Nokia Validated Design

AI-DC Nokia Validated Design - NVIDIA

3HE-21918-AAAA-TQZZA
Issue 2
March 2026

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

Contents

1 Executive summary

Nokia Validated Designs (NVDs) is a workstream dedicated to producing validated recommendations to the consumer about Nokia's portfolio across market segments.

Nokia reviews industry-relevant designs and solutions, using requirement analysis to gather relevant industry designs, forms a prescriptive solution, validates the solutions in our labs, and provides it to the consumer.

After the design has been compiled, Nokia performs an intense array of hardware, software, traffic, and failure tests to form the validated design. The resultant design and collateral provide the consumer with a template that they can use to deploy the solution in their own environment.

NVDs are structured as core and ancillary (extension) designs. This document demonstrates the deployment of network infrastructure for AI clusters, including a backend and collapsed frontend/storage fabric, covering various physical and logical connectivity aspects and associated technologies involved in such a design.

The AI NVD represented in this document focuses on both the training and inference aspects of Artificial Intelligence/Machine Learning (AI/ML) demonstrated via a two-stripe two-tier architecture. It covers additional tenets such as multitenancy and various industry standard benchmark tests to showcase the efficacy of the Nokia AI-DC portfolio.

Find more information about NVDs at:

https://www.nokia.com/ip-networks/validated-designs/

2 AI training and inference clusters

2.1 Introduction to AI/ML

Machine learning is a method by which models, without being explicitly programmed, learn from data by identifying patterns from datasets and making decisions based on those patterns. The AI-DC backend fabric focuses on machine learning, how the workload is distributed, and how the network is utilized while training a model with data.

2.1.1 Types of machine learning

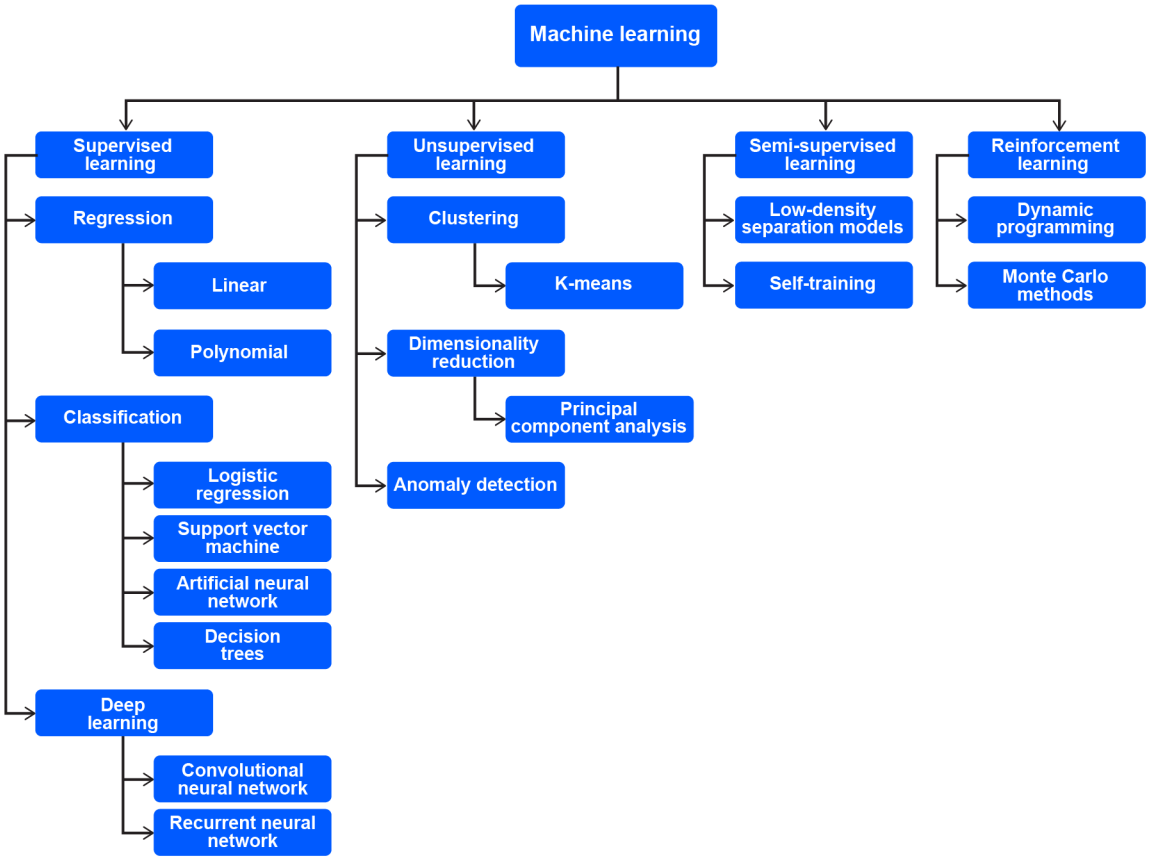


Figure 1. **Types of machine learning**

Supervised learning is a type of machine learning where a computer learns from examples that have been labeled or categorized.

Training data: A dataset that contains input-output pairs; each input has a corresponding correct output (label)

Learning process: The computer uses labeled data to learn the relationship between inputs and outputs (labels of inputs)

Unsupervised learning is a type of machine learning where the model learns from data that has not been labeled or categorized.

Training data: Unlabeled and unclassified inputs, for example, pictures

Learning process: The unsupervised learning algorithm analyzes the unlabeled data to identify similarities and differences among them.

Semi-supervised learning is a type of machine learning where the model learns first from labeled and then unlabeled data.

Training data: Labeled and unlabeled inputs

Learning process: The model first learns from labeled data inputs and then analyzes the unlabeled data to identify similarities and differences among them.

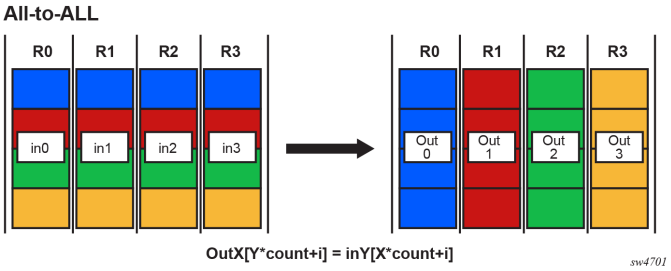
Reinforcement learning is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize rewards.

Training data: Unlabeled input

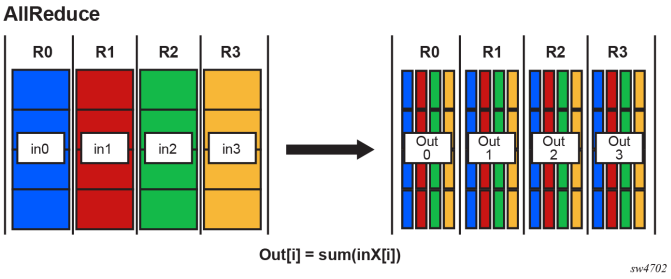
Learning process: The agent learns through exploration and feedback (trial and error) rather than through labeled data. Via a reward system, correct choices are reinforced and incorrect choices are penalized.

2.1.2 Collective communication library operations

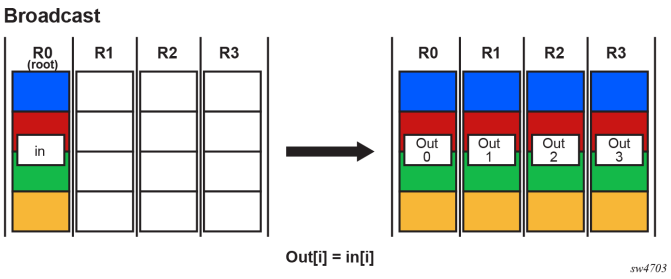
The learning mechanisms described in section 2.1.1 need specific workload distribution and synchronization mechanisms. Some of these operations are described below.



The All-to-All operation gathers *N* values from *k* ranks into an output buffer of size *k*×*N* and distributes that result to all ranks.



The AllReduce operation performs reductions on data (for example, sum, min, max) across devices and stores the result in the receive buffer of every rank.



The Broadcast operation sends data from a designated root process to all other processes. The root process holds the original data, which is copied to all other ranks.

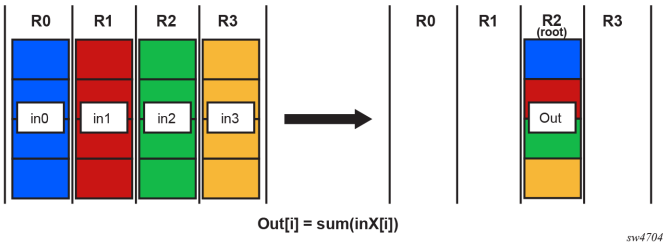
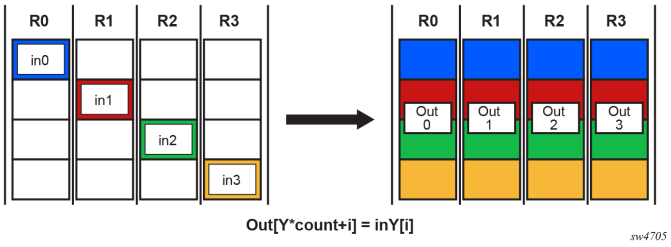
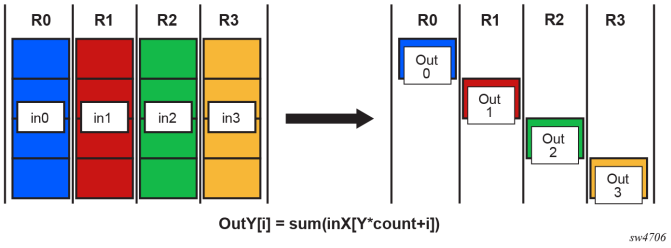
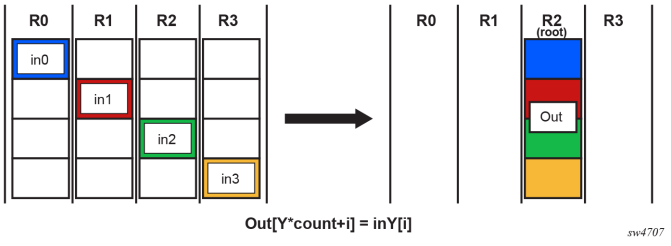
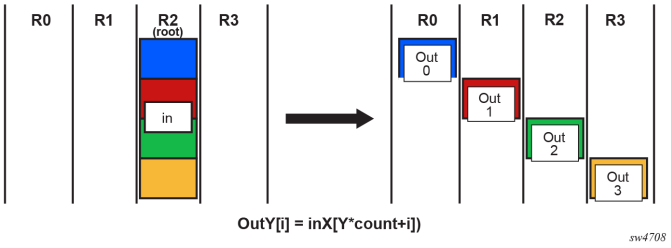
Reduce  	The Reduce operation performs the same operation as AllReduce but stores the result only in the receive buffer of a specified root rank.
AllGather  	The AllGather operation collects data from all processes and distributes the combined result to every process. Each process contributes its own data, and the output is an aggregation of all contributions.
ReduceScatter  	The ReduceScatter operation performs the same operation as Reduce, except that the result is scattered in equal-sized blocks between ranks, each rank getting a chunk of data based on its rank index.
Gather  	The Gather operation gathers *N* values from *k* ranks into an output buffer on the root rank of size *k*×*N*.
Scatter  	The Scatter operation distributes a total of *N*×*k* values from the root rank to *k* ranks, with each rank receiving *N* values.

Table 1 Collective communications library operations

2.2 Types of AI/ML clusters

Training cluster

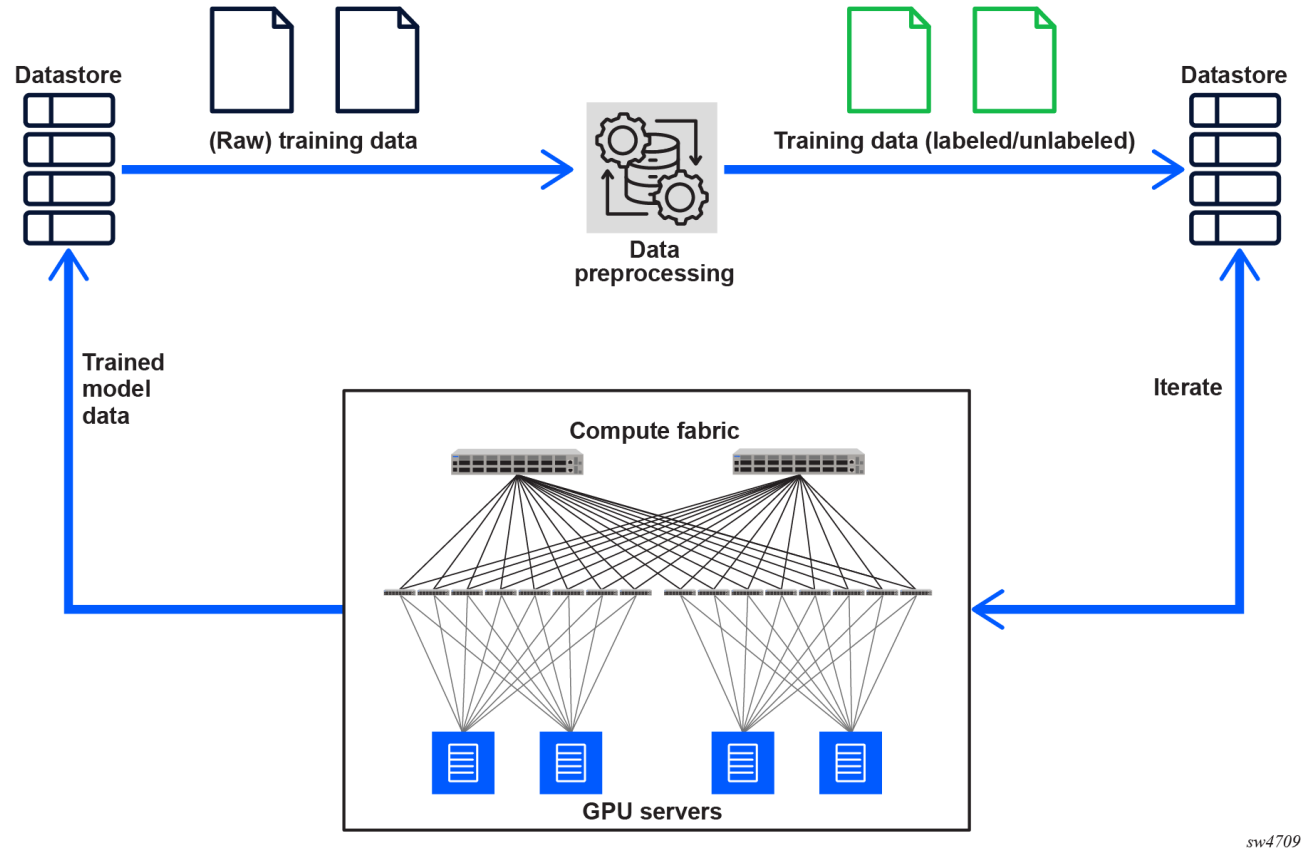


Figure 2. **Training cluster operation**

The primary function of a training cluster is to train the learning models with different forms of data (labeled and unlabeled), based on the type of learning being performed. The learning models have billions of parameters (internal variables) that determine how comprehensive and accurate the model is. When the training process is complete, the trained model is stored in the dedicated storage and can be used for inference. This is a very intensive process and requires high-end GPU servers, storage nodes, and high-bandwidth networking gear.

Inference cluster

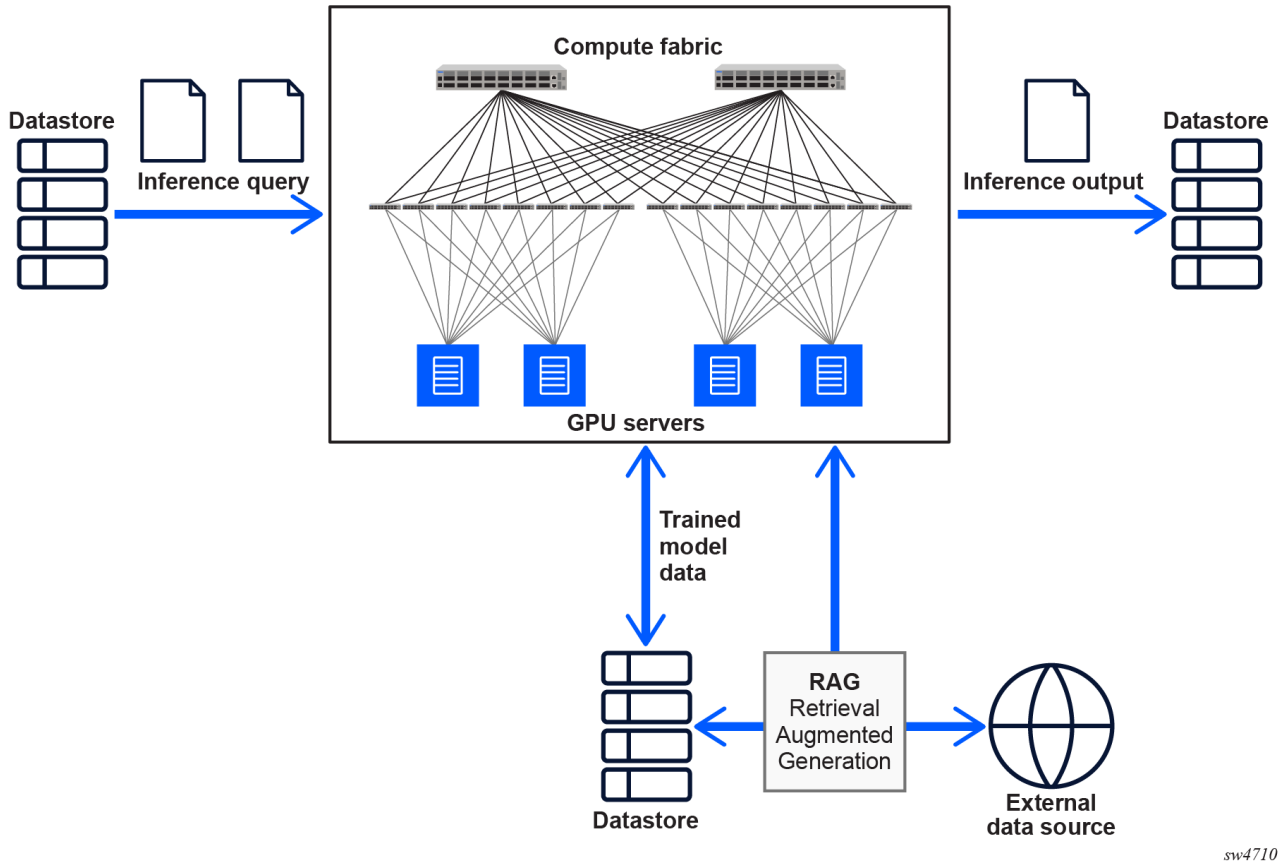


Figure 3. **Inference cluster operation**

When the model has been trained, as shown in the training cluster, the trained model can be used to infer results to user queries. The inference cluster can be single- or multi-node based on the size of the model and the number of queries that need to be processed. The trained model can be reinforced with retrieval augmented generation (RAG), which uses external data sources to keep the information that is provided current. Inference clusters are generally less intensive and can work with lower-order gear. A single AI-DC can be used as a hybrid training and inference cluster.

2.3 AI-DC cluster architecture and scale

2.3.1 Rail optimization

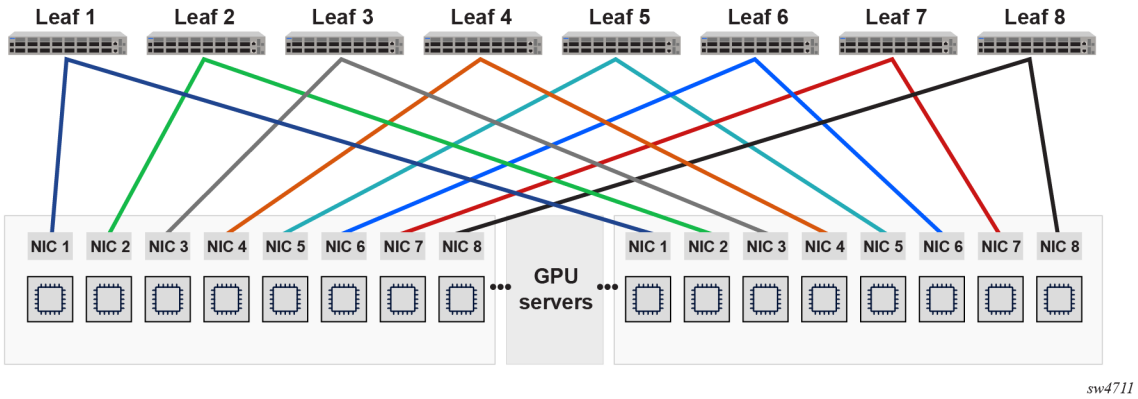


Figure 4. **Rail-optimized fabric**

Rail optimization is a concept adopted first by NVIDIA and then by other GPU vendors as a mechanism to minimize network interference with inter-GPU communication.

Because the earlier models of GPU servers had eight GPUs, the corresponding design of an atomic unit of a backend GPU fabric, also known as a stripe, was designed with eight leaf nodes. Each GPU index is attached to the same leaf, for example: GPU 1 of Server 1 and Server 2 are connected to Leaf 1. This leaf, which contains all GPU numbers, is called a rail; Leaf 1 is Rail 1 in this example. A rail can extend beyond a stripe and a pod, which are defined in subsequent sections of this document.

This structure ensures that intra-rail communication between GPUs only traverses the connected leaf, while inter-rail communication traverses the internal switch.

2.3.2 Rail-optimized stripe

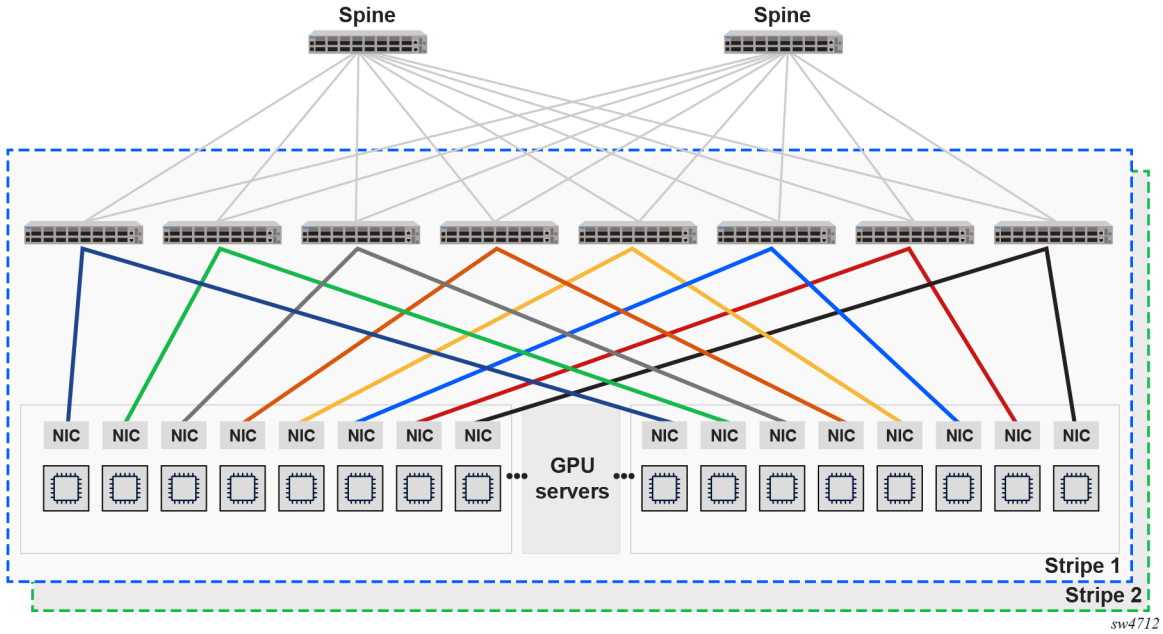


Figure 5. **Rail-optimized stripe**

A stripe is an atomic unit of a GPU cluster.

Earlier, we discussed how eight leafs became the design choice for rail optimization because the GPU servers can accommodate eight GPUs. This eight-leaf design forms a stripe. By definition, a full stripe is the maximum number of GPUs that can be supported by these eight leafs, keeping in mind that half the ports of the leafs are used to connect to the spines due to a prescribed oversubscription ratio of 1.

For example: with a Nokia 7220-IXR-H4 with 32 port switches in the leaf role, assuming an oversubscription ratio of 1, the number of GPUs in one stripe is 16 × 8, which is 128 GPUs.

The number of stripes in a pod is determined by the number of spines and the port density of the spines.

2.3.3 Cluster scale

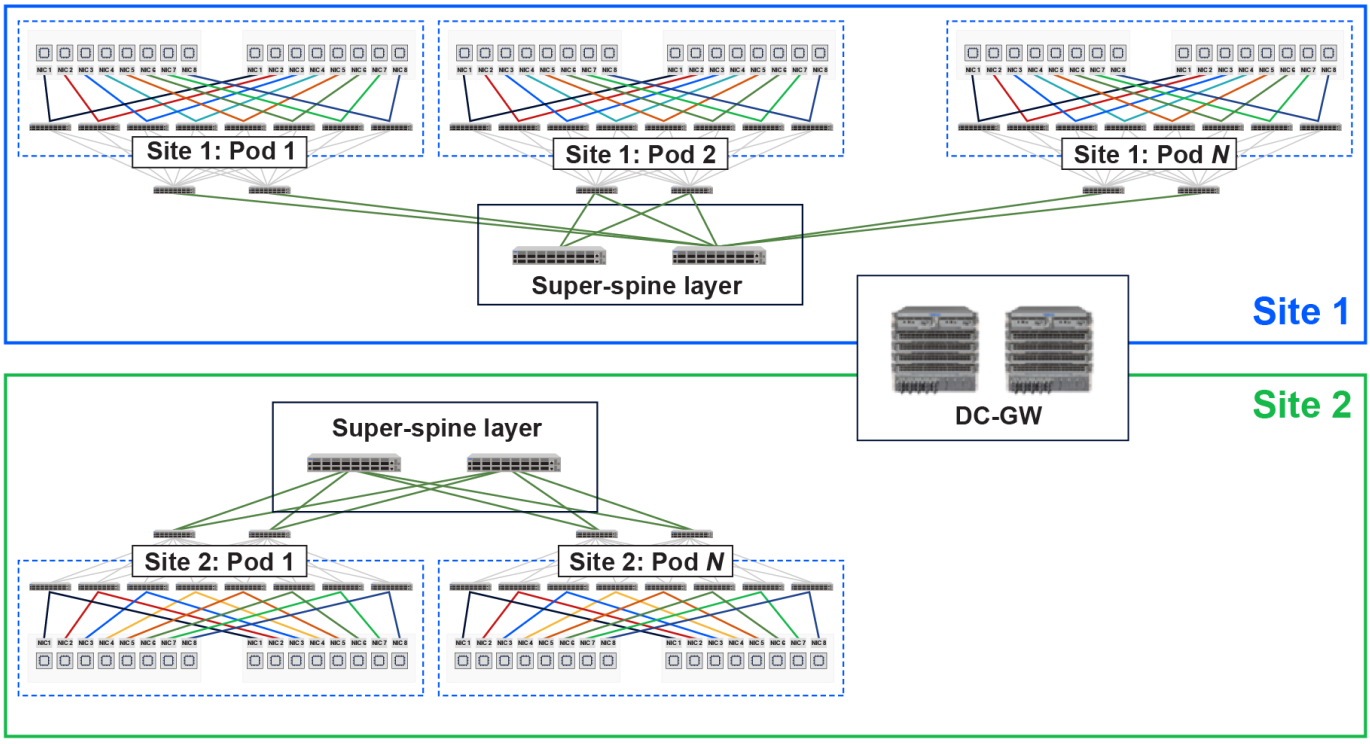


Figure 6. **Multi-site, multi-pod, multi-stripe cluster**

The architecture shown above details the mechanism by which a backend training cluster is built. The basic unit of a pod is a rail-optimized stripe. A stripe is fully scheduled when all south-bound ports of all eight leafs are filled.

The pod is considered fully scheduled when all the ports of the spines are utilized. The port radix of the spines defines the size of the pod.

This architecture focuses on scale out. For example, using the Nokia 7220-IXR-H5-64x800G, each stripe can have 512 GPUs, and each spine can support two stripes, which means each pod can have 1024 GPUs. We can increase the number of supported stripes by increasing the number of spines. The example discussed above can support 32 spines, bringing a single fully subscribed pod size to 64 × 64 = 4096 GPUs.

The pods can scale out within a site via a super-spine layer. The prescribed oversubscription ratio from the GPUs to the super-spine layer is 2.5:1, but this is highly subjective to the kind of workloads being deployed and the resultant traffic patterns across various layers of the fabric.

Multisite clusters replicate the same design paradigm, connect to each other via gateway routers, and employ stitching and handoff mechanisms to connect the site.

A rail and/or a high-bandwidth zone defined by RCCL/NCCL collectives can extend from a single pod to multiple pods over multiple sites.

2.4 Components of an AI/ML cluster

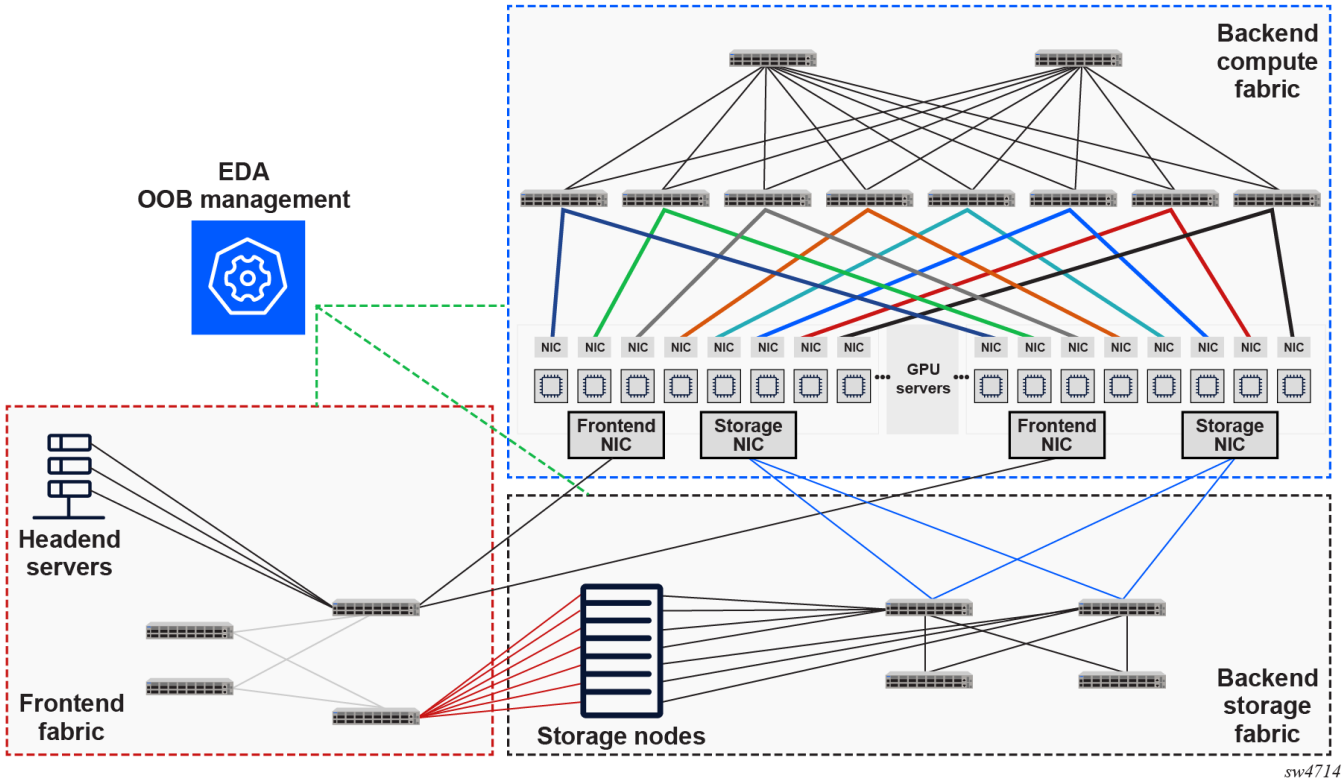


Figure 7. **Components of an AI/ML cluster**

Backend compute fabric is a fabric that hosts the GPU servers.

The GPU direct ports on the server, the number of which is usually equal to the number of GPUs in the higher-end training servers (such as NVIDIA H200 GPUs), are connected to this fabric.

All AI/ML training optimization concepts such as rail optimization, dynamic load-balancing, and congestion control are relevant to this segment.

Front-end fabric is connected to the in-band front end ports of the GPU servers and the dedicated storage nodes. All training and inference jobs are scheduled via the servers connected to the front-end fabric. The front-end fabric is also connected to the headend servers and/or the internet, as well as the headend and MCP servers and RAG gateways.

The front-end fabric is also connected to the dedicated storage portions to manage volumes and checkpoints.

Backend storage fabric connects the storage ports of the GPU server to the dedicated storage nodes. The dedicated storage nodes are used to store the data processed by the GPUs during a training or inference process. WEKA, VAST, and Pure storage are examples of dedicated storage node vendors. The number of required dedicated storage nodes required depends on the number of GPUs present in the cluster. Unlike the compute fabric, there are no direct GPU ports to the storage fabric and there are usually only one or two NICs per GPU server going to the storage fabric.

2.5 Traffic flow in AI/ML DC clusters

2.5.1 Intra-node communication

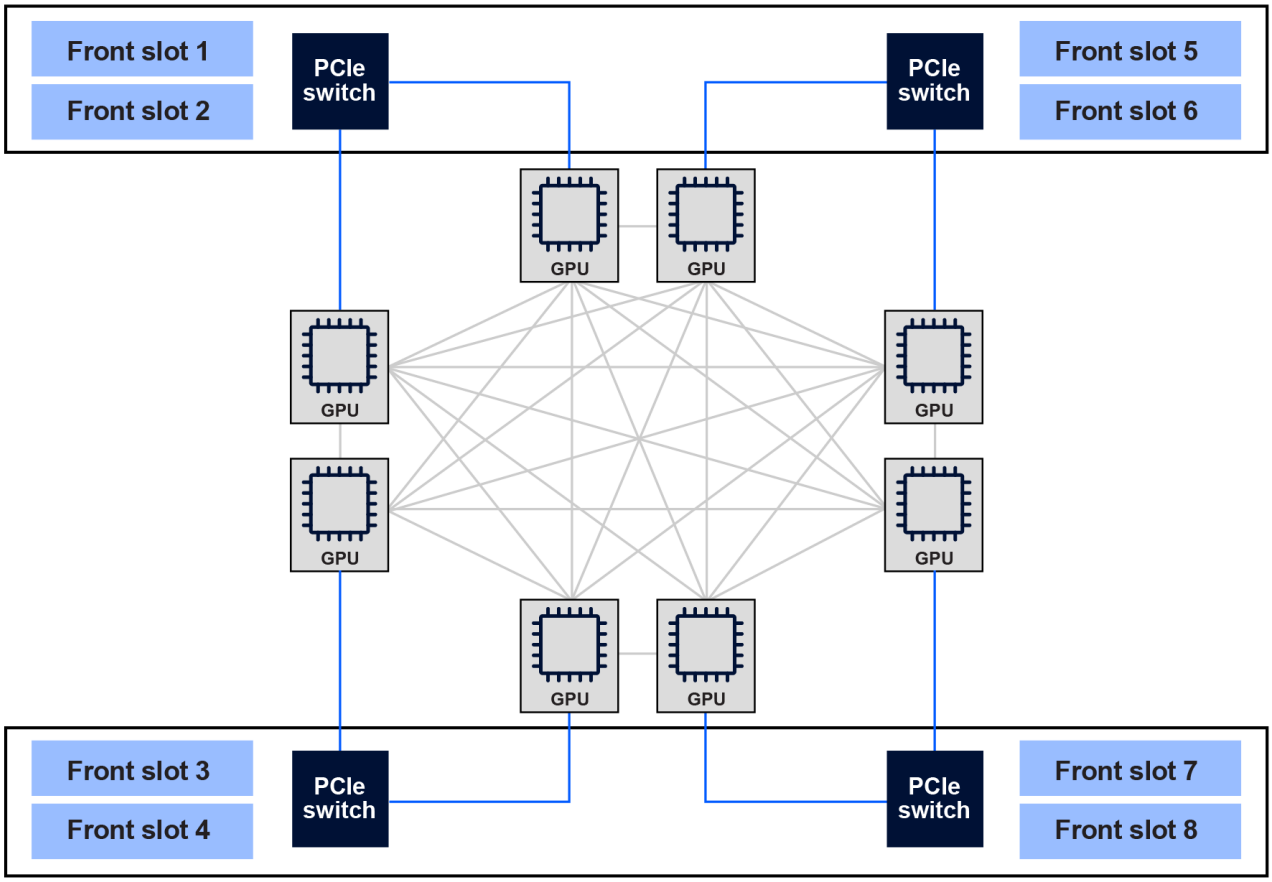


Figure 8. **Internal network architecture of a GPU server**

Figure 8 shows the GPU switching reference architecture for the Lenovo server capable of hosting the NVIDIA H200-series GPU.

As shown, the GPUs have a full mesh connectivity with each other via the NVLink, which means every GPU in the server is one hop away from every other server. The PCIe switches are connected to their respective GPUs and provide access to the eight GPU direct ports, which connect to the compute fabric.

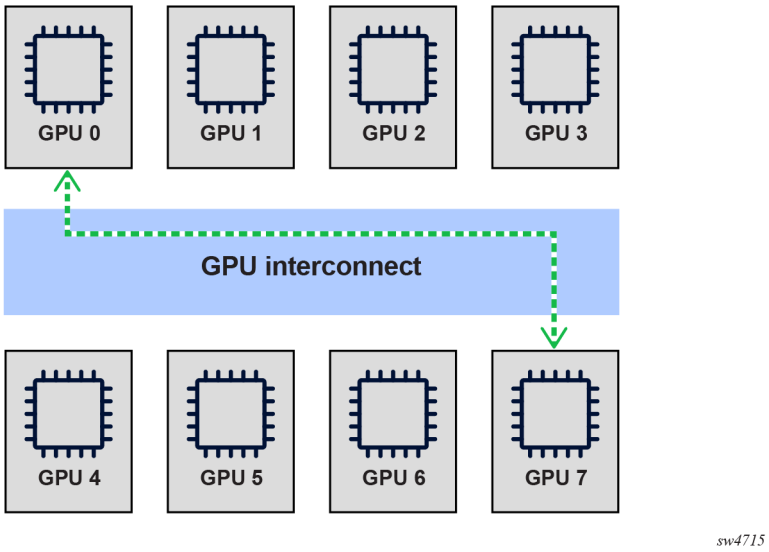


Figure 9. **Intra-node communication**

Intra-node communication outlines the communication between GPUs in a single server.

Figure 9 shows that each NVIDIA H200 GPU has full mesh connectivity to every other GPU via the internal switch. For example, if GPU0 wants to talk to the GPU7 rail, it can place the payload internally via the NVSwitch and switch it to GPU7.

2.5.2 Inter-node communication

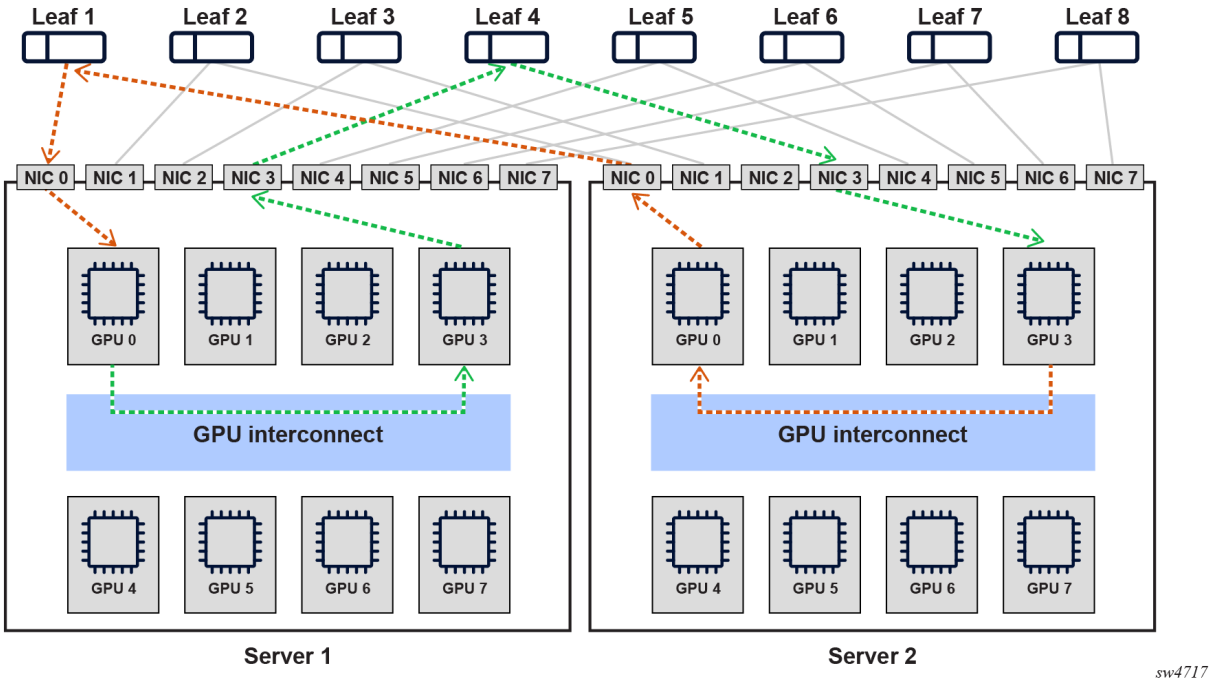


Figure 10. **Inter-node communication**

Inter-node communication refers to the communication between GPUs in a single stripe.

In a rail-optimized fabric, each GPU number is connected to the same leaf; that is, GPU0 of all servers in the stripe are connected to leaf1. For example, if GPU0 on Server 1 wants to send data to GPU 3 on Server 2, it first sends it via the GPU interconnect to its own GPU 3, then GPU 3 on Server 1 sends it to Leaf 4, and finally to GPU 3 on Server 2, as indicated by the green path.

If GPU 3 on Server 2 wants to send data to GPU 0 on Server 1, it sends it to its own GPU 0, then it gets forwarded to Leaf 1, and finally to GPU 0 on Server 1, as indicated by the orange path.

2.5.3 Inter-stripe communication

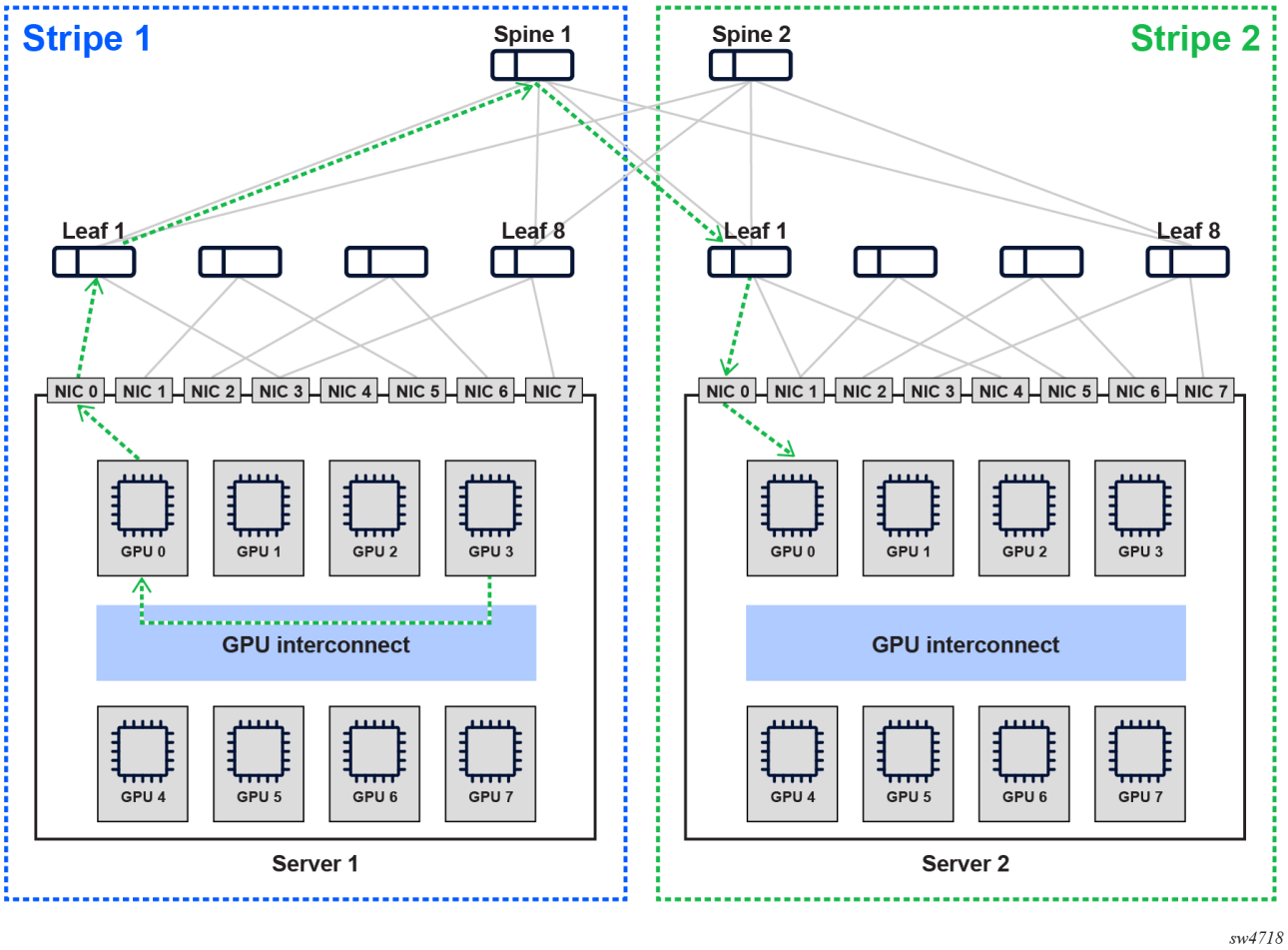


Figure 11. **Inter-stripe communication**

A stripe is defined as all the GPUs that are connected to a set of eight leafs.

In the scenario shown in Figure 11, there are two stripes, which are depicted by the green and blue labels. Each stripe is connected to the same set of spines. Server 1 is part of Stripe 1 and Server 2 is part of Stripe 2, which means they need to traverse the spine to reach each other.

In the example shown in Figure 11, if GPU 3 of Server 1 wants to communicate with GPU 0 of Server 2, it places the payload onto GPU 0 of Server 1 via the infinity link GPU interconnect. GPU 0 is connected to Leaf 1 of Stripe 1 and GPU 0 of Server 2 is connected to Leaf 1 of Stripe 2.

As a result, GPU 0 of Server 1 sends the packet to one of the spines, which is decided by the load balancing mechanism—in this example, Spine 1. Spine1 then sends the packet to GPU 0 in Stripe 2.

2.6 Congestion management

2.6.1 Congestion scenarios in AI/ML networks

There are several congestion scenarios, and several congestion points that can form in these scenarios, based on the flow and nature of traffic. Congestion scenarios and points are unique and are driven by the positioning of end points, workload distribution, and the natures of the learning models and applications. Some of these scenarios are listed below.

2.6.1.1 Scenario 1: Intra-rail simultaneous microburst

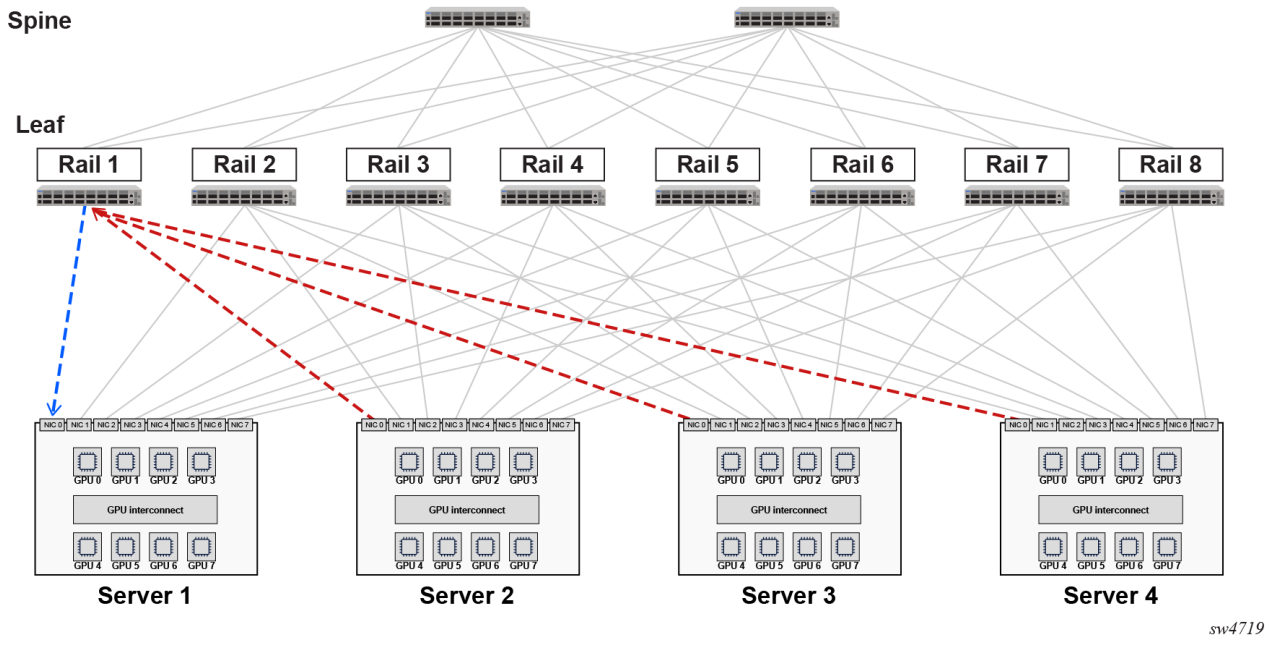


Figure 12. **Intra-rail intra-stripe congestion**

The traffic pattern in AI/ML training clusters is based on the data, model, or pipeline parallelism that is employed to distribute the workload; based on this parallelism, the individual GPU receives a piece of the workload, processes it, places the entire payload onto the network queue, and sends it out to the other GPUs at the same time to synchronize the state.

Note: Ethernet-based fabrics use RoCEv2 for communication.

In the scenario shown in Figure 12, all communication for Rail 1 needs to go through Leaf 1 (which is Rail 1 in this case). This might happen, for example, if all GPU 0s need to synchronize with each other.

If all GPU 0s decide to synchronize with GPU 0 of Server 1 at the same time, thereby sending out a microburst at the same time, there is a chance that the dedicated buffer on the switch will become overwhelmed and the shared buffer utilization will also become impacted, leading to congestion.

2.6.1.2 Scenario 2: Inter-stripe spine polarization

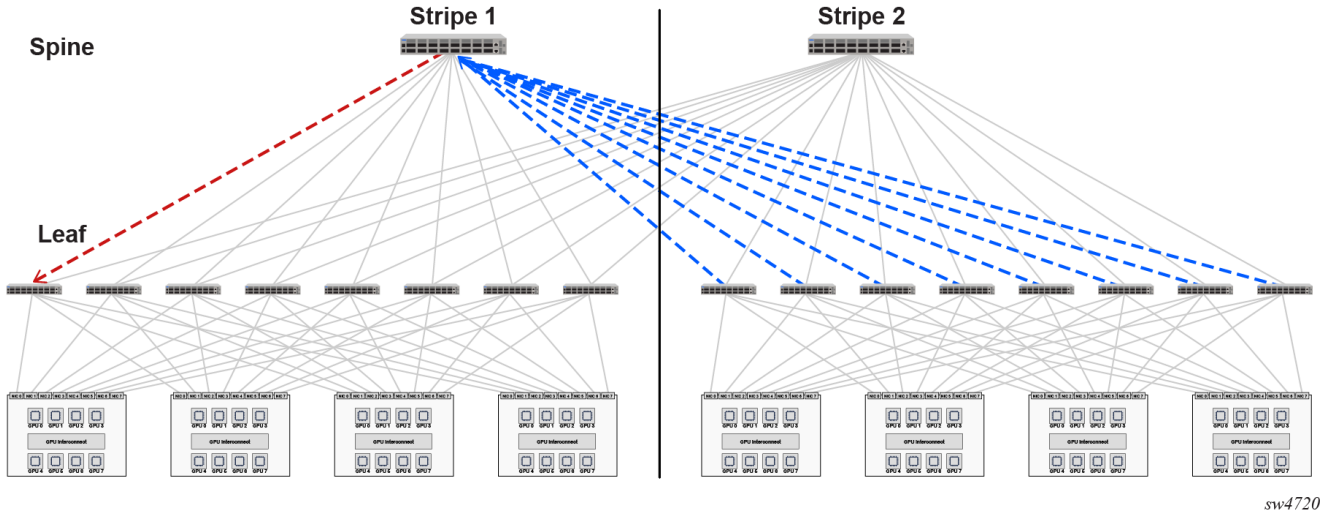


Figure 13. **Inter-stripe spine polarization**

In the scenario shown in Figure 13, consider an All-to-All operation where every GPU in the high-bandwidth zone must synchronize with every other GPU and, for an instance in time, all the GPUs polarize to a single spine to send traffic across to a rail in another stripe. This can lead to congestion at the spine layer.

2.6.1.3 Scenario 3: Inter-stripe leaf-GPU congestion

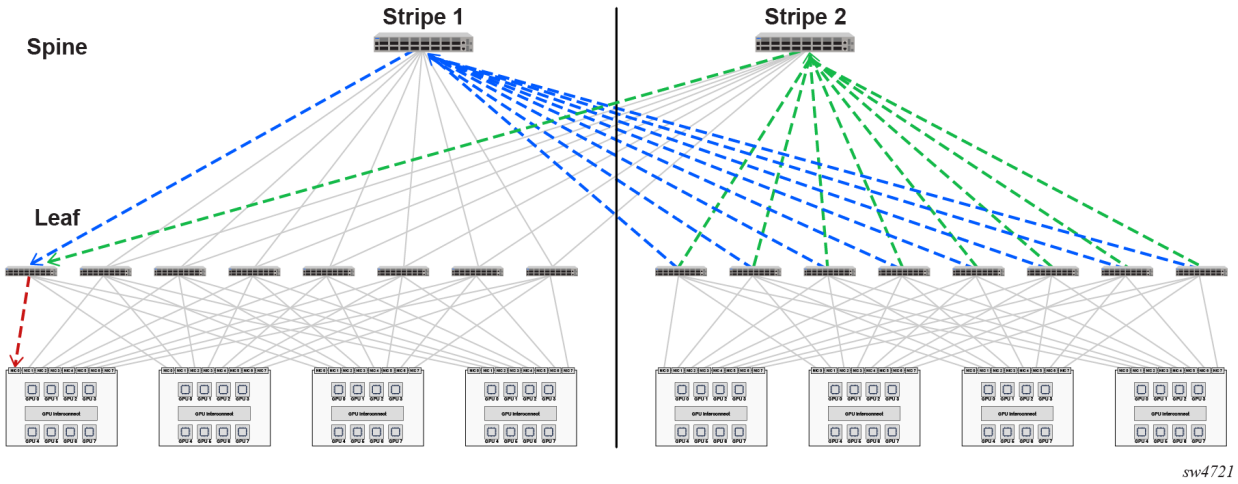


Figure 14. **Inter-stripe leaf-GPU congestion**

The workload distribution may happen over several stripes, and this can lead to a scenario when multiple stripes try to synchronize with GPU 0 of a single server at the same time, as shown in Figure 14. This can lead to congestion in the GPU southbound ports.

2.6.1.4 Scenario 4: Inter-pod super-spine polarization

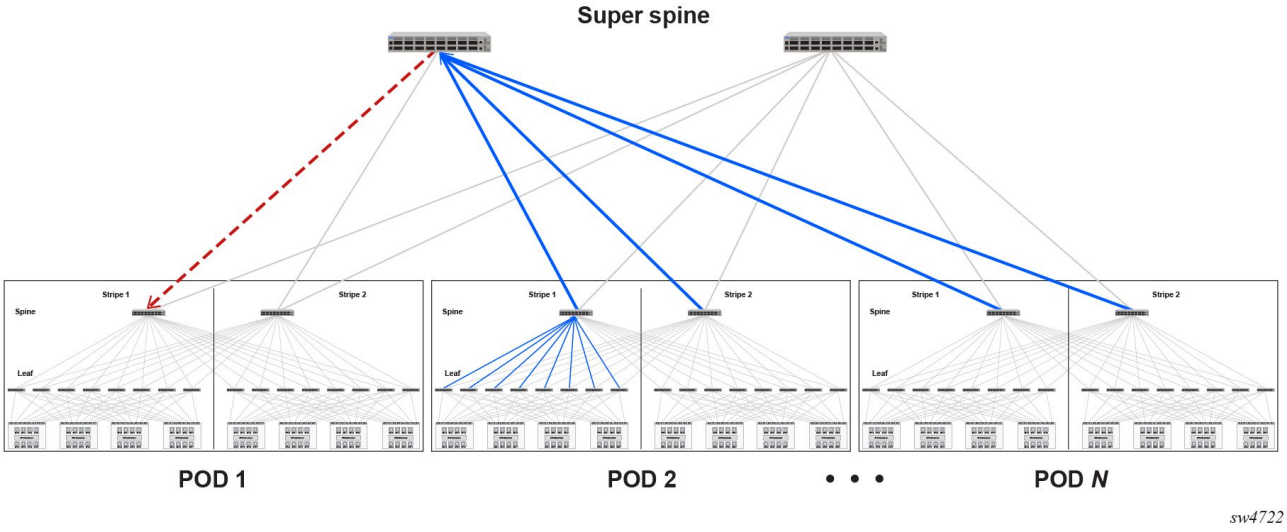


Figure 15. **Inter-pod super-spine polarization**

In Figure 15, the workload is distributed among multiple pods, which have multiple stripes. There are two scenarios that are possible here:

- Multiple pods polarize to the same spine, as shown in the figure above, and cause buffer overflow on the super-spine due to polarization and the simultaneous burst.

- Oversubscription to the super-spine layer is greater than 1, which means that there are more leaf-to-spine ports than there are spine-to-super-spine ports. As a result, the rate of incoming traffic to a spine can exceed the rate at which the spine can send it out, which can cause congestion.

Note: Many probable congestion scenarios can be avoided by using non-blocking architectures and maintaining an oversubscription ratio of 1.

2.6.2 Datacenter quantized congestion notification

A lossless fabric is very important to an AI/ML network, and it can be ensured in part by having a non-blocking network with an oversubscription ratio of 1. However, due to the bursty nature of traffic as shown in the scenarios above, the fabric can still face congestion. We need congestion detection and control mechanisms to alleviate congestion and ensure optimal LLM training times and inference token utilization.

Datacenter quantized congestion notification (DCQCN) is a mechanism that can be used to identify and manage congestion. It is an end-to-end congestion control mechanism designed for RDMA over converged ethernet (RoCEv2) networks.

DCQCN combines explicit congestion notification (ECN), which is an end-to-end congestion detection and control mechanism, and priority flow control (PFC), which is a per-segment control mechanism.

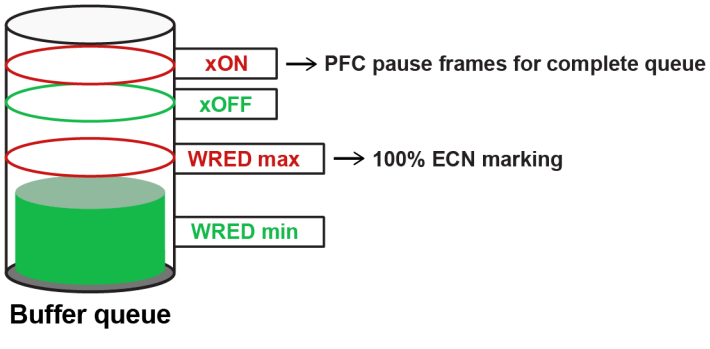


Figure 16. **DCQCN order of operation based on shared buffer utilization**

2.6.2.1 Explicit Congestion Notification

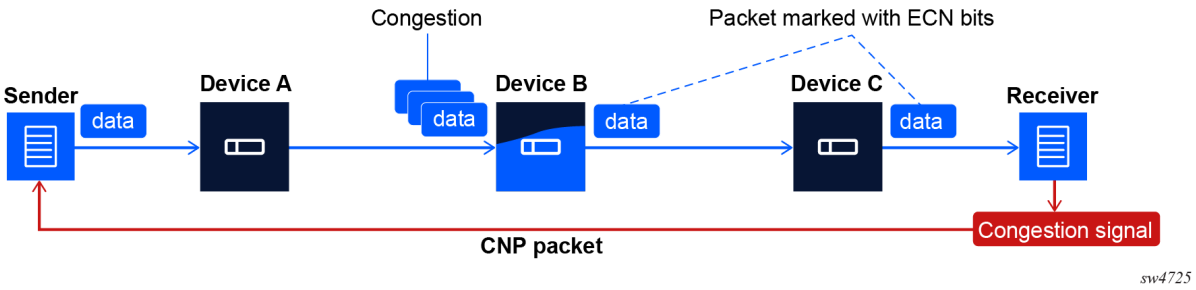


Figure 17. **ECN workflow**

Explicit Congestion Notification (ECN) RFC 3168 facilitates end-to-end congestion management without dropping packets and is ideal for lossless AI fabrics.

Figure 17 shows the communication between two endpoints, a sender and a receiver, with three network devices in between them on the network path.

Device B on the network path faces congestion and, when this happens, it starts marking packets with ECN 11 bits. When one of these marked packets reaches the receiver, the ECN sensitive receiver sends a congestion notification packet (CNP) back to the sender via a dedicated queue that has been allocated strictly for it on all the network devices in the path.

After the sender receives this packet, it will realize that there is congestion in the network, and it needs to decrease the rate of traffic that is being sent out.

2.6.2.2 Priority flow control

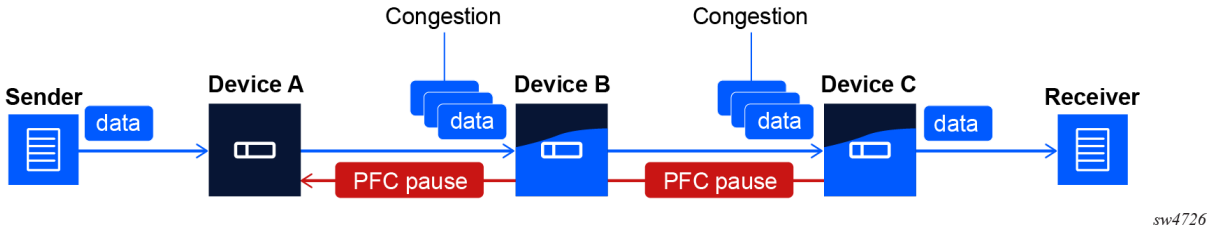


Figure 18. **PFC workflow**

PFC, unlike ECN, works on a per-segment basis. As shown in Figure 18, data is being sent from the sender to the receiver via network Devices A, B, and C. Note that Devices B and C are facing congestion. PFC pause frames will be sent from Device C to B, and from Device B to A. If either Device C or B stops being congested, the corresponding pause frames will stop as well.

Note: Unlike Ethernet pause, which is sent on the entire segment, PFC pause is only sent on the queue that is facing congestion. That is, if only queue 3 of 0 to 7 is facing congestion, queues 0 to 2 and 4 to 7 do not see the PFC pause and, unlike a CNP packet which flows all

the way to the sender, PFC pause ends at the segment where the congestion was seen.

2.6.2.3 ECN and PFC threshold

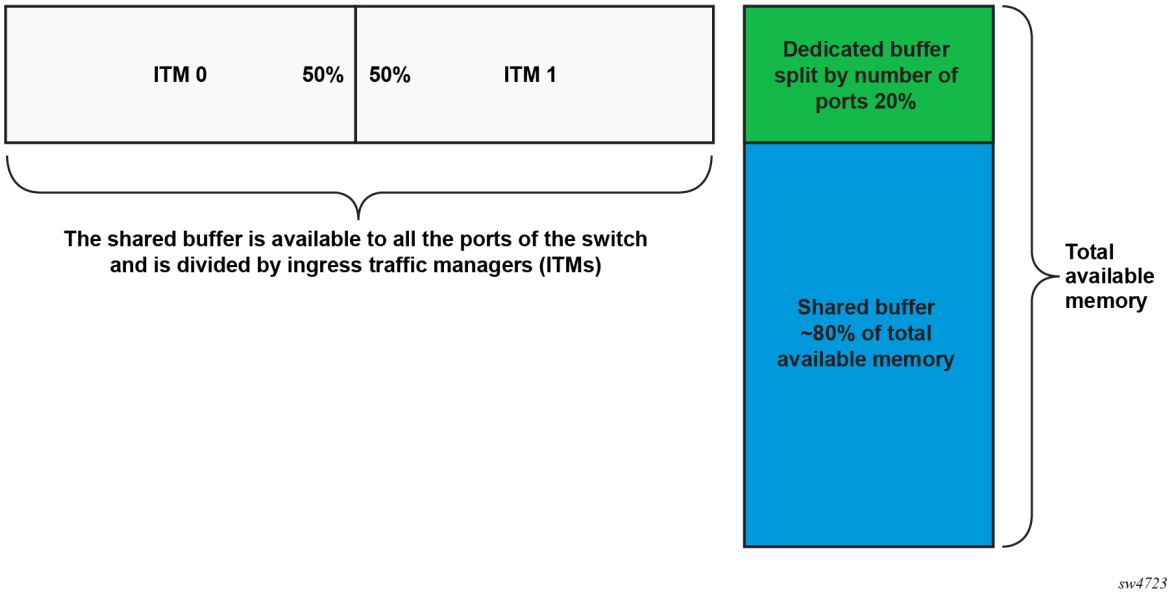


Figure 19. **Dedicated and shared buffer allocation**

Note: The following explanation is given with approximate ratios and does not connote actual memory capacity of a specific Nokia platform.

The total available memory of the router is split into two kinds of buffers: the dedicated buffer and the shared buffer. The dedicated buffer is then further allocated equally to all available ports.

For example: If the total memory is 100 mb, about 20 mb is reserved as the dedicated buffer and, since we have 32 ports, the dedicated buffer available to each port is 20/32 = 0.625 mb.

As shown in the diagram, the remaining ~80% is split per ITM and, assuming default alpha values, each ITM receives 50%, which will be available to all ports that are part of that ITM (16 per ITM out of 32 ports, not in sequential order). See section 8.1.7 for examples.

The buffer is further allocated per port per queue. For example, if there are 16 ports and queues 3 and 6 have congestion on all the ports, the buffer allocation is:

40 mb (per ITM) / 16 ports × (2 queues per port)

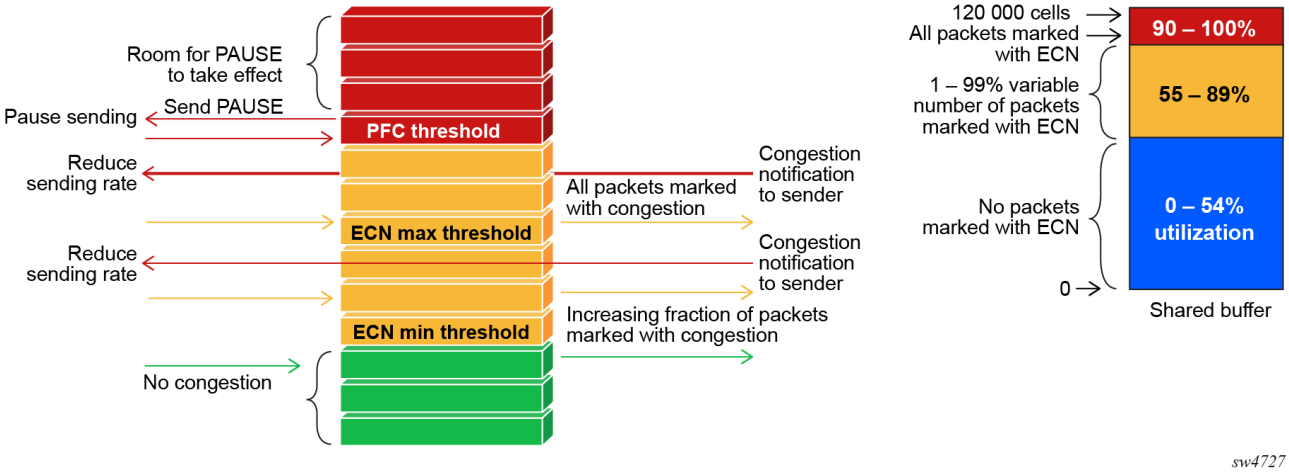


Figure 20. **ECN and PFC threshold**

ECN and PFC thresholds are all factors of shared buffer utilization; as a result, the thresholds shown in the above figure are percentages of the available shared buffer memory shown in the calculation above. In this example, the ECN minimum is set as 55% and the maximum is set to 90%, which means that it will start marking packets when the shared buffer utilization hits 55% of the maximum available share buffer and that it will mark 100% of all packets with ECN bit 11 at 90% shared buffer utilization.

Note: These threshold values are highly specific to customer environment, type of workload, traffic patterns and so on, and should be set to optimize performance for that specific environment.

A general set of rules that can be followed are:

- ECN should trigger before PFC and should be optimized so that the end points reduce traffic before the network sees PFC pause frames during congestion.
- ECN should be optimized such that the network doesn’t reduce traffic, and hence performance, without congestion in the network, as an early trigger of ECN can result in suboptimal utilization.
- PFC pause should trigger before tail drops occur in the segment.

2.7 Load balancing

2.7.1 Static ECMP hash-based load balancing

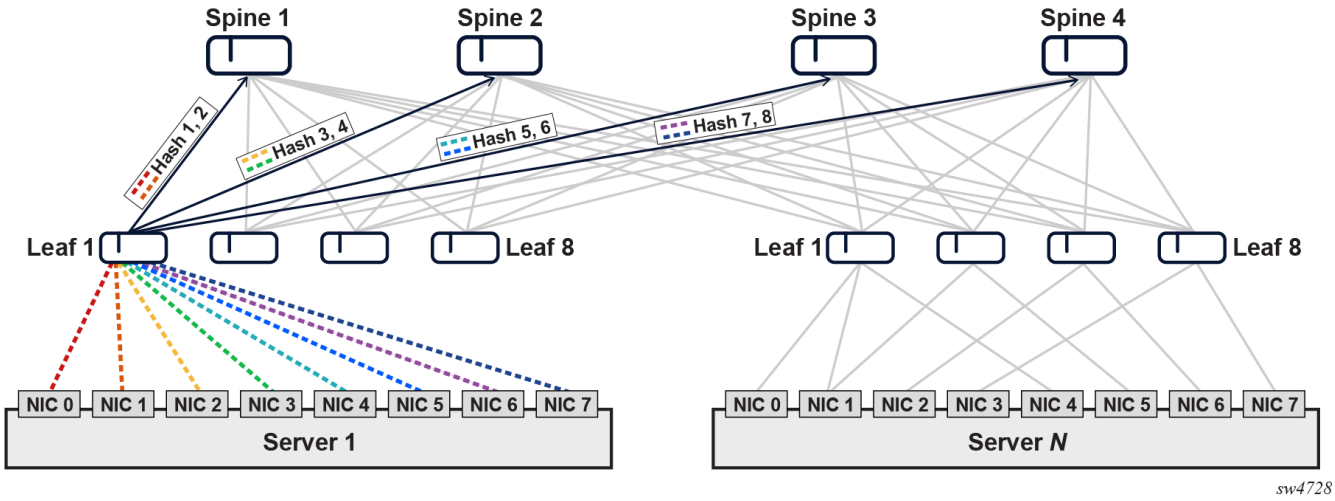


Figure 21. **Static ECMP hash-based load balancing – normal state**

In the case of static ECMP load balancing, when traffic arrives at the network node, a hash is calculated (typically a 5-tuple hash consisting of the source IP address, destination IP address, source port, destination port, and protocol). This makes the hash unique for every traffic stream if any of the 5-tuples vary.

After the hash is calculated, the flows are grouped based on the number of member ports or exit interfaces available and one or more flows are assigned to the interface, as shown in Figure 21.

During operations, if one of the links becomes congested or degraded as shown in Figure 22, ECMP hash does not change the exit interface of the flow until the link is down, and it only re-calculates after that failure. This impacts the flows mapped to that degraded link and results in sub-optimal performance.

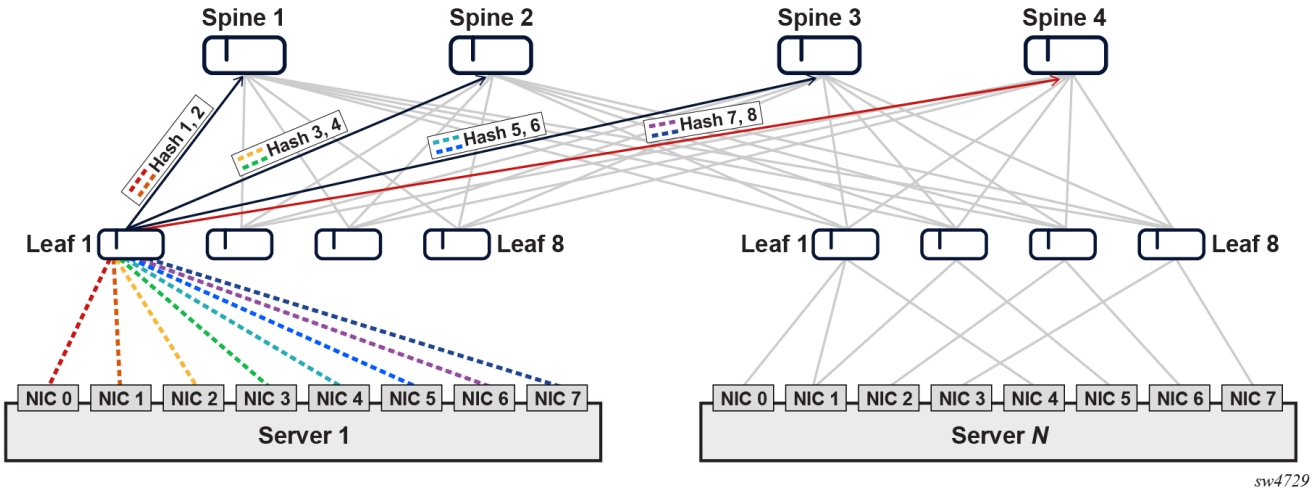


Figure 22. **Static ECMP hash-based load balancing – congested state**

Therefore, a more reactive form of load balancing, dynamic load balancing (DLB), has been developed to accommodate for link congestion and degradation.

2.7.2 Dynamic load balancing

DLB is a reactive load-balancing mechanism that improves the hash-based load balancing by adding an additional metric that determines the state of the member link. When the member link degrades, the flows are rebalanced across the other available links.

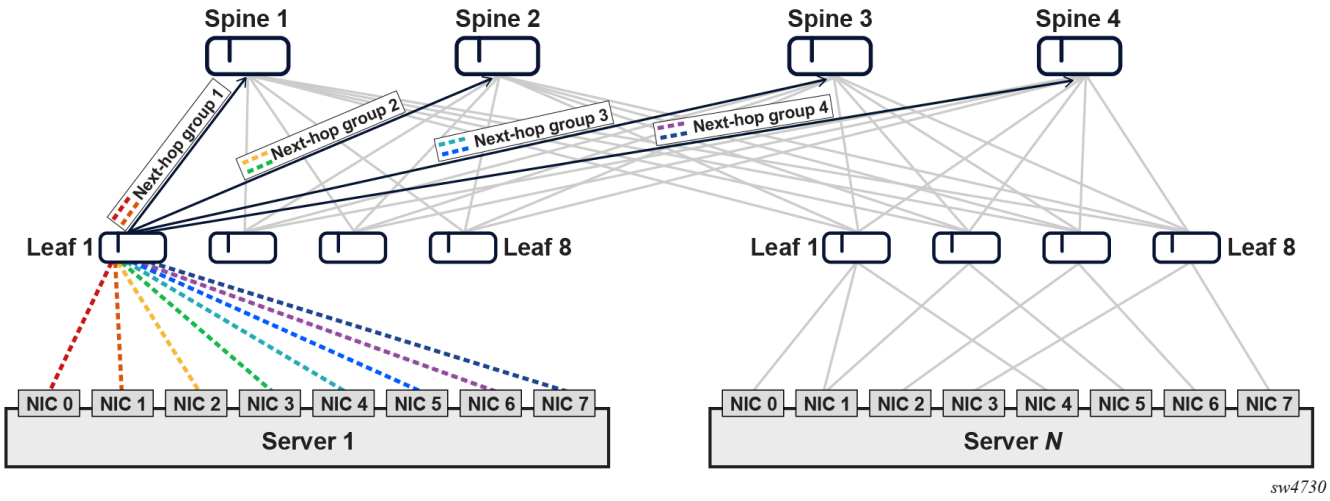


Figure 23. **DLB – normal state**

Each link is assigned a metric, which is dynamically adjusted by sampling the link quality. The link quality sampling interval can be set from 1 to 255 microseconds, with the default being 5 microseconds.

Some of the characteristics that are used to calculate the metric of each link are:

- total egress port queue size
- egress port utilization
- ingress traffic manager port queue size

The default weight percentage that each of the above hold is 70%, 20%, and 10%, respectively.

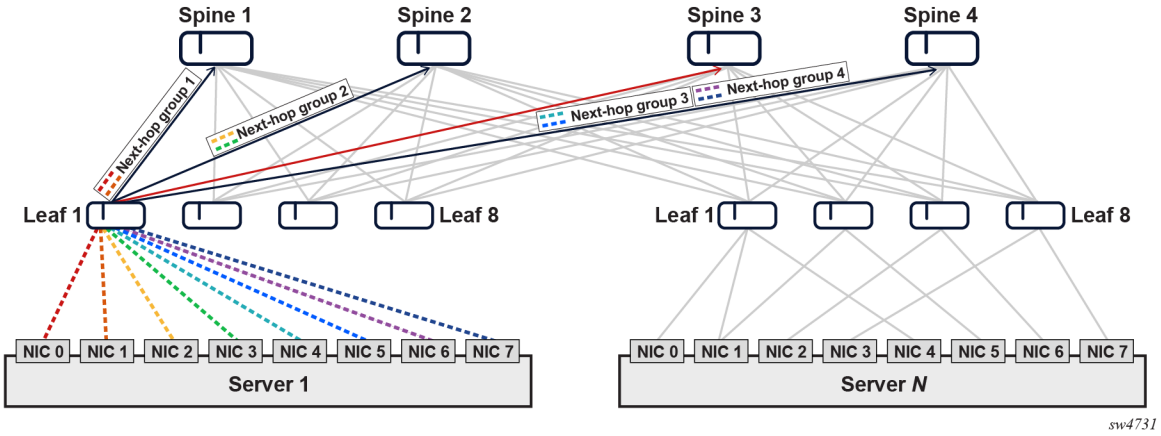


Figure 24. **DLB – congested state**

When the link becomes congested, the sub flow idle time (the time interval between flows in a flow group) exceeds the inactivity timer, and this makes it eligible for reassignment to another member link. See section 8.1.4 for examples.

Note: If the inactivity timer is not optimized to the environment and traffic pattern, it results in the flows jumping between member links even when there is no congestion in the network.

The DLB implementation on Nokia enables queue-pair based hashing by default.

2.7.2.1 Queue-pair hashing

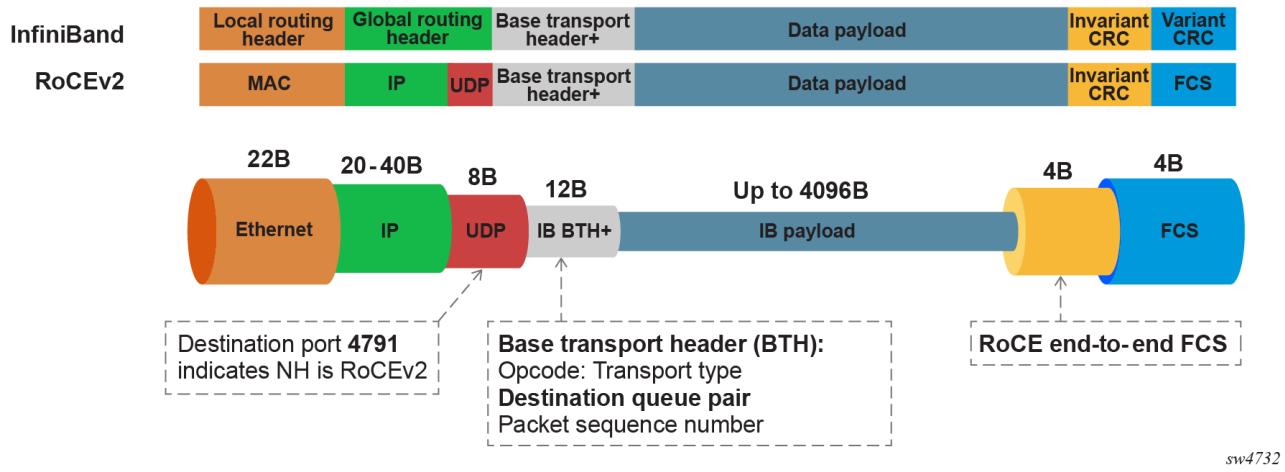


Figure 25. **RoCEv2 packet format**

RDMA allows reading and writing directly to memory via send and receive queues, which form a queue pair. Because traffic from GPU servers is fairly homogeneous, the traditional 5-tuple hash may not be enough to differentiate between the flows.

To add more variance into each flow, with quadratic probing (QP) hashing the queue pairs are also added to the hash calculation such that even if the source and the destination are the same for two flows, the calculation generates a different hash based on the send and receive queues. QP hashing helps load balancing by giving more granularity to the algorithm.

2.8 Multitenancy

The main design consideration for multitenancy in an AI/ML cluster is that the GPUs servers today have internal switching mechanisms that allow all the GPUs in a server to communicate with each other without traversing the external network. Due to this mechanism, there are limitations on how multitenancy can be achieved via the network fabric, and the control of this operation remains with the endpoints.

Some of the mechanisms of achieving multitenant clusters are described below.

2.8.1 Server isolation

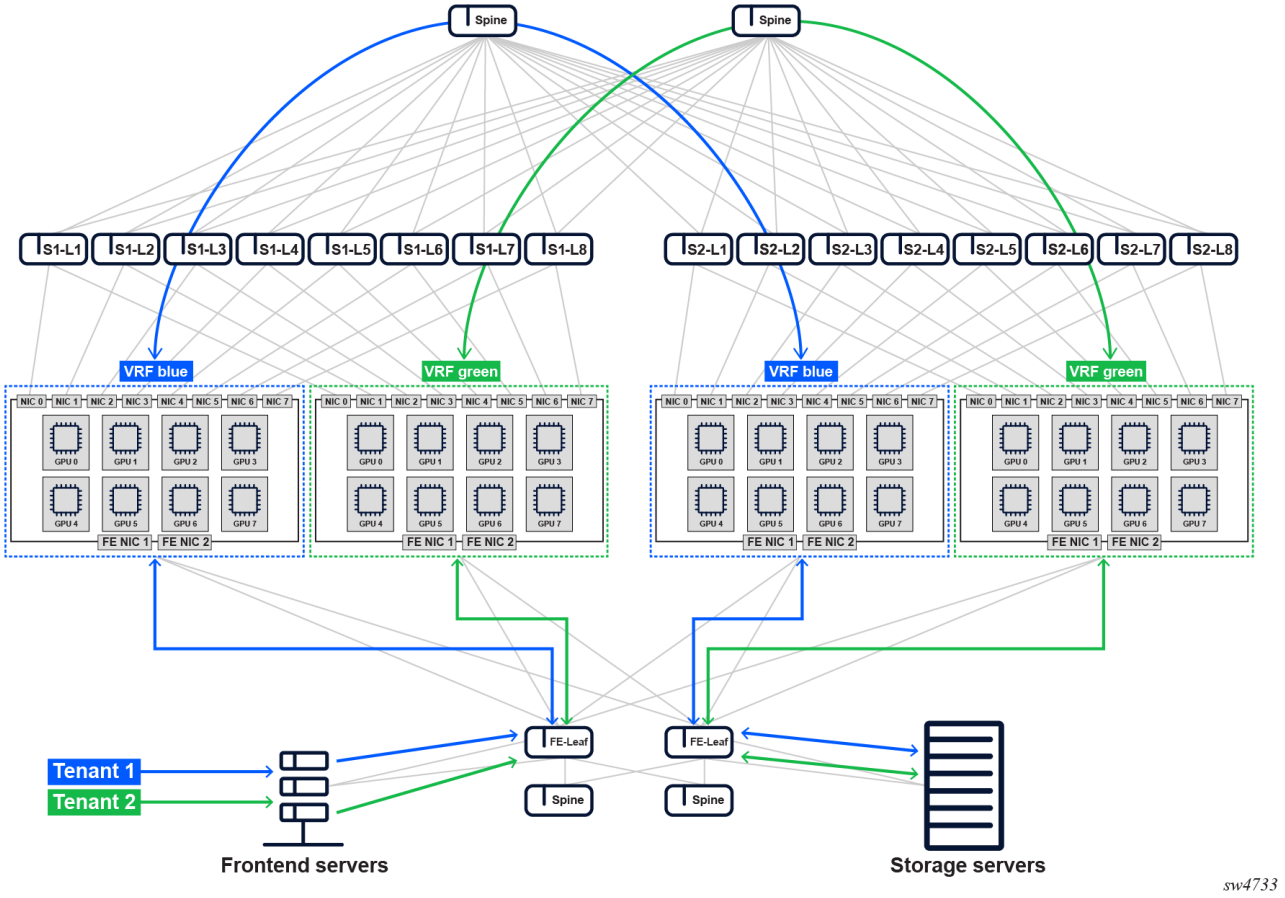


Figure 26. **Server isolation**

One of the mechanisms for providing multitenancy in an AI/ML cluster is server isolation. The positive aspect of this design is that it can be entirely orchestrated via the fabric. The endpoints are isolated by limiting their visibility in the fabric.

The challenge, however, is that the isolation is limited to multiples of servers, which means that we can only provide full servers to the tenants as shown in the diagram.

With this mechanism, we cannot provide isolation at the GPU level because the recommendation by the vendor is that the internal switching mechanism remains enabled, and as a result all the GPUs in a server are able to talk to each other.

The design must ensure that all the GPU ports of servers allocated to a particular tenant are allocated to an IP-VRF, as shown in the figure above. This can be achieved either with VRF-lite on an IP fabric-based solution or an EVPN/VXLAN-based solution.

In an EVPN fabric, the routes from the servers are translated into T5 routes and extended in the fabric. In the case of VRF-lite IP fabric-based solution, the VRF needs to be extended or translated as per the tenant's reach.

In the example shown in Figure 26, irrespective of whether the internal switching is optimized and enabled, the GPUs in the server in VRF blue are not able to talk to the servers in VRF green and vice versa. They are only able to talk to other servers in the same VRF across the cluster, which results in enabling tenant-level isolation.

2.8.2 GPU isolation

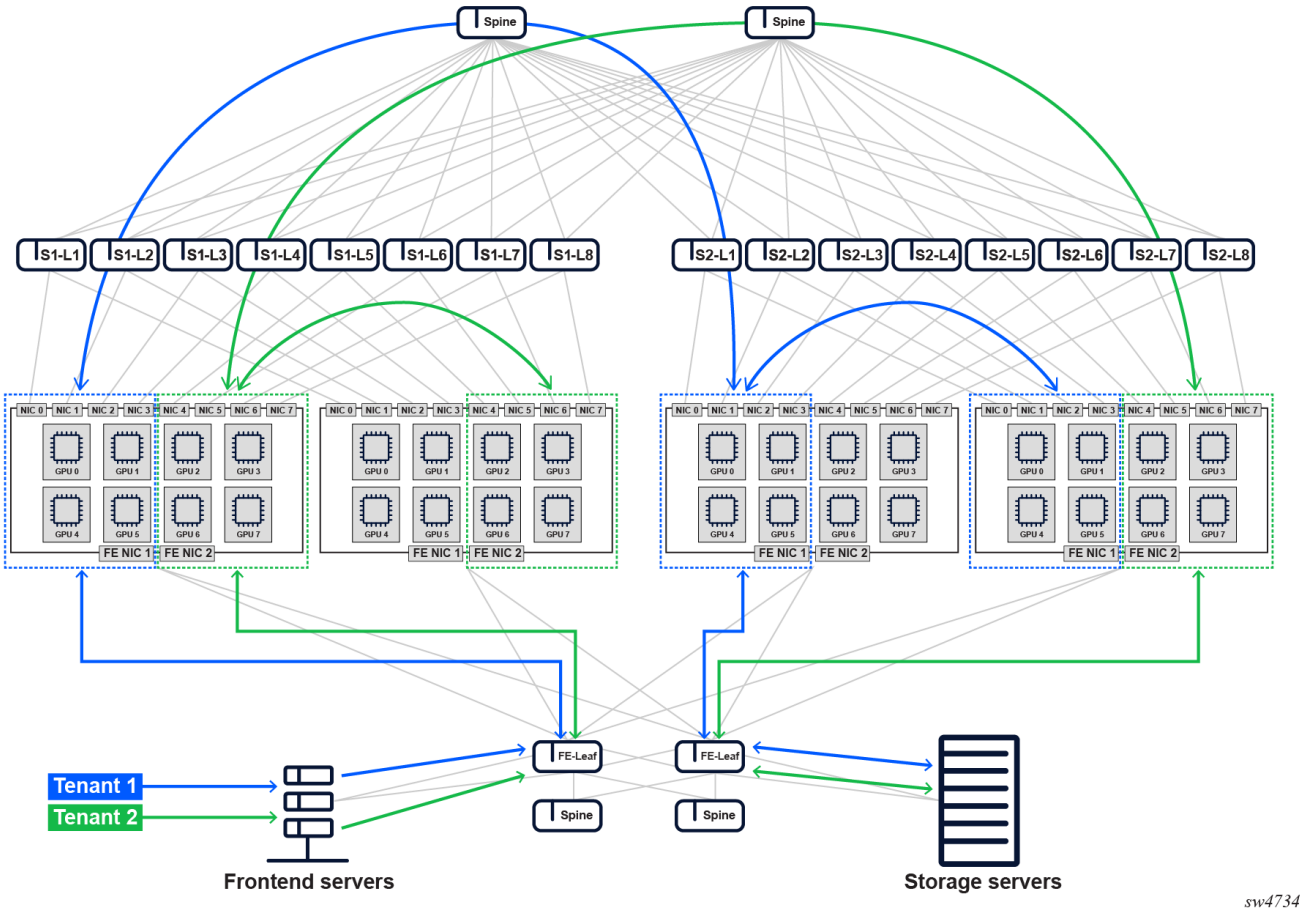


Figure 27. **GPU isolation**

GPU-level isolation for multitenancy, as shown in Figure 27, is where a subset of a server (for example 4 out of 8 GPUs) is allocated to a tenant. This is not limited to a single server, and the GPU allocation can be across the cluster. One mechanism to achieve GPU-level allocation is to allocate a subset of GPUs to a tenant and provide an appropriate IP addressing schema to ensure tenant-based connectivity.

After connectivity has been established, the internal switching mechanism and the optimization must both be disabled so that the GPUs cannot communicate internally. However, GPU vendors recommend to avoid disabling the internal switching mechanisms because it may lead to unexpected behavior and internal optimization can still occur.

Note: Disabling the internal switching mechanisms isn't the preferred option for GPU-level isolation.

Another way to enable GPU-level isolation is to use the CCL framework over the networking layer. In this design, we can provide a network-level segregation, but the actual isolation comes from the upper layers.

For example: CUDA variables can define which GPU is visible to which other GPU in the framework. Schedulers such as Slurm can work in conjunction with CCL to allocate a workload to a specific number of GPUs in a cluster.

With this method, at the network level, all GPUs can still talk to each other, and so it is important to restrict the activity of the tenant to the frontend servers and to have system hardening procedures in place to prevent unauthorized access directly to the GPU servers.

Using network-level segregation is the preferred method for GPU isolation for multitenancy in an AI/ML cluster.

3 Hardware and optics

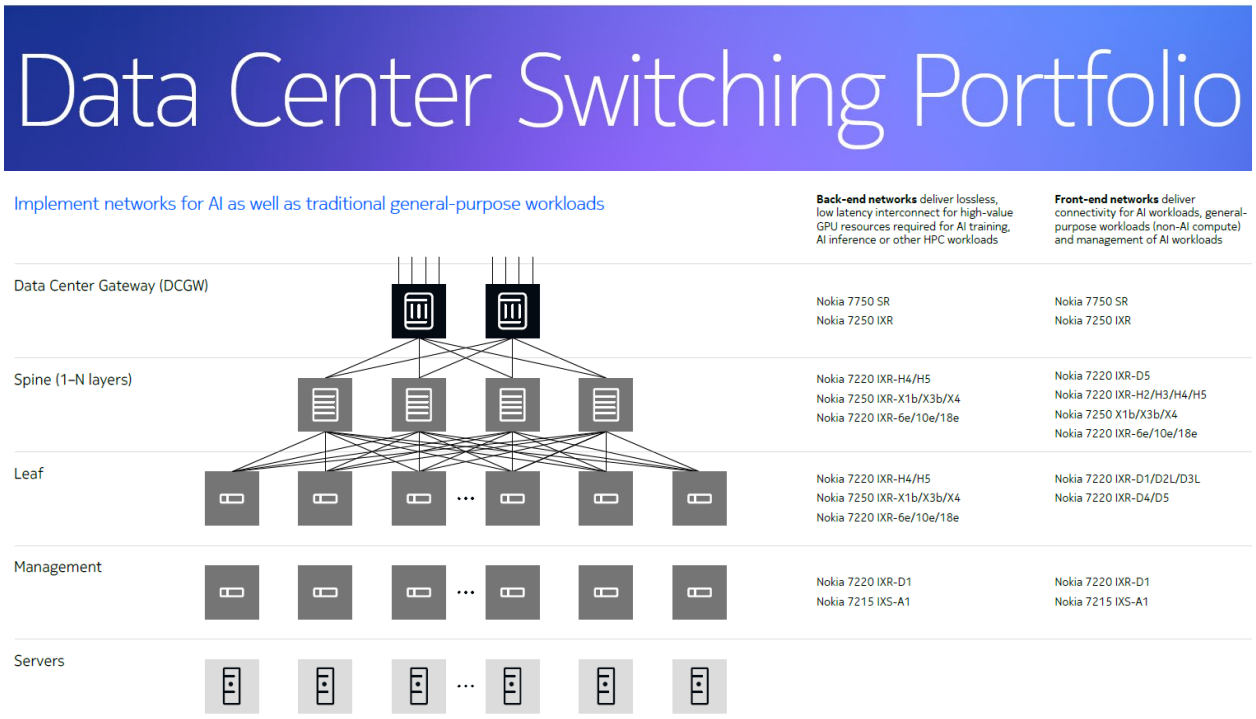
The following table shows the hardware that has been used in the validation process. The selection of optics was determined by laboratory availability.

Connection					Connectivity option
Endpoints			Speed	Port type	DAC, AOC, or transceiver and cable
1	A	NVIDIA BlueField-3 B3140H	400G	QSFP112	Transceiver: • 3HE19539AARA01 • QSFP112 400G-DR4 Cable: MPO-12
	B	7220 IXR-H4-32D	400G	QSFP-DD	Transceiver: • 3HE15271AARA01 • QSFP56DD 400G-DR4 Cable: MPO-12
2	A	NVIDIA BlueField-3 B3140H	400G	QSFP112	AOC: • Eoptolink • EOLD-8HG-PCT-05C1 • 800G QSFPDD to 2 x 400G QSFP112 Breakout AOC 5m
	B	7220 IXR-H5-32D	800G	QSFP-DD	
3	A	NVIDIA BlueField-3 B3140H	400G	QSFP112	AOC: • Eoptolink • EOLD-8HG-PCT-05C1 • 800G QSFPDD to 2 x 400G QSFP112 Breakout AOC 5m
	B	7220 IXR-H5-64D	800G	QSFP-DD	
4	A	7220 IXR-H4-32D	400G	QSFP-DD	Transceiver: • 3HE21007AARA01 • QSFPDD 400G-SR4

					Cable: MPO-12
	B	7220 IXR-H5-32D	400G	QSFP-DD	Transceiver: • 3HE21007AARA01 • QSFPDD 400G-SR4 Cable: MPO-12
5	A	7220 IXR-H4-32D	400G	QSFP-DD	Transceiver: • Eoptolink • EOLD-854HG-01M46 • 400G-VR4 Cable: MPO-12
	B	7220 IXR-H5-640	800G	OSFP	Transceiver: • Eoptolink • EOLO-858HG-01-D • 2 x 400G-VR4 OSFP Cable: MPO-12
6	A	7220 IXR-H5-32D	800G	QSFP-DD	DAC: • LUXSHARE-TECH • LQ8DD020-SD-R • 800G DAC
	B	7220 IXR-H5-32D	800G	QSFP-DD	
7	A	7220 IXR-H5-32D	800G	QSFP-DD	AOC: • Eoptolink • EOLO-8HG-PCT-05G • 800G OSFP to QSFP-DD 5m
	B	7220 IXR-H5-640	800G	OSFP	
8	A	7220 IXR-D5	400G	QSFP-DD	Transceiver: • Eoptolink • EOLD-854HG-01M46 • 400G-VR4 Cable: MPO-12
	B	7220 IXR-H5-640	800G	OSFP	Transceiver: • Eoptolink • EOLO-858HG-01-D • 2 x 400G-VR4 OSFP Cable: MPO-12
9	A	AOC-S200G-B2C BCM57608	200G	QSFP56	AOC: • Eoptolink • EOLD-4HG-PCT-05C45 400G • QSFP56-DD to 2 x 200G QSFP56 5m
	B	7220 IXR-D5	400G	QSFP56-DD	
10	A	AOC-CX7AH0078-DTZ	200G	QSFP112	DAC: • Amphenol • NDYRYH-C105 • 400G QSFP-DD to 2 x 200G QSFP-DD
	B	7220 IXR-D5	400G	QSFP-DD	
11	A	Intel Ethernet Controller XL710	40G	QSFP+	Transceiver: • 3HE07928AAAA01 • QSFP+ 40G-SR4 Cable: MPO-12
	B	7220 IXR-D5	40G	QSFP+	Transceiver:

					• 3HE07928AAAA01 • QSFP+ 40G-SR4 Cable: MPO-12

4 Nokia portfolio



Shallow Buffer Switches

- Fixed-configuration, high-performance 7220 Interconnect Router (IXR) platforms for data center leaf, spine, front-end network and back-end network deployments
- Flexible native and breakout options with support for 800GE, 400GE, 200GE, 100GE, 50GE, 40GE, 25GE, 10GE and GE interfaces
- Universal connectors enables seamless integration of a broad range of compatible optics, including 800G QSFP-DD and 800G OSFP optics
- Multiple chassis variants with system capacities from 88 Gb/s to 51.2 Tb/s
- Hot-swappable and redundant power supplies and fans
- Powered by SR Linux and SONiC (on specific platforms)

7220 IXR-D1
- 88 Gb/s (FD); fixed; 1RU
- 4 x 10G SFP+
- 48 x 10/100/1000M RJ45

7220 IXR-D2L
- 2.0 Tb/s (FD); fixed; 1RU
- 8 x 100G QSFP28
- 48 x 25G SFP28
- 2 x 10G SFP+

7220 IXR-D3L
- 3.2 Tb/s (FD); fixed; 1RU
- 32 x 100G QSFP28
- 2 x 10G SFP+

7220 IXR-D4
- 6.0 Tb/s (FD); fixed; 1RU
- 8 x 400G QSFP-DD
- 28 x 100G QSFP28

7220 IXR-D5
- 12.8 Tb/s (FD); fixed; 1RU
- 32 x 400G QSFP-DD
- 2 x 10G SFP+

7220 IXR-H2
- 12.8 Tb/s (FD); fixed; 4RU
- 128 x 100G QSFP28

7220 IXR-H3
- 12.8 Tb/s (FD); fixed; 1RU
- 32 x 400G QSFP-DD
- 2 x 10G SFP+

7220 IXR-H4-32D
- 12.8 Tb/s (FD); fixed; 1RU
- 32 x 400G QSFP-DD
- 1 x 10G SFP+

7220 IXR-H4
- 25.6 Tb/s (FD); fixed; 2RU
- 64 x 400G QSFP-DD
- 2 x 10G SFP+

7220 IXR-H5-32D
- 25.6 Tb/s (FD); fixed; 1RU
- 32 x 800G QSFP-DD
- 2 x 10G SFP+

7220 IXR-H5-64D
- 51.2 Tb/s (FD); fixed; 2RU
- 64 x 800G QSFP-DD
- 2 x 10G SFP+

7220 IXR-H5-64O
- 51.2 Tb/s (FD); fixed; 2RU
- 64 x 800G OSFP
- 2 x 10G SFP+

Deep Buffer Switches

- Terabit-scale, modular and fixed 7250 Interconnect Router (IXR) platforms designed for data center spine, back-end network and WAN deployments
- Industry-leading design and density with taller line card pitch and honeycomb mesh air intakes
- Modular platforms use high-quality orthogonal direct cross-connect with no midplane connectors to limit system lifespan
- Designed for upgradability, with fabric and cooling tuned for ultra-low power consumption today and tomorrow
- Flexible native and breakout options with support for 800GE, 400GE, 200GE, 100GE, 50GE, 40GE, 25GE and 10GE interfaces
- Universal connectors enables seamless integration of a broad range of compatible optics, including 800G QSFP-DD, OSFP and ZR/ZR+ coherent optics
- Powered by SR Linux and SONiC (on specific platforms)

7250 IXR-X1b
- 7.2 Tb/s (FD); fixed; 1RU
- 24 x 100G QSFP28
- 12 x 400G QSFP-DD

7250 IXR-X4 QSFP-DD
- 25.6 Tb/s (FD); fixed; 1RU
- 32 x 800G QSFP-DD

7250 IXR-X3b
- 14.4 Tb/s (FD); fixed; 1RU
- 36 x 400G QSFP-DD

7250 IXR-X4 OSFP
- 25.6 Tb/s (FD); fixed; 1RU
- 32 x 800G OSFP

7250 IXR-6e
- 115.2 Tb/s (FD); 10RU
- 4 slots, 28.8 Tb/s (FD) each
- 144 x 800G QSFP-DD
- 144 x 800G OSFP
- 288 x 400G QSFP-DD
- 288 x 400G OSFP
- 1152 x 100G QSFP28

7250 IXR-10e
- 230.4 Tb/s (FD); 16RU
- 8 slots, 28.8 Tb/s (FD) each
- 288 x 800G QSFP-DD
- 288 x 800G OSFP
- 576 x 400G QSFP-DD
- 576 x 400G OSFP
- 2304 x 100G QSFP28

7250 IXR-18e
- 460.8 Tb/s (FD); 35RU
- 16 slots, 28.8 Tb/s (FD) each
- 576 x 800G QSFP-DD
- 576 x 800G OSFP
- 1152 x 400G QSFP-DD
- 1152 x 400G OSFP
- 4608 x 100G QSFP28

Management Switches

- Fixed-configuration 7215 Interconnect System (IXS) platform designed for data center fabric management connectivity
- Integrated redundant fans and PSUs
- Powered by SR Linux and community SONiC

7215 IXS-A1
- 88 Gb/s (FD); fixed; 1RU
- 4 x 10G SFP+
- 48 x 10/100/1000M RJ45

Service Router Linux (SR Linux)

SR Linux

- A Linux®-based NOS that enables scalability, flexibility and efficiency
- Open, extensible architecture built around model-driven management and modern interfaces
- Field-proven routing protocol stacks from the Nokia Service Router Operating System (SR OS)
- Industry-leading streaming telemetry framework provides ubiquitous data access
- State-of-the-art NetOps Development Kit (NDK) for customized network agents and applications

Community SONiC

SONiC

- SONiC® open-source NOS leverages the strength of a large ecosystem and community
- Brings choice and flexibility to data center and cloud environments
- Supported on specific Nokia data center platforms for a broad range of deployment roles

Event-driven Automation (EDA)

- Automate the entire data center network lifecycle for consistent and reliable performance
- Abstracts the complexity of multivendor networks with exceptional insight into network operations
- Provision and monitor data center network in real time to ensure it operates as expected
- Built on Kubernetes, leveraging a vast open-source ecosystem to lower risks and barriers to entry
- Compatible with various tools and public clouds for a versatile integration framework

3HE-21918-AAAA-TQZZA

5 Reference architecture and network orchestration

5.1 High-level overview

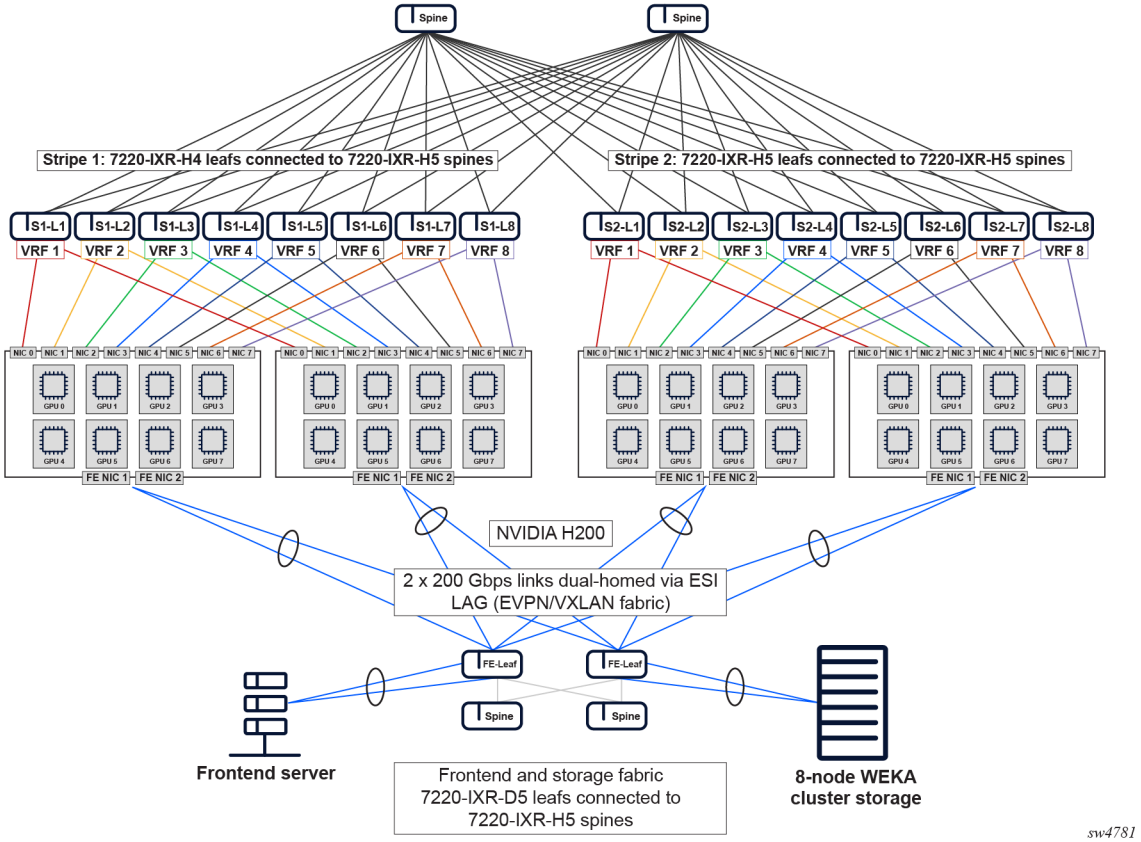


Figure 28. **HLD – Nokia hybrid training and inference cluster design**

Figure 28 shows the reference architecture for the Nokia Validated Design that is described in this document. It is a rail-optimized, two-stripe design hosting four Lenovo SR685v3 servers that have eight NVIDIA H200 GPUs per server.

The design has two main segments under test: the backend compute stripes consisting of eight 7220-IXR-H4 switches (Stripe 1) and eight 7220-IXR-H5 switches (Stripe 2) dual-homed to 7220-IXR-H5 spines.

The combined frontend and storage fabric consists of two 7220-IXR-D5 leafs dual-homed to two 7220-IXR-H5 spine switches. It has an eight-node WEKA cluster storage which is dual-homed to both leafs as well as a frontend-headend server dual-homed to the fabric.

5.2 Design considerations

5.2.1 Backend design

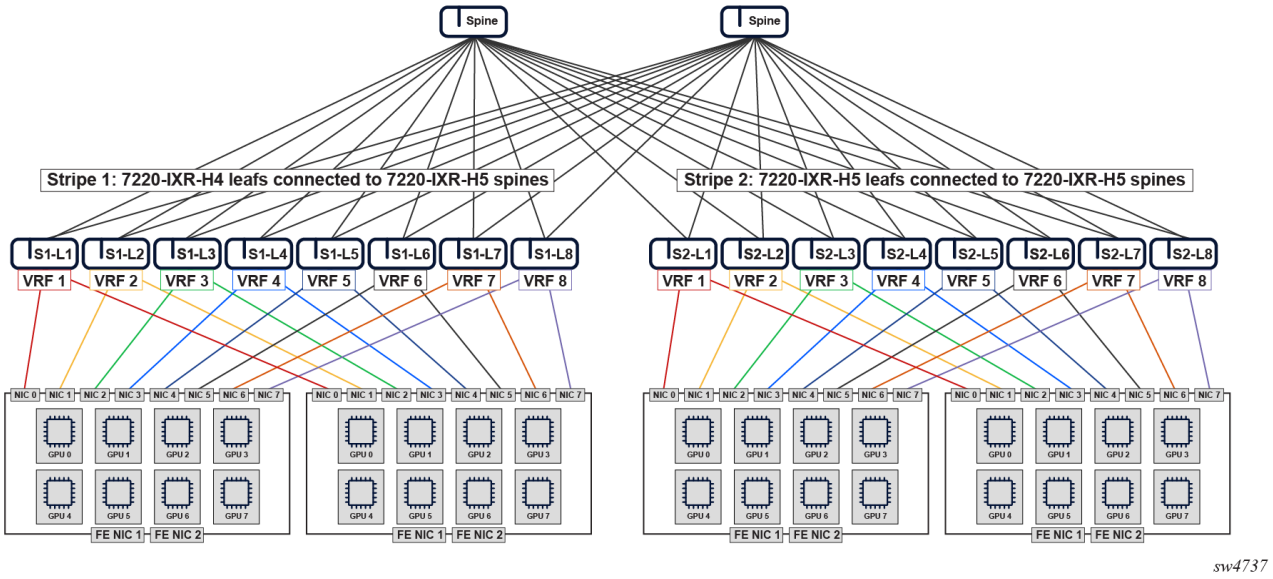


Figure 29. **Backend compute fabric**

- The backend fabric consists of two stripes of eight leafs each interconnected via two spines.
- Two servers have been connected to each stripe so that both intra- and inter-stripe communication and performance can be demonstrated.
- eBGP over IPv6 link local addressing is used for the leaf-to-spine connectivity. It is an IP fabric.
- The GPU direct ports connecting to the leaf have IPv6 unique local addresses, which are advertised to all eBGP peers.
- Each leaf is considered a rail and every link that is part of that leaf is in a VRF, as shown in the diagram above. The rail is extended across the stripes and the routes between rails (VRFs) are leaked across the spine onto the VRF on the corresponding rail on the other stripe.
- This design can be scaled by replicating the stripes as-is, and the cluster size depends on the port radix and number of spines in use.

5.2.2 Collapsed frontend and storage design

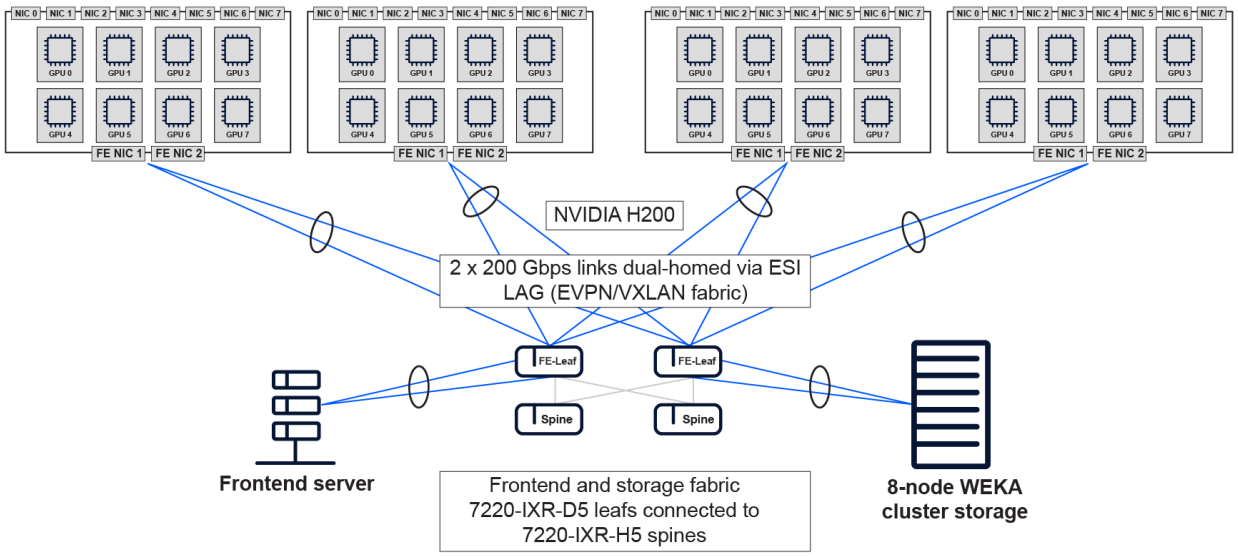


Figure 30. **Combined backend storage and frontend fabric**

The design considerations for this architecture are as follows:

- This validated design has a combined storage and frontend fabric where the dedicated storage nodes, the frontend ports on the GPU servers, and the frontend servers share the same set of leafs.
- For the purposes of this exercise, the fabric is a 2x2 7220-IXR-D5 fabric since there are only 2 × 200 Gbps links on each GPU server. However, this fabric can be scaled by adding more leaf and spine nodes as per customer requirements.
- This is an EVPN/VXLAN fabric with a single eBGP session over IPv6 link local addressing with dual address families (IP and EVPN) being carried over the same session.
- The main reason that an EVPN/VXLAN fabric is used is because all the links in the fabric are Layer 2- and dual-homed with an ESI-LAG. The storage and frontend servers are using a Layer 2-bonded link with a virtual IP for communication for the entire cluster.
- An end-to-end Layer 3 fabric can also be created without the use of EVPN/VXLAN.
- To have separate fabrics for the frontend and storage servers, one link each from the front-end ports of the GPU servers must be dedicated to frontend and to storage; however, this configuration does not provide host-level redundancy to the servers and is more prone to failure.

5.2.3 Multitenant reference design

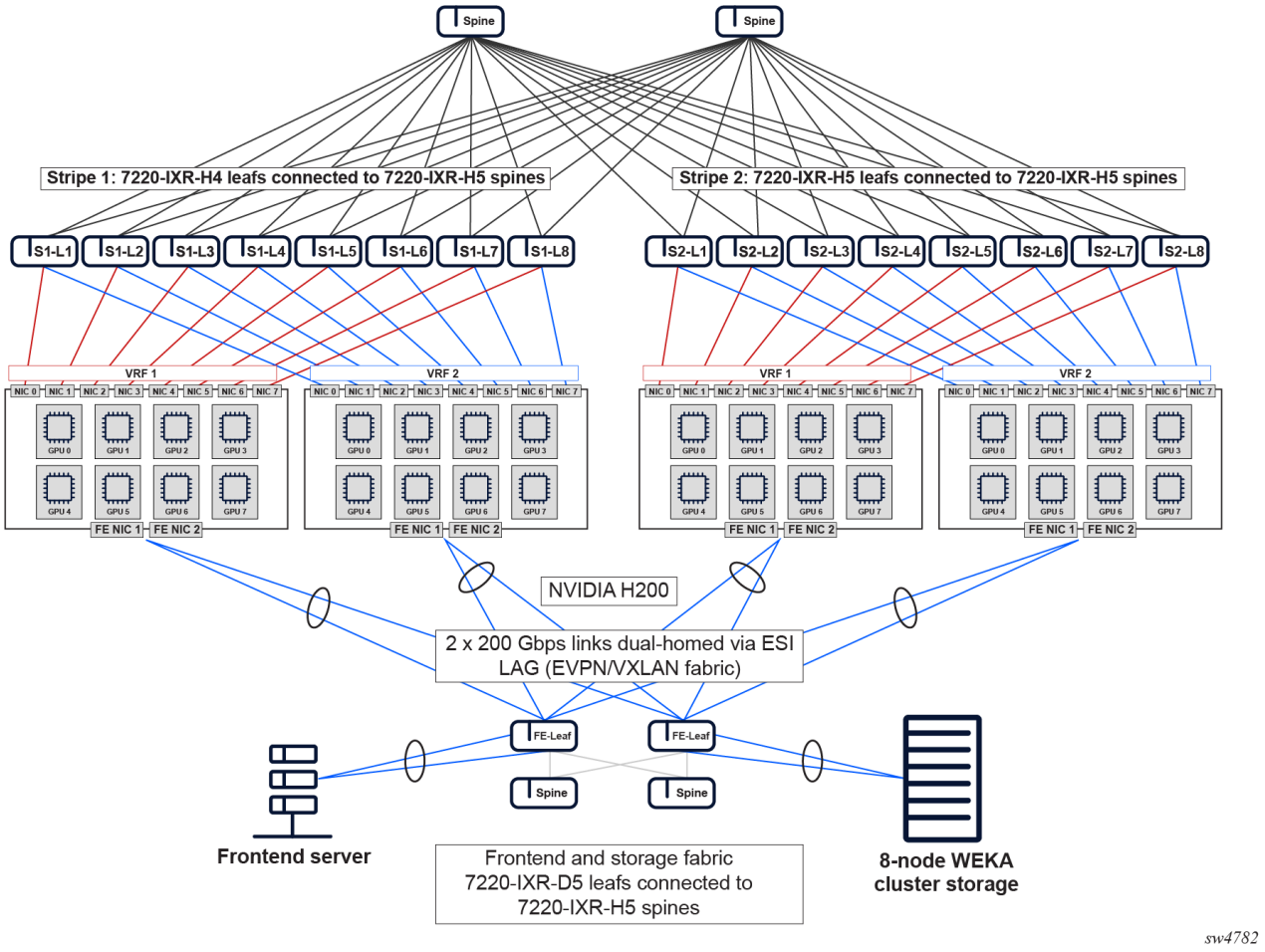


Figure 31. **Multitenant reference design**

- The validated method for creating a multitenant design is the server isolation method, which is the only network-controlled multitenancy that is possible in AI/ML clusters due to internal PXN switching on GPU servers.
- In this method, all the ports in a server as placed in a single VRF and the route extension only happens between the VRFs that are allocated to a particular tenant.
- The availability of devices to each tenant is static and can be increased by adding servers to the tenant VRF.
- The fabric can be an IP or an EVPN fabric based on customer preference. We have chosen to validate an IP fabric for single- and multi-tenant environments.

6 Compute server orchestration

6.1 Server resource specifications

Component	Specification
CPU	2 x AMD EPYC 9654 (96 cores, 192 threads each)
Memory	24 x 96 GB TruDDR5-4800 (total 2.3 TB)
Disk storage	2 x 1.9 TB NVMe drives
Networking	8 x 400G BlueField-3 NIC 1 x 2-port 200G CX-7 NIC 1 x 4-port 1G Intel I350-T4 OCPNIC
GPU	8 x NVIDIA H200

6.2 Lenovo SR685a V3 – NVIDIA H200 server orchestration

This section outlines the NVIDIA-specific orchestration steps that must be performed to have the servers updated to deploy the NCCL collective and workloads for benchmarking.

6.2.1 NVIDIA H200 system optimization

The GPU server must be optimized before running any benchmarking, acceptance, health, and stress tests. It is important to follow the guidelines to optimize the system by checking the BIOS, GRUB, and operating system settings.

6.2.1.1 BIOS settings

BIOS setting location (under the UEFI Setup tab)	Parameter	Value
System Settings	PCI 64-Bit Resource Allocation	Enabled
System Settings/Devices and I/O Ports	PCIe Gen Speed Selection/Slot10	Gen5
System Settings/Operating Modes	Choose Operating Mode	Maximum Performance

6.2.1.2 GRUB settings

In any modern Linux distribution, the /etc/default/grub file is used to configure GRUB (bootloader). In this file, the string assigned to GRUB_CMDLINE_LINUX is the command line parameters that Linux uses during boot, including the following:

- **iommu=pt**
  The iommu=pt setting enables IOMMU pass-through mode. When in pass-through mode, the adapter does not need to use DMA translation to the memory, which can improve performance.
- **pci=noacs**

  Disables PCIe Access Control Services (ACS)
- **cgroup_enable=memory**

  Enables cgroup v1 memory controller for container memory monitoring of limits and statistics.
- **systemd.unified_cgroup_hierarchy=0**

  Forces legacy cgroup v1 hierarchy, avoiding v2 incompatibilities with NVIDIA GPU tools.

To modify the GRUB settings, perform the following steps:

1. Update GRUB to use the new modified configuration.

   ```
   nvidia@slate5:~$ cat /etc/default/grub
   …
   GRUB_CMDLINE_LINUX="cgroup_enable=memory
   systemd.unified_cgroup_hierarchy=1 pci=noacs iommu=pt"
   …
   ```
2. Update GRUB to use the new modified configuration.

   ```
   sudo update-grub
   ```
3. Reboot the server for the GRUB settings to take effect.

   ```
   Sudo reboot now
   ```
4. Verify the GRUB settings after reboot.

   ```
   cat /proc/cmdline
   ```

6.2.1.3 Operating system settings

1. Disable NUMA auto-balancing.

   ```
   sudo sh -c 'echo 0 > /proc/sys/kernel/numa_balancing'
   ```

```
cat /proc/sys/kernel/numa_balancing << value should be 0
```

2. Disable PCI ACS via the script. The script can be found here:

https://github.com/ROCm/cluster-networking/blob/main/general_scripts/dis_acs.sh

```
touch /etc/init.d/disable_acs.sh
chmod +x disable_acs.sh
```

3. Validate that ACS is disabled using the following command.

Note: None of the lines in this step should show “SrcValid+”.

```
sudo lspci -vvv | grep -i "acsctl"
```

4. Add the following **memlock** limits to /etc/security/limits.conf:

```
* soft memlock unlimited
* hard memlock unlimited
* soft nofile 1048576
* hard nofile 1048576
```

6.2.2 NVIDIA GPU drivers and packages

The benchmark setup is built on a layered software stack from NVIDIA and other packages designed to ensure performance and is scalable across single and multi-node configurations. The software stack defined here starts from GPU drivers and extends to all the communication libraries across multi-nodes.

6.2.2.1 NVIDIA GPU driver installation

The NVIDIA GPU driver provides the foundational interface between the operating system and GPU hardware. It is responsible for device initialization and installation to bring up the GPU. The driver must be properly aligned with CUDA version to ensure the compatibility and feature support including NVLink and GPUDirect RDMA capabilities.

The installation is shown in section 6.2.2.2 using Debian package manager. We have used open kernel module of NVIDIA kernel module.

6.2.2.2 CUDA Toolkit

The CUDA Toolkit provides the runtime libraries, development tools, and performance-optimized libraries required for GPU accelerated workloads. It includes core components such as CUDA runtime, cuDNN, NVCC compiler, and so on. CUDA Toolkit ensures that applications can efficiently executes the job and leverage GPU memory bandwidth and parallelism.

The toolkit version must match the driver version.

For more information, see the NVIDIA CUDA Toolkit page:

https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&Distribution=Debian&target_version=13&target_type=deb_network

Installation:

1. Set up the repository.

```
wget https://developer.download.nvidia.com/compute/cuda/repos/debian13/x86_64/cuda-keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt-get update
sudo apt-get -y install cuda-toolkit-13-1
```

2. Install NVIDIA open kernel modules.

```
sudo apt-get install -y nvidia-open
```

3. Install NVIDIA CUDA drivers.

```
sudo apt-get install -y cuda-drivers
```

4. Verify the installation.

```
nvidia@slate8:~$ nvidia-smi
Tue Feb 24 20:05:26 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 590.48.01              Driver Version: 590.48.01      CUDA Version: 13.1     |
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |                  MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H200                    On  |   00000000:03:00.0 Off |                    0 |
| N/A   28C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA H200                    On  |   00000000:23:00.0 Off |                    0 |
| N/A   25C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA H200                    On  |   00000000:43:00.0 Off |                    0 |
| N/A   27C    P0             72W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   3  NVIDIA H200                    On  |   00000000:63:00.0 Off |                    0 |
| N/A   25C    P0             74W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   4  NVIDIA H200                    On  |   00000000:83:00.0 Off |                    0 |
| N/A   29C    P0             74W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   5  NVIDIA H200                    On  |   00000000:A3:00.0 Off |                    0 |
| N/A   26C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   6  NVIDIA H200                    On  |   00000000:C3:00.0 Off |                    0 |
| N/A   28C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   7  NVIDIA H200                    On  |   00000000:E3:00.0 Off |                    0 |
| N/A   28C    P0             72W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

6.2.2.3 NVIDIA Fabric Manager and Persistence Mode

NVIDIA Fabric Manager is required in the system for multi-GPU interconnect topologies. It initializes and manages high-speed GPU communication fabrics, whereas NVIDIA Persistence Mode keeps GPUs initialized even when no active workloads are running, reducing initialization overhead

preventing repeated driver reload cycles. Together, NVIDIA Fabric Manager and Persistence Mode reduces latency during benchmark execution.

Fabric Manager installation

```
sudo apt-get install -y nvidia-fabricmanager-590
```

Verification

```
nvidia@slate8:~$ nv-fabricmanager --version
Fabric Manager version is : 590.48.01
```

```
nvidia@slate8:~$ systemctl status nvidia-fabricmanager
● nvidia-fabricmanager.service - NVIDIA fabric manager service
     Loaded: loaded (/usr/lib/systemd/system/nvidia-fabricmanager.service; enabled;
preset: enabled)
     Active: active (running) since Thu 2026-02-12 14:09:25 PST; 1 week 5 days ago
   Main PID: 6214 (nv-fabricmanage)
      Tasks: 18 (limit: 629145)
     Memory: 14.1M (peak: 28.5M)
        CPU: 13min 3.711s
     CGroup: /system.slice/nvidia-fabricmanager.service
             └─6214 /usr/bin/nv-fabricmanager -c
/usr/share/nvidia/nvswitch/fabricmanager.cfg
```

Persistence Mode

No additional packages are required; Persistence Mode must be enabled on the GPUs.

```
sudo nvidia-smi -pm 1
```

```
nvidia@slate8:~$ systemctl status nvidia-persistenced
● nvidia-persistenced.service - NVIDIA Persistence Daemon
     Loaded: loaded (/usr/lib/systemd/system/nvidia-persistenced.service; enabled; preset:
enabled)
     Active: active (running) since Thu 2026-02-12 14:09:24 PST; 1 week 5 days ago
   Main PID: 4943 (nvidia-persiste)
      Tasks: 1 (limit: 629145)
     Memory: 864.0K (peak: 1.7M)
        CPU: 94ms
     CGroup: /system.slice/nvidia-persistenced.service
             └─4943 /usