nodes with GPU



PCIe node supports up to:

4x

Intel Data Center
GPU Flex 140

2x

Intel Data Center
GPU Flex 170

2x

Nvidia A16

2x

Nvidia H100

Nvidia L40

Nvidia L40S

4x

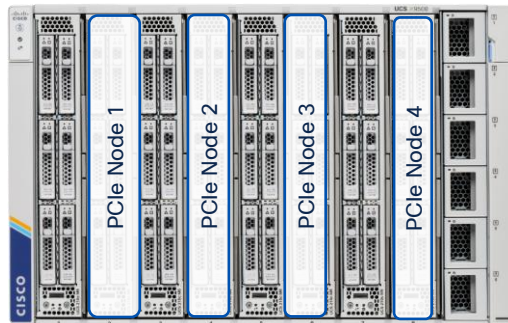
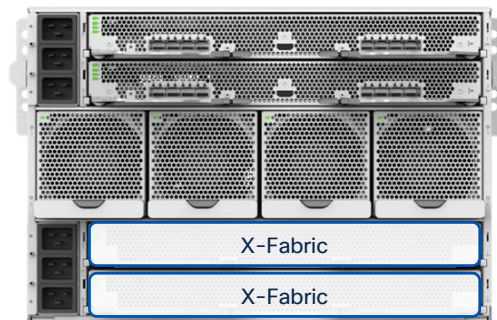
Nvidia T4

Nvidia A40

Nvidia L4

Nvidia A100

Nvidia H100 NVL



X-Fabric decouples the lifecycles of CPU, GPU, memory, storage and fabrics - providing a perpetual architecture that efficiently brings you the latest innovations.

- ✓ Cloud-powered composability with Cisco Intersight
- ✓ Flexible GPU acceleration across server nodes
- ✓ No backplane or cables = easily upgrades

High-Density GPU Servers

Built for data-intensive use cases like model training and deep learning

UCS Accelerated | UCS c885A M8

NEW



2 CPUs

AMD Genoa (96 cores)/Turin

NVIDIA HGX with 8 GPUs

NVIDIA H100, H200, B100 with NVLink
AMD MI300 with Infinity Fabric

Network

(8) NVIDIA ConnectX-7 BF3 B3140H (E-W)
(5) NVIDIA BF3 B3220, B3240 (N-S)

Power

(6) 3000W and (2) 2400W

System Tunning Recommendations

BIOS tunings:

- **Processor Configuration:**

Intel Virtualisation Technology : <Enabled>

Intel Hyper-Threading Technology : <Enabled>

CPU Performance : <HPC>

- **Power & Performance Configuration:**

Processor C1E : <Disabled>

Processor C6 : <Enabled>

Package C-State control : <C0/C1 State>

Uncore Frequency Scaling: <Enabled>

Thermal : Fan Control Policy : <Maximum Power>

PCIe : Width, Speed, Max Payload Size, Max Read Request

OS tunings : CPU Power, TCP Parameters & Memory Settings

C1E is one of the automatic power-saving features which is triggered when the system is idle. **Best to disable it when overclocking**

C6 BIOS will automatically disable CPU core and cache for power saving

In C0/C1 state all of the cores are locked at maximum performance and will cause a higher consumption of power

Uncore Frequency Scaling “enabled” stops the processor from dynamically changing frequencies based on workload to save power



AI Pods

Faster time to value with pre-configured solutions

COMING SOON

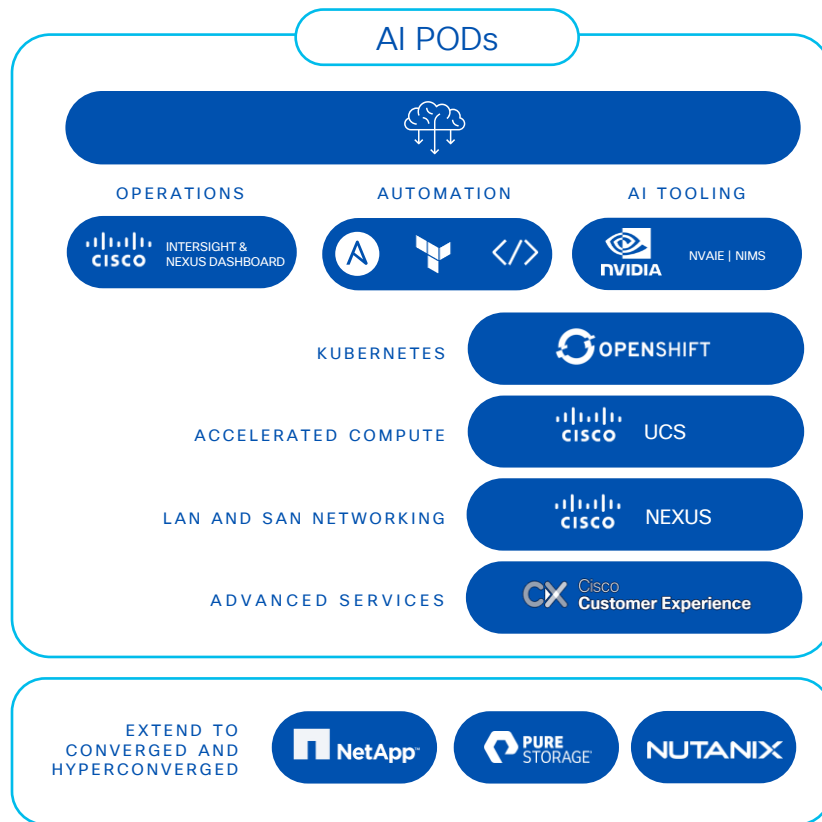
Deploy AI with confidence

Orderable, validated AI-Ready infrastructure stacks

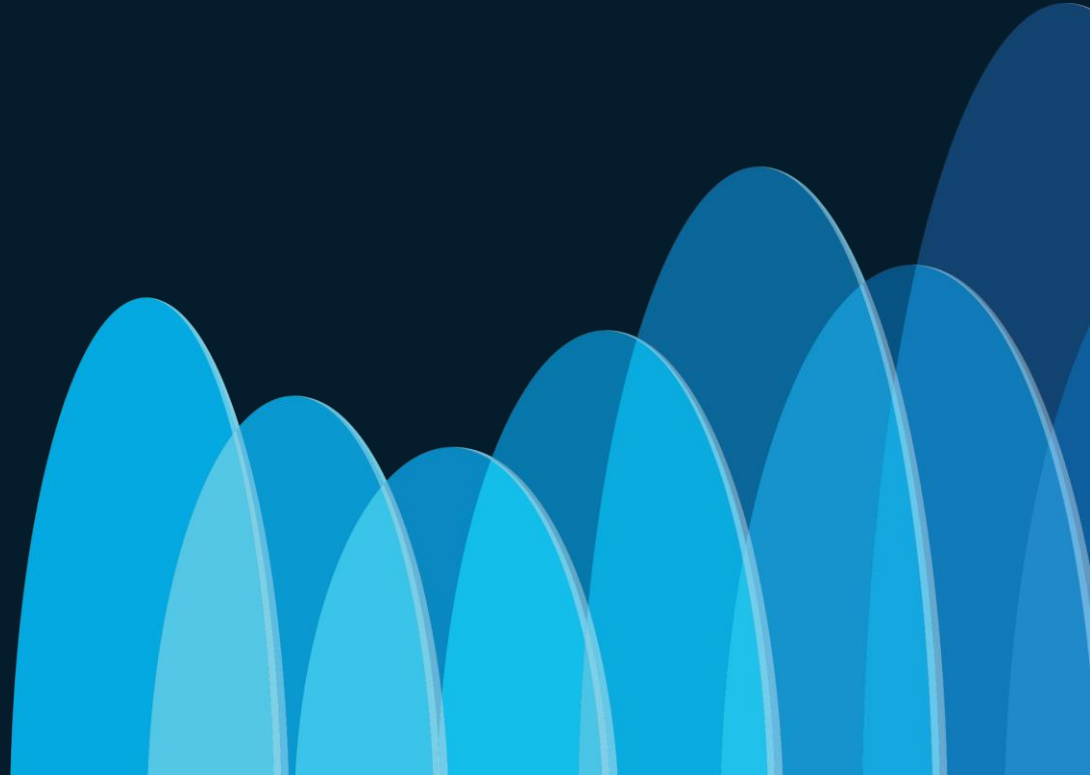
Fully supported stack including Cisco and 3rd party components

AI Advisor tool for configuration guidance

Cisco AI-Ready Infrastructure Stacks



Conclusion



Key Takeaways



Understand your AI requirements

- Identify AI projects and key stakeholders
- Determine AI workload types and likely deployment locations
- Understand infrastructure requirements for AI projects



Define your AI Infrastructure Strategy

- Don't get distracted by the latest shiny object
- Ensure Infrastructure strategy is correctly mapped to real AI requirements
- “Evolution not Revolution” – align AI initiatives with existing investments



Accelerate AI Initiatives

- “Standardize and De-risk” – streamline AI deployments with validated reference architectures
- Leverage Cisco's experience
- Visit us at the Cisco Booth

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.



Continue your education

- Visit the Cisco Stand for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me via WebEx/Email:
cgascoig@cisco.com



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC

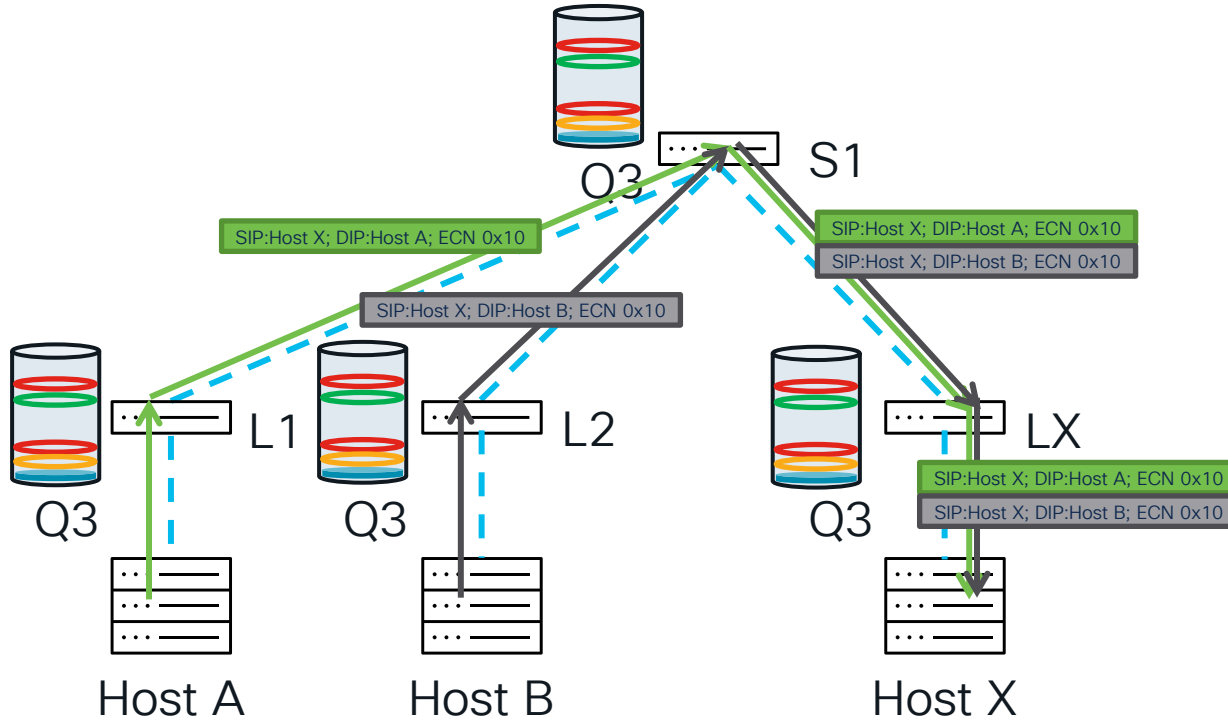
Appendix – Reference slides



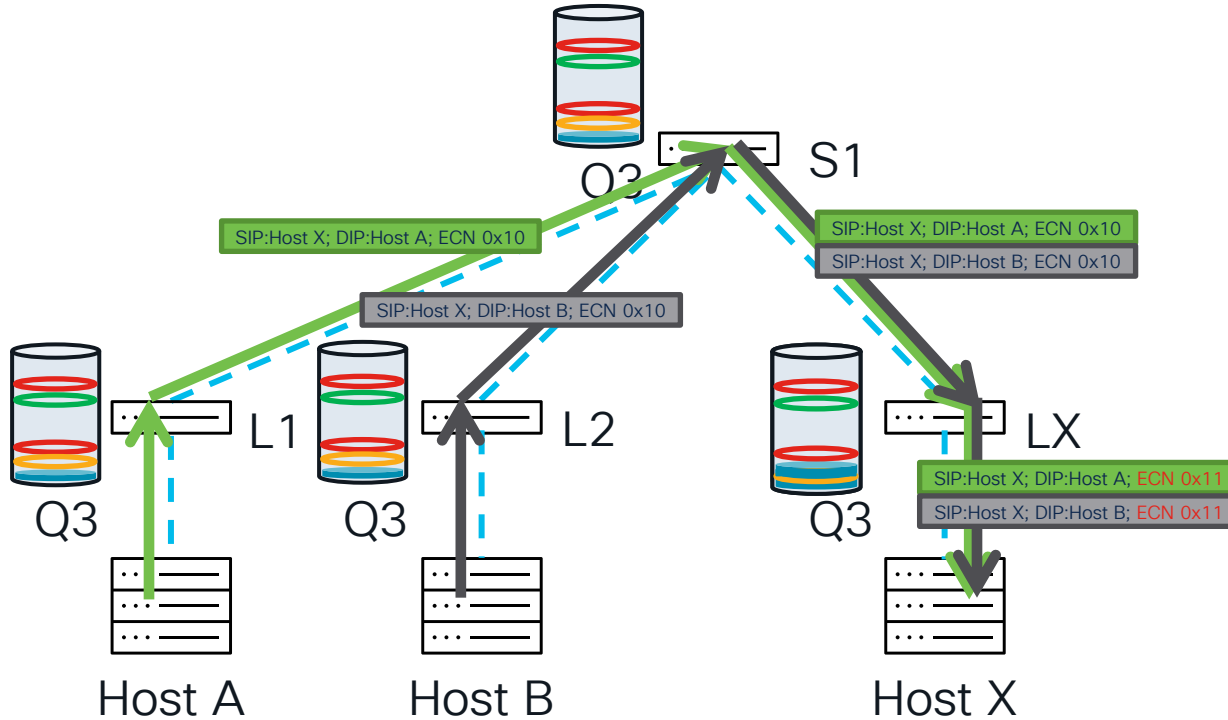
Congestion Management Needs to Work As a System



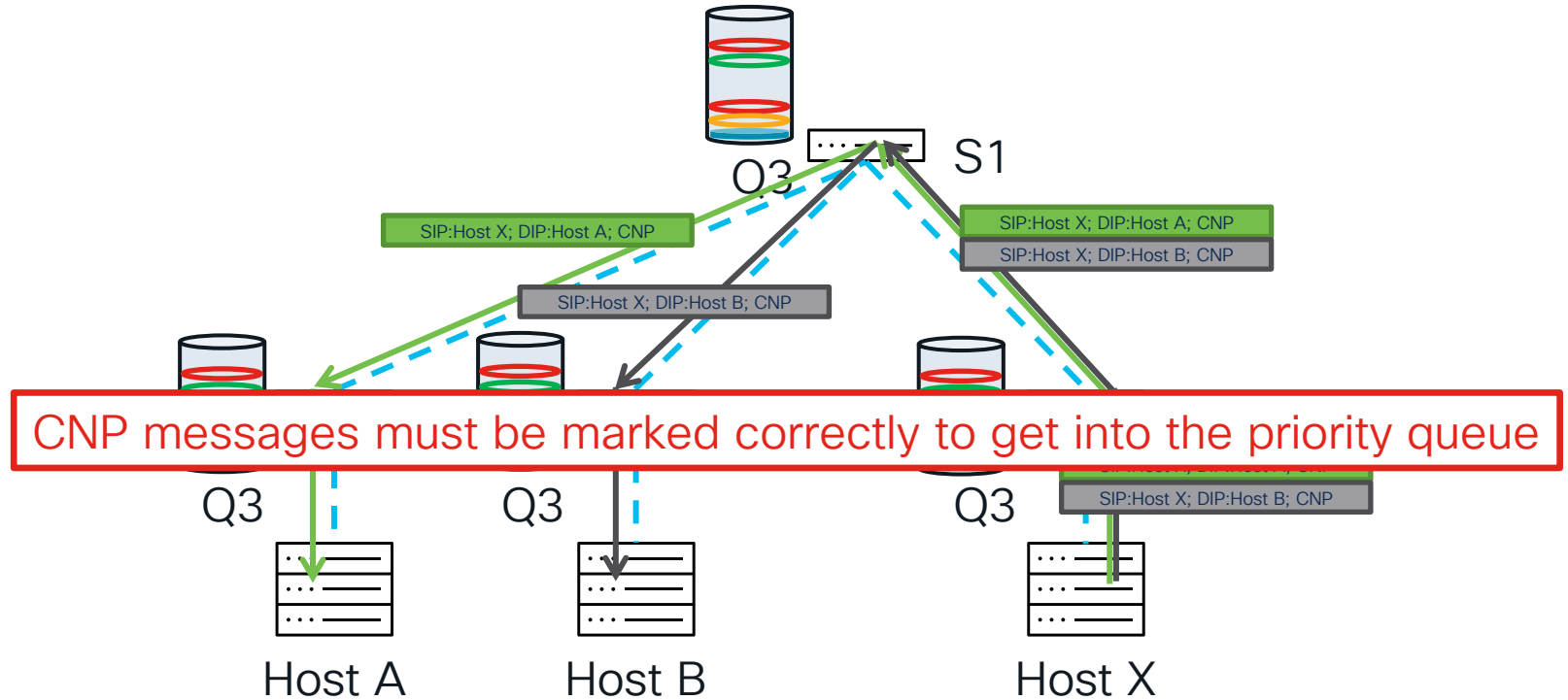
No Congestion Example



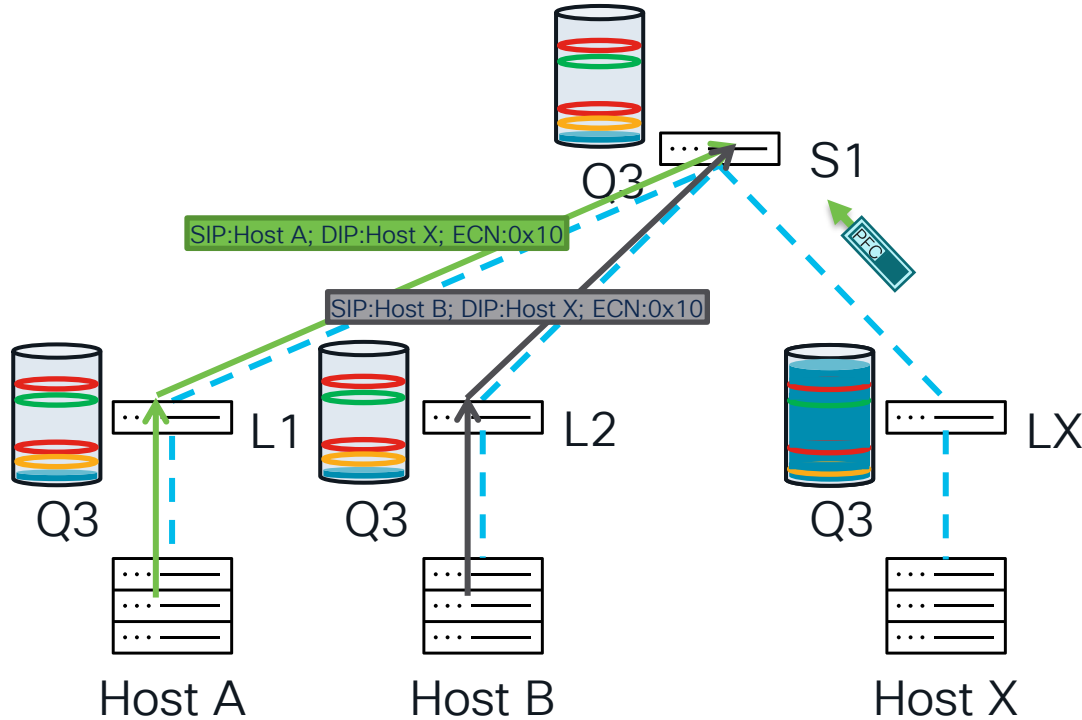
Light Congestion Example



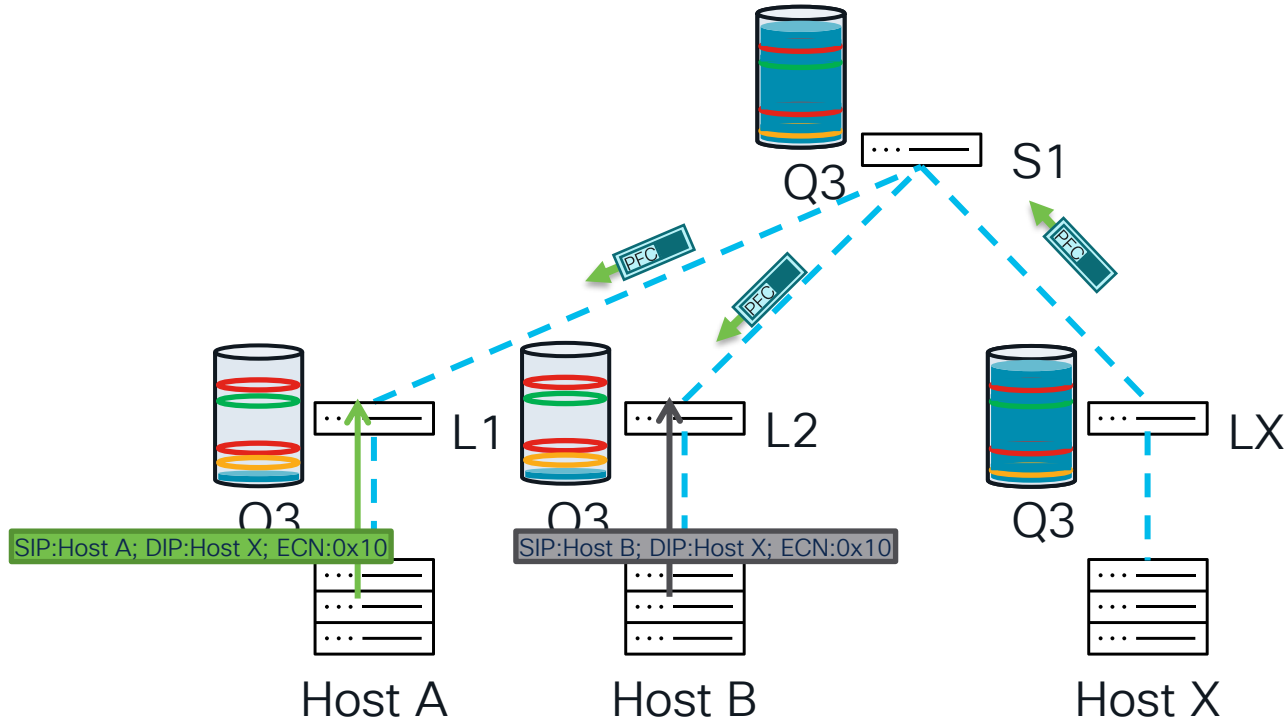
Light Congestion Example -CNP Message



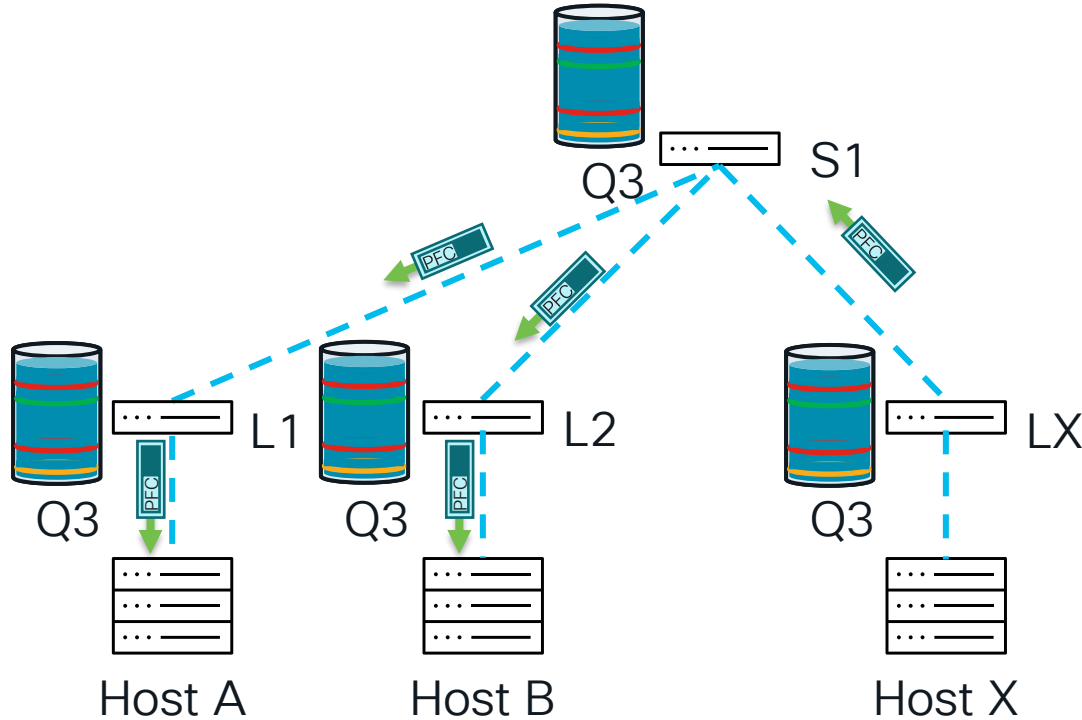
PFC Congestion Leaf



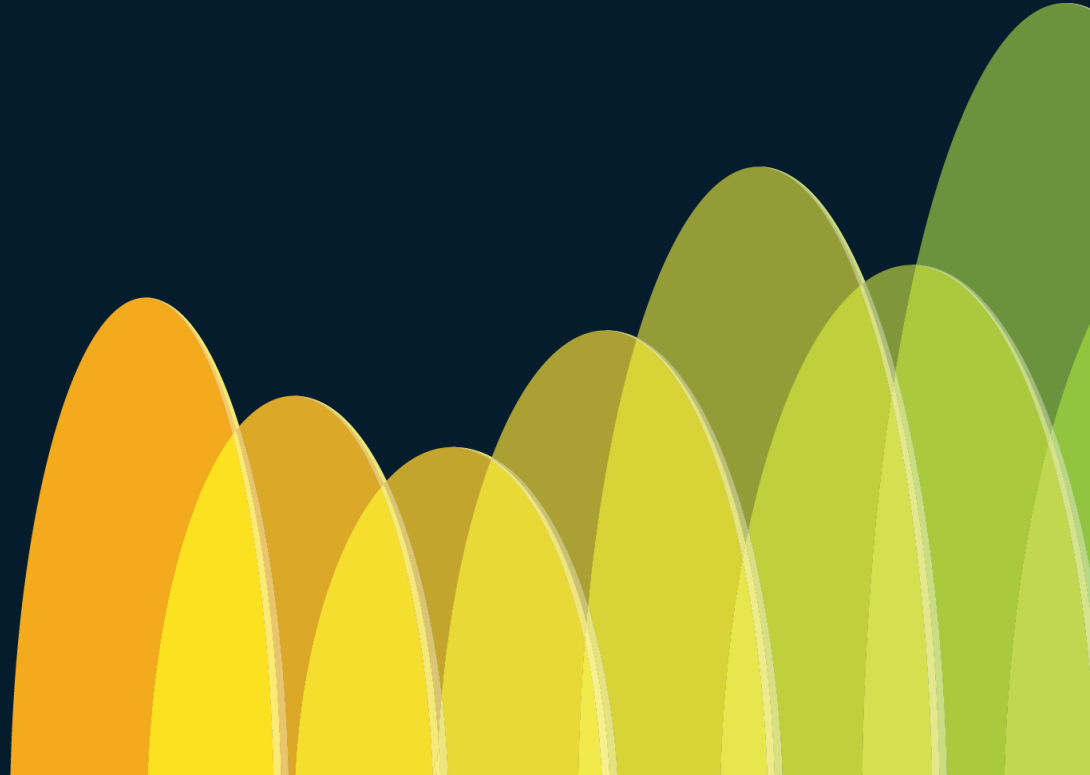
PFC Congestion Leaf + Spine



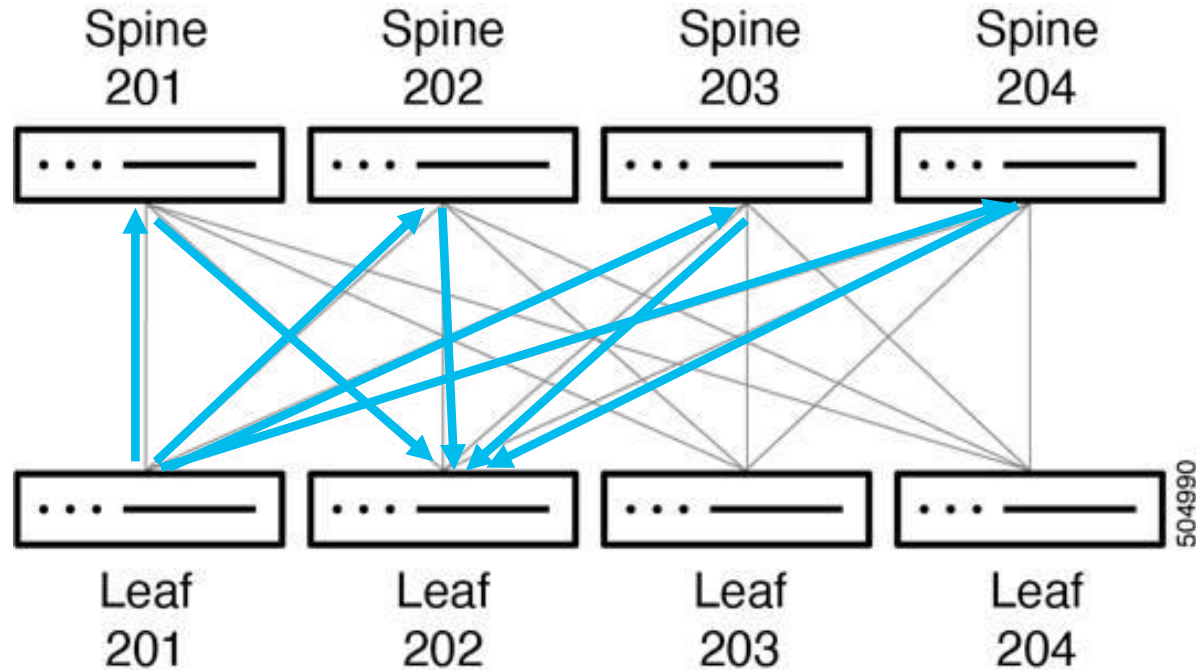
PFC Congestion Leaf + Spine + Leaf



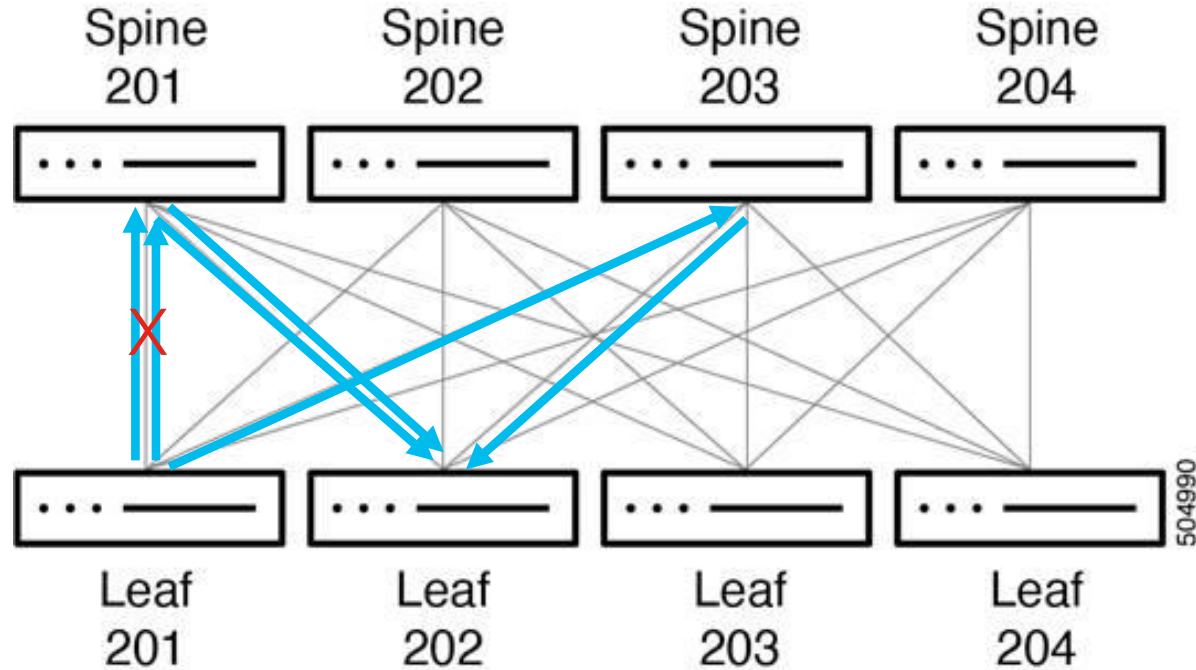
ECMP Hashing



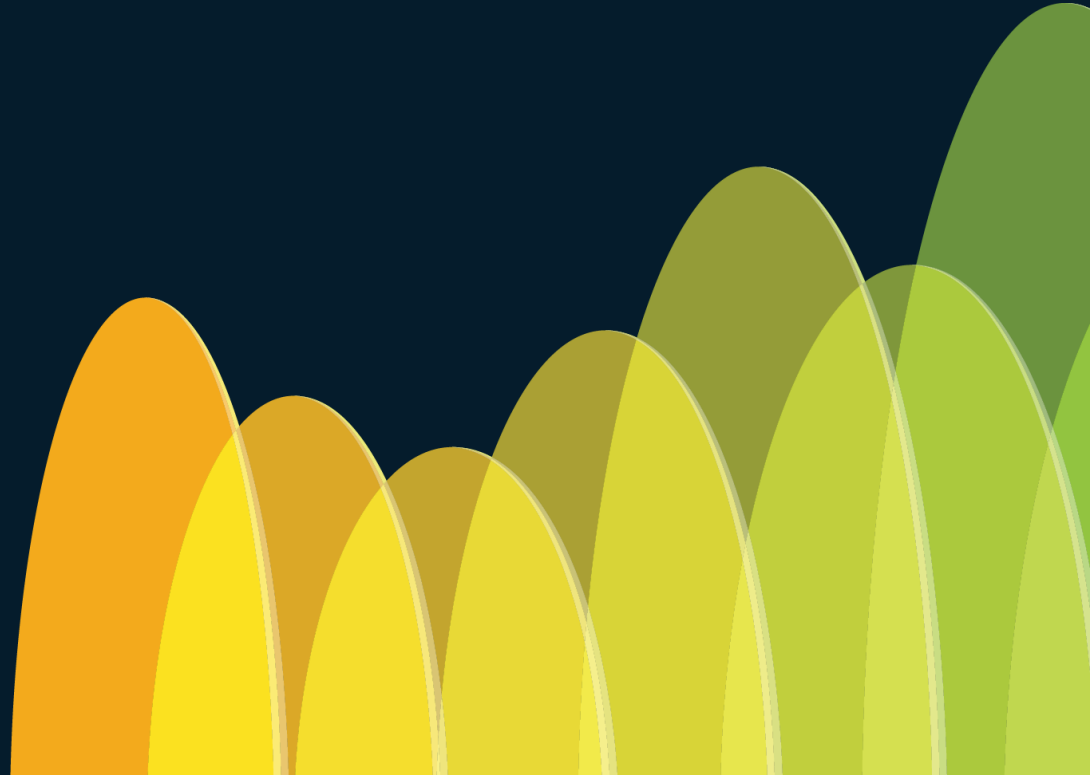
ECMP, Telemetry Assisted Ethernet and Fully Scheduled Fabrics



ECMP, Telemetry Assisted Ethernet and Fully Scheduled Fabrics



How About Pinning?



Default ECMP algorithm

By default, ECMP looks at source and destination IP and Layer 4 ports

Hosts in an AI fabric may belong to uniform subnets

Layer 4 ports, destination port is 4791 for RoCEv2

Entropy comes from Layer 4 Source Port

```
N9K-switch# show ip load-sharing
```

```
IPv4/IPv6 ECMP load sharing:
```

```
Universal-id (Random Seed): 2467474893
```

```
Load-share mode : address source-destination port source-destination
```

```
Rotate: 32
```

User Defined Field (UDF) ECMP algorithm

UDF ECMP looks at source and destination IP and User Defined Field in a packet

Users can choose what field to use to enhance entropy

Every RoCE conversation is identified by Destination Queue Pair, in the IB header

Destination Queue Pair is a 3-byte field

ip load-sharing address source-destination udf offset 33 length 24

```
N9K-switch# show ip load-sharing
```

```
IPv4/IPv6 ECMP load sharing:
```

```
Universal-id (Random Seed): 908907021
```

```
Load-share mode : address source-destination udf offset 33 length 24
```

```
Rotate: 32
```

User Defined Field (UDF) – Destination Q Pair

UDF offset of Nexus 9000 switches starts from the IP header

Destination Q Pair is 33 bytes from the beginning of IP field
(IP (20B) + UDP (8B) + IB (5B)) or 6th byte in the InfiniBand header

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.103.11	172.16.101.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000a19
2	0.000020260	172.16.103.11	172.16.101.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000a19
5	0.116549167	172.16.103.11	172.16.104.11	RRoCE	78	RC RDMA Read Request QP=0x000a00
6	0.116561041	172.16.103.11	172.16.104.11	RRoCE	78	RC RDMA Read Request QP=0x000a00
7	0.122176963	172.16.103.11	172.16.112.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000903
8	0.122185013	172.16.103.11	172.16.112.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000903
9	0.236039751	172.16.103.11	172.16.101.11	RRoCE	78	RC RDMA Read Request QP=0x000a19
10	0.236050373	172.16.103.11	172.16.101.11	RRoCE	78	RC RDMA Read Request QP=0x000a19

> Frame 7: 1086 bytes on wire (8688 bits), 1086 bytes captured (8688 bits) on interface	0030	ff ff 00 00 09 03 00 14	72 81 f5 e
> Ethernet II, Src: MellanoxTech_c4:7c:ab (b8:ce:f6:c4:7c:ab), Dst: Cisco_23:d3:95	0040	64 3d 50 c3 60 3d db 77	94 3d 05 e
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 103	0050	1b 3d fd f2 da 3c c7 a1	87 3d 78 t
> Internet Protocol Version 4, Src: 172.16.103.11, Dst: 172.16.112.11	0060	a2 3c 4f 94 47 3d 00 24	f4 3a bc 5
> User Datagram Protocol, Src Port: 49152, Dst Port: 4791	0070	8c 3d 89 88 16 3d 9d c4	a0 3c b5 c
> InfiniBand	0080	57 3d a3 14 9d 3d 39 76	85 3d a5 2
> Base Transport Header	0090	a2 3d 43 f0 96 3d a9 a5	96 3d f2 e
> Opcode: Reliable Connection (RC) – RDMA READ response Middle (14)	00a0	0c 3b 45 49 11 3d 61 df	82 3d 6b 7
> .0... = Solicited Event: False	00b0	1a 3d b5 8e 38 3d 25 33	ed 3c bd t
> .1... = MigReq: True	00c0	57 3d 6d 44 55 3d 80 12	da 3a 90 3
> .00... = Pad Count: 0	00d0	ad 3c 5d 92 3d 3d 2b 2a	22 3c fd 3
> .0000 = Header Version: 0	00e0	31 3d 60 ca 70 3d 1f 68	6d 3d 6d 5
> Partition Key: 65535	00f0	10 3d 65 e8 aa 3c e4 4f	75 3d 4b e
> Reserved: 00	0100	81 3d 5c 15 a5 3d b8 1e	82 3d 22 3
> Destination Queue Pair: 0x000903	0110	0f 3d 88 9e 1a 3d 65 97	7f 3c 29 c
> .0... = Acknowledge Request: False	0120	8d 3d a7 8c 41 3d 58 48	ff 3c 07 5
> .000000 = Reserved (7 bits): 0	0130	17 3d 5b 72 64 3c d5 b7	d7 3a 71 5
> Packet Sequence Number: 1340033	0140	a1 3d d9 f9 45 3d eb 6a	1c 3c 03 3
> Invariant CRC: 0xb5baee4d	0150	06 3d 73 c4 60 3d 1b 8d	a3 3d 50 4
> Data (1024 bytes)	0160	7a 3c a0 6e e6 3b a0 16	6d 3c b9 e
	0170	28 3d ec b9 5e 3d eb f6	28 3c a3 e
	0180	13 3d 4b 93 86 3d 85 28	c4 3c 6b e
	0190	53 3a 53 c9 14 3c 6c fa	51 3d 64 2
	01a0	91 3d 14 44 69 3d f2 8e	90 3d 0b 1
	01b0	5f 3d 91 cb 97 3d 43 51	ba 3c e5 c

User Defined Field (UDF) – Destination Q Pair

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.103.11	172.16.101.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000a19
2	0.000020260	172.16.103.11	172.16.101.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000a19
5	0.116549167	172.16.103.11	172.16.104.11	RRoCE	78	RC RDMA Read Request QP=0x000a00
6	0.116561041	172.16.103.11	172.16.104.11	RRoCE	78	RC RDMA Read Request QP=0x000a00
7	0.122176963	172.16.103.11	172.16.112.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000903
8	0.122185013	172.16.103.11	172.16.112.11	RRoCE	1086	RC RDMA Read Response Middle QP=0x000903
9	0.236039751	172.16.103.11	172.16.101.11	RRoCE	78	RC RDMA Read Request QP=0x000a19
10	0.236050373	172.16.103.11	172.16.101.11	RRoCE	78	RC RDMA Read Request QP=0x000a19

> Frame 9: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface ps-inb, id 0

> Ethernet II, Src: MellanoxTech_c4:7c:ab (b8:ce:f6:c4:7c:ab), Dst: Cisco_23:d3:95 (00:fc:ba:23:d3:95)

> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 103

> Internet Protocol Version 4, Src: 172.16.103.11, Dst: 172.16.101.11

> User Datagram Protocol, Src Port: 49152, Dst Port: 4791

> InfiniBand

Base Transport Header

Opcode: Reliable Connection (RC) – RDMA READ Request (12)

0... = Solicited Event: False

.1.. = MigReq: True

..00 = Pad Count: 0

... 0000 = Header Version: 0

Partition Key: 65535

Reserved: 00

Destination Queue Pair: 0x000a19

1... = Acknowledge Request: True

.000 0000 = Reserved (7 bits): 0

Packet Sequence Number: 1365501

RETH – RDMA Extended Transport Header

Invariant CRC: 0x4f0ee094

0000 00 fc ba 23 d3 95 b8 ce f6 c4 7c ab 81 00 00 67

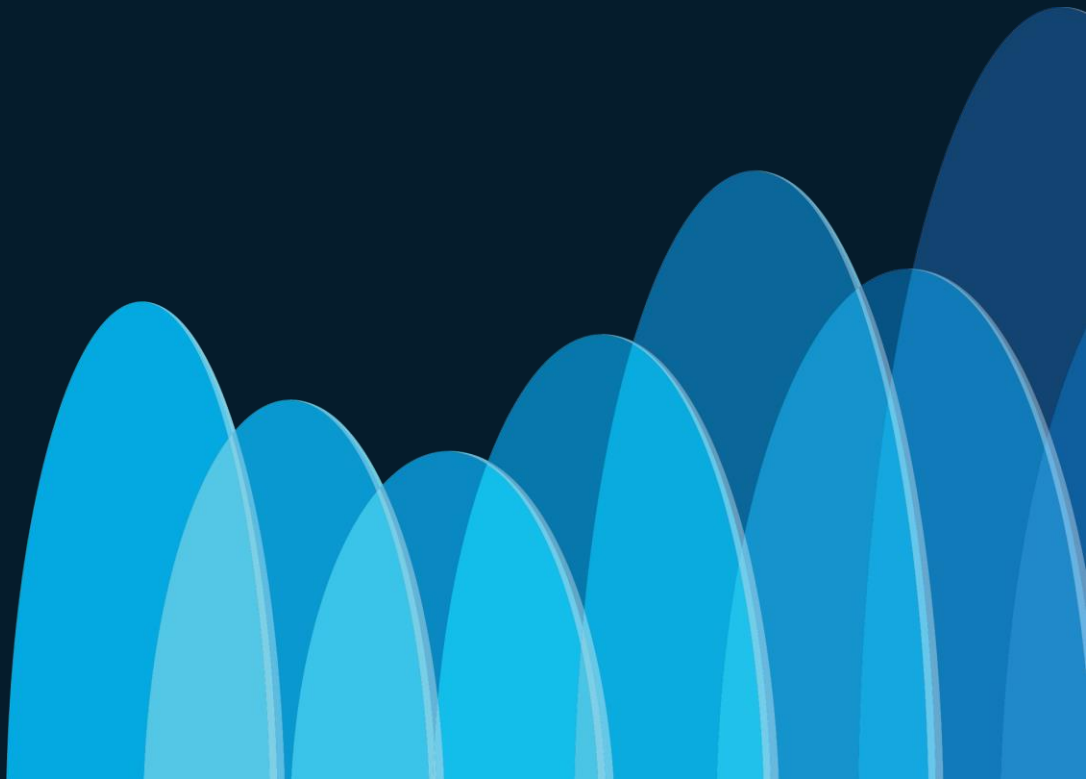
0010 08 00 45 6a 00 3c 90 3b 40 00 ff 11 c6 d3 ac 10

0020 67 0b ac 10 65 0b c0 00 12 b7 00 28 00 00 0c 40

0030 ff ff 00 00 0a 19 80 14 d5 fd 00 00 00 0e 83

0040 38 c0 00 12 5c 52 00 00 03 40 4f 0e e0 94

NVIDIA Rail- Optimised Design



Some Things I Learned at NVIDIA GTC

The screenshot shows the NVIDIA GTC 2024 attendee portal. The browser address bar displays the URL: <https://register.nvidia.com/flow/nvidia/gtcs24/attendeeportal/page/sessioncatalog?search=S62129>. The top navigation bar includes a search bar and a 'Log Out' button. Below the navigation bar, a banner displays the event schedule: 'Workshops March 17-21 | AI Conference and Expo March 18-21 | Keynote March 18 | San Jose, CA and Virtual'. The main content area shows a search for 'S62129' resulting in '1 sessions'. On the left, a sidebar lists filter categories: 'My Interests', 'Session Format', 'Industry', 'Session Type', 'Audience Type', 'Audience Level', 'Session Spoken Language', and 'Products & Technologies'. The session card for 'S62129' is displayed, featuring tabs for 'Talks & Panels', 'Live Stream', and 'In-Person'. The session title is 'Training Deep Learning Models at Scale: How NCCL Enables Best Performance on AI Data Center Networks [S62129]'. The speaker is 'Sylvain Jeaugey, Principal Engineer, NVIDIA'. The description states: 'Discover how NCCL uses every capability of all DGX and HGX platforms to accelerate inter-GPU communication and allow deep learning training to scale further. See how Grace Hopper platforms can leverage multi-node NVLink to compute in parallel at...'. A 'Replay' button is visible. At the bottom of the session card, there is a 'Show More' link with a green checkmark.

Workshops March 17-21 | AI Conference and Expo March 18-21 | Keynote March 18 | San Jose, CA and Virtual

Log Out

Search 1 sessions

☐ My Interests

Session Format

Industry

Session Type

Audience Type

Audience Level

Session Spoken Language

Products & Technologies

Talks & Panels Live Stream In-Person

Training Deep Learning Models at Scale: How NCCL Enables Best Performance on AI Data Center Networks [S62129]

Sylvain Jeaugey, Principal Engineer, NVIDIA

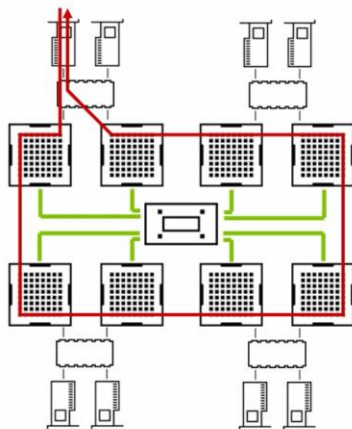
Discover how NCCL uses every capability of all DGX and HGX platforms to accelerate inter-GPU communication and allow deep learning training to scale further. See how Grace Hopper platforms can leverage multi-node NVLink to compute in parallel at...

Replay

Show More

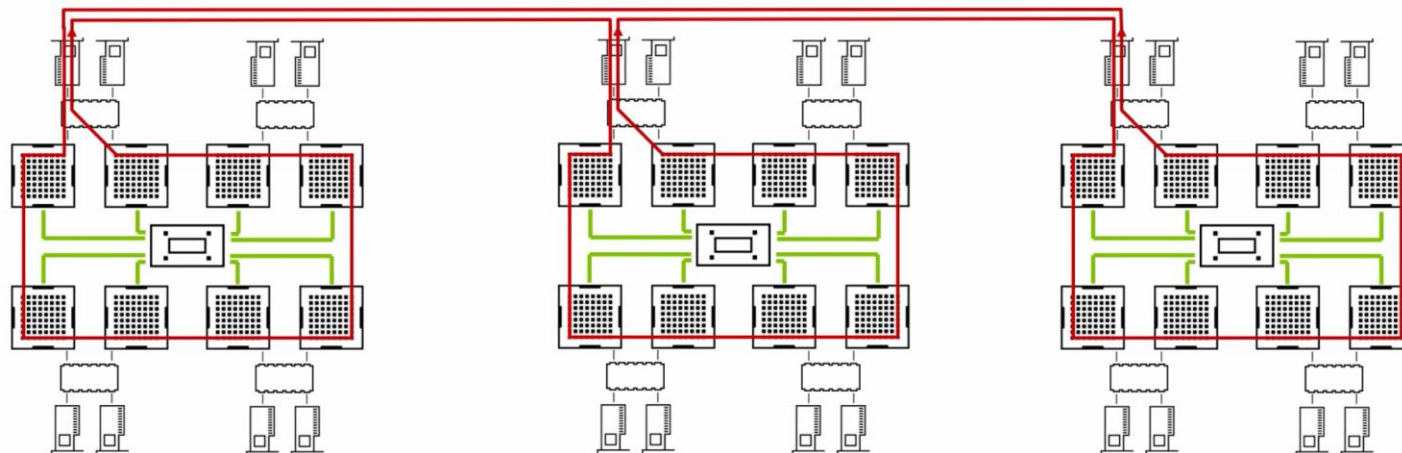


Rings on DGX A100

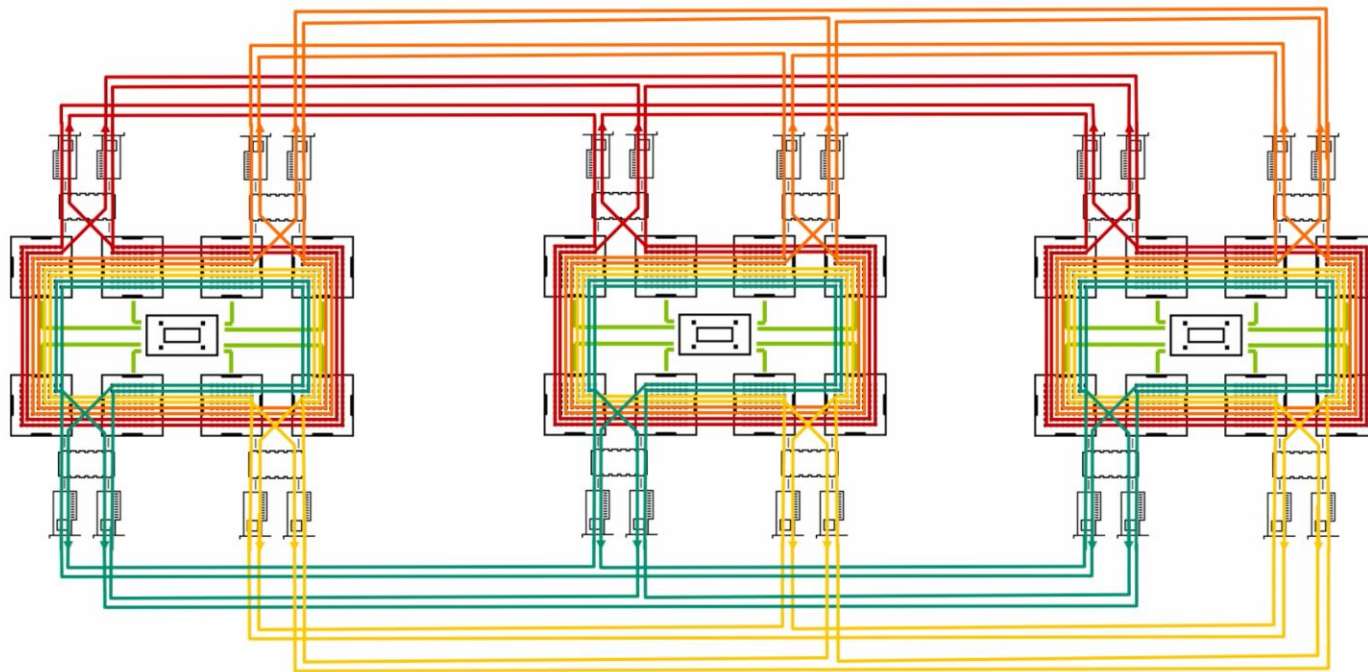




Rings on DGX A100



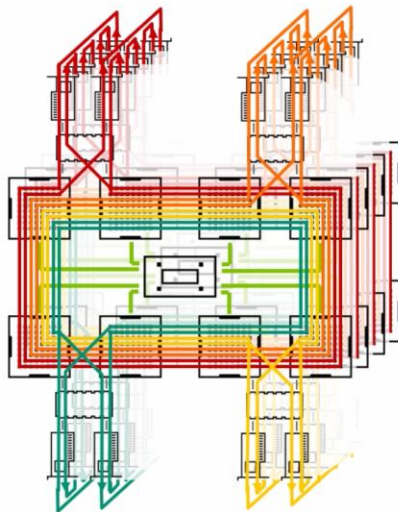
Rings on DGX A100





Inter-node Communication

Rail-optimized design

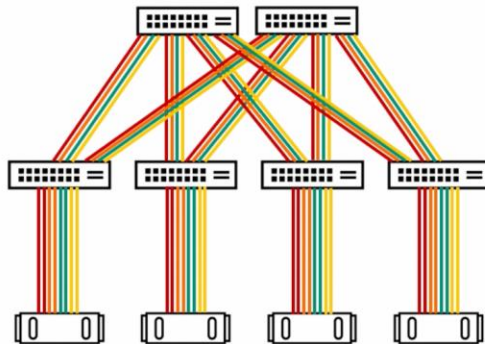
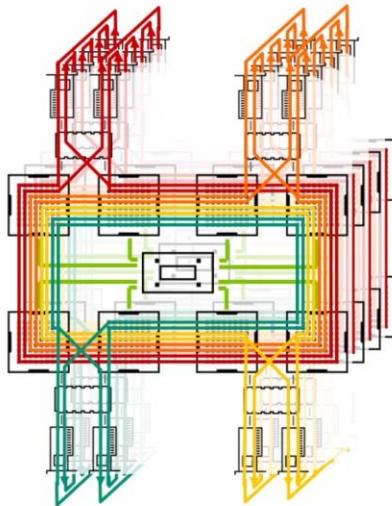


Inter-node Communication

Rail-optimized design

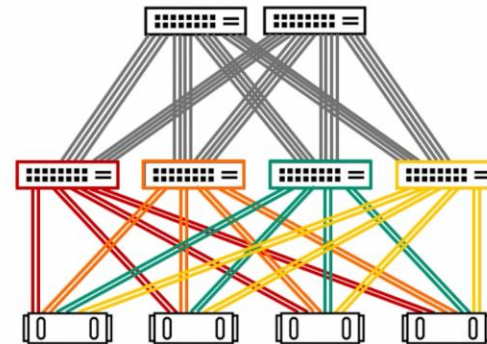
Classic fabric design

8 flows go across multiple level of switches.
Each flow needs to use a different path.



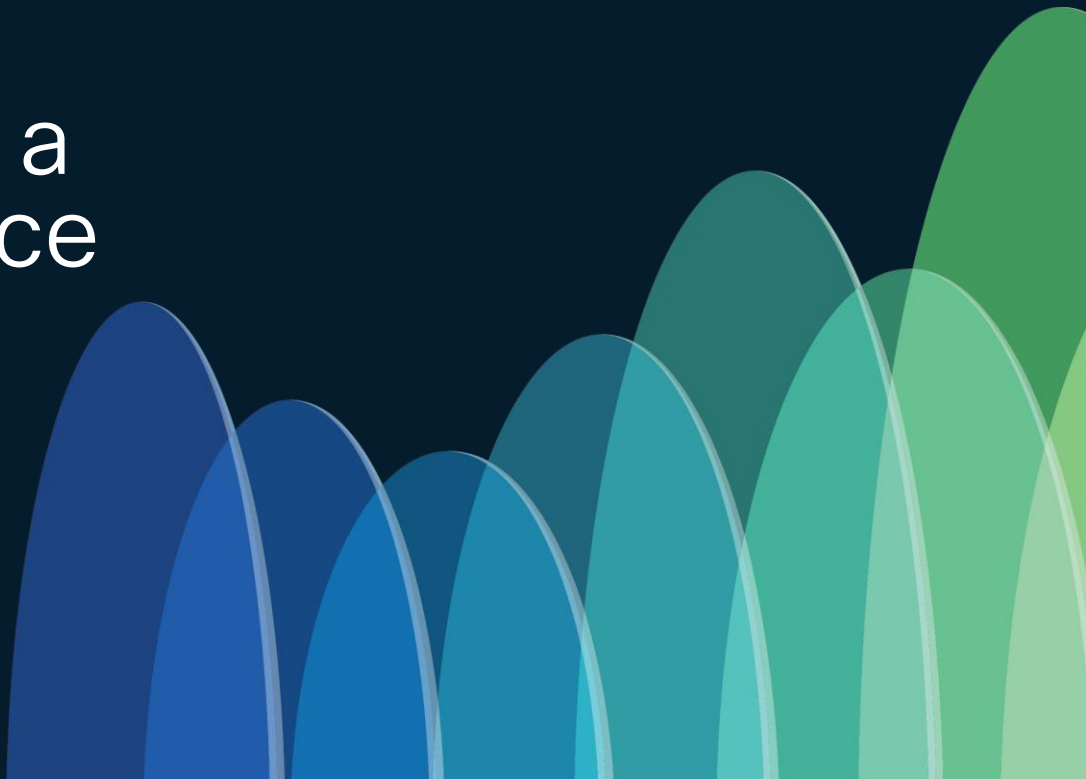
Rail-optimized design

Flows are physically separated.
Routing collisions cannot occur.



Remember about rail
optimised designs when
building large NVIDIA
GPU clusters

Tips for Building a High-Performance AI Fabric



*To get RoCEv2 to work
as expected it is critical
to configure the NICs
and OS correctly*

Configure Your Ethernet NICs

Mellanox ConnectX-6 NIC
2 Port 100G Ethernet NIC

cat /etc/netplan/00-installer-config.yaml

```
<CLIP>
ens1f0np0:
  dhcp4: false
ens1f1np1:
  dhcp4: false
vlands:
  ens1f0np0.101:
    id: 101
    link: ens1f0np0
    addresses:
      - 192.168.101.11/24
    routes:
      - to: 192.168.0.0/16
        via: 192.168.101.1
        metric: 100
  ens1f1np1.101:
    id: 101
    link: ens1f1np1
    addresses:
      - 172.16.101.11/24
    routes:
      - to: 172.16.0.0/16
        via: 172.16.101.1
        metric: 100
```

VLAN Config Required for PFC

VLAN Encap is 101

Configure the “marking” Ubuntu Example

```
sudo apt-get install vlan
```

```
sudo bash -c 'echo 1 > /sys/class/net/ens1f0np0/ecn/roce_np/enable/3'
sudo bash -c 'echo 1 > /sys/class/net/ens1f1np1/ecn/roce_np/enable/3'
```



Enable ECN for class 3

```
sudo bash -c 'echo 6 > /sys/class/net/ens1f0np0/ecn/roce_np/cnp_802p_prio'
sudo bash -c 'echo 6 > /sys/class/net/ens1f1np1/ecn/roce_np/cnp_802p_prio'
```



Enable CNP L2 egress priority to 6,
“enable CNP”

```
sudo bash -c 'echo 48 > /sys/class/net/ens1f0np0/ecn/roce_np/cnp_dscp'
sudo bash -c 'echo 48 > /sys/class/net/ens1f1np1/ecn/roce_np/cnp_dscp'
```



Map CNP to DSCP 48

```
sudo mlx_qos -i ens1f0np0 --trust=dscp --pfc 0,0,0,1,0,0,0,0
sudo mlx_qos -i ens1f1np1 --trust=dscp --pfc 0,0,0,1,0,0,0,0
```



Map PFC to priority 3

```
sudo sysctl -w net.ipv4.tcp_ecn=1
```



Enable ECN for traffic

```
sudo cma_roce_mode -d mlx5_0 -p 1 -m 2
sudo cma_roce_mode -d mlx5_1 -p 1 -m 2
```



Set mode to RoCEv2 on the NIC

```
sudo cma_roce_tos -d mlx5_0 -t 96
sudo cma_roce_tos -d mlx5_1 -t 96
```



Set TOS to 96 (DSCP CS3 or 24
Dec) for RoCEv2 traffic

```
sudo vconfig set_egress_map ens1f0np0.101 4 3
sudo vconfig set_egress_map ens1f1np1.101 4 3
```



Egress map RoCE traffic to priority 3

Recommendations for Server Tuning*

*These recommendations were
done in conjunction with
[Vijay Durairaj](#)

System Tuning Recommendations

BIOS tunings:

- Processor Configuration:

Intel Virtualisation Technology : <Enabled>

Intel Hyper-Threading Technology : <Enabled>

CPU Performance : <HPC>

- Power & Performance Configuration:

Processor C1E : <Disabled>

Processor C6 : <Enabled>

Package C-State control : <C0/C1 State>

Uncore Frequency Scaling: <Enabled>

Thermal : Fan Control Policy : <Maximum Power>

PCIe : Width, Speed, Max Payload Size, Max Read Request

OS tunings : CPU Power, TCP Parameters & Memory Settings



C1E is one of the automatic power-saving features which is triggered when the system is idle. **Best to disable it when overclocking**

C6 BIOS will automatically disable CPU core and cache for power saving

In C0/C1 state **all of the cores are locked at maximum performance and will cause a higher consumption of power**

Uncore Frequency Scaling “enabled” stops the processor from dynamically changing frequencies based on workload to save power

BIOS Tuning – CPU Performance (HPC)

The screenshot displays the Cisco Intersight interface for BIOS tuning. The top navigation bar includes the Cisco logo, 'Intersight', and 'Infrastructure Service'. A search bar is present. The main content area is titled 'Edit' and shows the 'Policy Details' for 'ucs-m7-bios_NVCERT'. The left sidebar lists various system components. The main panel shows several BIOS settings with dropdown menus. The 'CPU Performance' dropdown is set to 'hpc'. The 'Intel Dynamic Speed Select' dropdown is set to 'platform-default'. The 'Intel Speed Select' dropdown is set to 'platform-default'. The 'Intel(R) VT' dropdown is set to 'disabled', with a red box highlighting the 'enabled' option. The 'Intel HyperThreading Tech' dropdown is set to 'platform-default'. The 'Intel Turbo Boost Tech' dropdown is set to 'platform-default'. The 'I/O Error Enable' dropdown is set to 'platform-default'. The 'KTI Prefetch' dropdown is set to 'platform-default'.

Setting	Value
Frequency Floor Override	platform-default
CPU Performance	hpc
Power Technology	platform-default
Intel Dynamic Speed Select	platform-default
Intel Speed Select	platform-default
Intel(R) VT	disabled
Intel HyperThreading Tech	platform-default
Intel Turbo Boost Tech	platform-default
I/O Error Enable	platform-default
KTI Prefetch	platform-default

BIOS Tuning – Processor Power States

The screenshot displays the Cisco Intersight web interface for BIOS tuning. The top navigation bar includes the Cisco logo, 'Intersight', 'Infrastructure Service', and a search bar. The breadcrumb trail shows 'Policies > BIOS > ucs-m7-bios_NVCERT'. The left sidebar contains navigation links: Overview, Operate, Servers, Chassis, Fabric Interconnects, HyperFlex Clusters, Integrated Systems, Analyze, Explorer (with a 'New' button), and Configure. The main content area is titled 'Edit' and shows the 'Policy Details' tab for the 'General' section. Two settings are highlighted with red boxes: 'C0 C1 State' and 'Processor C1E'. The 'C0 C1 State' dropdown menu is open, showing 'platform-default', 'Auto', and 'C0 C1 State' (selected). The 'Processor C1E' dropdown menu is also open, showing 'disabled', 'platform-default', and 'enabled' (selected). Other visible settings include 'Patrol Scrub' (platform-default), 'PRMRR Size' (platform-default), 'Patrol Scrub Interval *' (platform-default), 'Processor C3 Report' (disabled), and 'CPU C State' (platform-default).

Overview

Policies > BIOS > ucs-m7-bios_NVCERT

Edit

General

Policy Details

Package C State Limit

C0 C1 State

platform-default

Auto

C0 C1 State

Patrol Scrub

platform-default

PRMRR Size

platform-default

Patrol Scrub Interval *

platform-default

Processor C1E

disabled

platform-default

enabled

disabled

CPU C State

platform-default

Thermal – Fan control Policy

The screenshot shows the Cisco Intersight Infrastructure Service interface. The top navigation bar includes the Cisco logo, 'Intersight', 'Infrastructure Service', and a search bar. The left sidebar contains a navigation menu with categories like Overview, Operate, Servers, Chassis, Fabric Interconnects, HyperFlex Clusters, Integrated Systems, Analyze, Explorer, Configure, Profiles, and Templates. The main content area is titled 'Edit' and shows the 'Policies > Thermal > ucsx-thermal' path. The 'Policy Details' tab is selected, displaying a dropdown for 'All Platforms' and a 'Fan Control' section. The 'Fan Control Mode' dropdown is open, showing options: Balanced, Low Power, High Power, Maximum Power (highlighted with a red box), and Acoustic.

Overview

Policies > Thermal > ucsx-thermal

Edit

General

2 Policy Details

Policy Details

Add policy details

All Platforms | UCS Server (Standalone) | UCS Server (FI-Attached) | UCS Chassis

Fan Control

Fan Control Mode

Maximum Power

- Balanced
- Low Power
- High Power
- Maximum Power**
- Acoustic

OS Tuning – 1

- *Mellanox Automatic tuning for Performance*

```
# mlx_tune -p HIGH_THROUGHPUT
```

- *Disable the TCP Timestamps for better CPU Utilisation*

```
# sysctl -w net.ipv4.tcp_timestamps=0
```

- *Enable the TCP selective acks option for better throughput:*

```
# sysctl -w net.ipv4.tcp_sack=1
```

- *Increase the maximum length of processor input queues:*

```
# sysctl -w net.core.netdev_max_backlog=300000
```

- *Increase the TCP maximum and default buffer sizes using setsockopt():*

```
# sysctl -w net.core.rmem_max=4194304
```

```
# sysctl -w net.core.wmem_max=4194304
```

```
# sysctl -w net.core.rmem_default=4194304
```

```
# sysctl -w net.core.wmem_default=4194304
```

```
# sysctl -w net.core.optmem_max=4194304
```



Part of the `mlx_tune` utility. This command does all the NIC optimisation recommended by Mellanox



Linux

The rest of these are recommendations by Mellanox for “Tuning the Network Adapter for Improved IPv4 Traffic Performance”

<https://mellanox.my.site.com/mellanoxcommunity/s/article/linux-sysctl-tuning>

OS Tuning – 2

- *Increase memory thresholds to prevent packet dropping:*

```
# sysctl -w net.ipv4.tcp_rmem="4096 87380 4194304"
```

```
# sysctl -w net.ipv4.tcp_wmem="4096 65536 4194304"
```

- *Enable low latency mode for TCP:*

```
# sysctl -w net.ipv4.tcp_low_latency=1
```

```
# sysctl -w net.ipv4.tcp_adv_win_scale=1
```

- *Disable OS watchdog & Numa balancing:*

```
# echo '0' > /proc/sys/kernel/nmi_watchdog'
```

```
# echo 0 > /proc/sys/kernel/numa_balancing
```

- *Fine tuning Kernel task scheduler :*

```
# echo 100000000 > /proc/sys/kernel/sched_min_granularity_ns
```

```
# echo 50000000 > /proc/sys/kernel/sched_migration_cost_ns
```

These are recommendations by Mellanox for “Tuning the Network Adapter for Improved IPv4 Traffic Performance”

<https://mellanox.my.site.com/mellanoxcommunity/s/article/linux-sysctl-tuning>



Linux

Set to maximum acceptable values in nanoseconds (1 sec and 1/2 sec)



PCIe Tuning for the Best Performance

Width | Speed | Max Payload Size | Max Read Request

```
[root@ucs19 ~]# lspci | grep Mell
5e:00.0 Ethernet controller: Mellanox Technologies MT28908 Family [ConnectX-6]
5e:00.1 Ethernet controller: Mellanox Technologies MT28908 Family [ConnectX-6]
[root@ucs19 ~]# lspci -s 5e:00.0 -vvv | grep PCIeGen
[V0] Vendor specific: PCIeGen4 x16
[root@ucs19 ~]# lspci -s 5e:00.0 -vvv | grep Speed
LnkCap: Port #0, Speed 16GT/s, Width x16, ASPM not supported, B
, L1 unlimited
LnkSta: Speed 8GT/s, Width x16, TrErr- Train- SlotClk+ DLActive
LnkCtl2: Target Link Speed: 16GT/s, EnterCompliance- SpeedDis-
[root@ucs19 ~]# lspci -s 5e:00.0 -vvv | grep MaxReadReq
MaxPayload 256 bytes, MaxReadReq 512 bytes
[root@ucs19 ~]# lspci -s 5e:00.0 -vvv | grep DevCtl: -C 2
DevCap: MaxPayload 512 bytes, PhantFunc 0, Latency L0s unlimite
ExtTag+ AttnBttn- AttnInd- PwrInd- RBE+ FLReset+ SlotPow
DevCtl: Report errors: Correctable- Non-Fatal- Fatal- Unsupport
RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+ FLReset-
MaxPayload 256 bytes, MaxReadReq 512 bytes
[root@ucs19 ~]#
```

List PCI command to find out where the Mellanox NICs are connected

List PCI command to show location 5e:00.0 settings. Speed determines the number of PCIe transactions possible. The speed is measured in GT/s which stands for "billion transactions per second"

The PCIe Max Payload Size determines the maximum size of a PCIe packet, or PCIe MTU (similar to networking protocols). This parameter is set only by the system and depends on the chipset architecture (e.g. x86_64, Power8, ARM, etc).

PCIe Tuning for the Best Performance

Tuned : Width | Speed | Max Payload Size | Max Read Request

```
[root@ucs19 ~]# lspci -s 5e:00.0 -vvv | grep DevCtl: -C 2
DevCap: MaxPayload 512 bytes, PhantFunc 0, Latency L0s unlimited, L1 unlimited
        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset+ SlotPowerLimit 0.000W
DevCtl: Report errors: Correctable- Non-Fatal- Fatal- Unsupported-
        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+ FLReset-
        MaxPayload 256 bytes, MaxReadReq 512 bytes

[root@ucs19 ~]# setpci -s 5e:00.0 68.W=5036
[root@ucs19 ~]# lspci -s 5e:00.0 -vvv | grep DevCtl: -C 2
DevCap: MaxPayload 512 bytes, PhantFunc 0, Latency L0s unlimited, L1 unlimited
        ExtTag+ AttnBtn- AttnInd- PwrInd- RBE+ FLReset+ SlotPowerLimit 0.000W
DevCtl: Report errors: Correctable- Non-Fatal+ Fatal- Unsupported-
        RlxdOrd+ ExtTag+ PhantFunc- AuxPwr- NoSnoop+ FLReset-
        MaxPayload 256 bytes, MaxReadReq 4096 bytes
```

The first digit is the PCIe Max Read Request size selector.

The acceptable values are:
0 - 128B, 1 - 256B, 2 - 512B,
3 - 1024B, 4 - 2048B
and 5 - 4096B.

Ubuntu/Debian: `sudo apt install pciutils`

What does the Network Configuration Look Like?



QOS Configs Classification

CLASS MAP

```
class-map type qos match-all class-q3
  match dscp 24
class-map type qos match-all class-q7
  match dscp 48
```



Put packets with DSCP 24 into
“class-q3”

Put Packets with DSCP 48 into
“class-q7”

POLICY MAP

```
policy-map type qos QOS_classification_policy
  class class-q3
    set qos-group 3
  class class-q7
    set qos-group 7
  class class-default
    set qos-group 0
```



Put packets from class-q3 into
queue 3

Put packets from class-q7 into
queue 7

Everything else goes into default or
queue 0

QOS Configs Queuing and Scheduling

POLICY MAP “type queuing”

policy-map type queuing custom-8q-out-policy

class type queuing c-out-8q-q7

priority level 1

class type queuing c-out-8q-q6

bandwidth remaining percent 0

class type queuing c-out-8q-q5

bandwidth remaining percent 0

class type queuing c-out-8q-q4

bandwidth remaining percent 0

class type queuing c-out-8q-q3

bandwidth remaining percent 60

random-detect minimum-threshold 150 kbytes maximum-threshold 3000 kbytes drop-probability 7 weight 0 ecn



Queues 7 is set up as a priority queue



Queues 3 received 60% of the bandwidth and ECN marking enabled starting at 150KB of buffer utilisation, drop probability of 7%

QOS Configs Queuing and Scheduling

POLICY MAP cont.

```
class type queuing c-out-8q-q2
  bandwidth remaining percent 0
class type queuing c-out-8q-q1
  bandwidth remaining percent 0
class type queuing c-out-8q-q-default
  bandwidth remaining percent 40
```



Queues 1 and 2 receive 0% guaranteed bandwidth

Default queue receives 40%

QOS Configs cont.

POLICY MAP “type networking”

```
policy-map type network-qos custom-8q-nq-policy
<snip>
class type network-qos c-8q-nq3
    mtu 9216
    pause pfc-cos 3
<snip>
```



Use a policy-map of type network-qos to add PFC to queue 3

MTU is used as a way to calculate headroom for the non-drop queue.

SYSTEM QOS

```
system qos
service-policy type network-qos custom-8q-nq-policy
service-policy type queuing output custom-8q-out-policy
```



Attach policy maps of "type queuing" and "type network-qos" system wide.

This will trigger ECN marking and ports configured with PFC will receive and honor those frames, as well as generate pause frames as needed

QOS Configs Interfaces

```
interface Ethernet1/1
service-policy type qos input QOS_classification_policy
priority-flow-control mode on
priority-flow-control watch-dog-interval on
```



Attach the service policy to the interface.

Enable PFC

Enable PFC Watch Dog using a default interval of 100 milliseconds.



Thank you



#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC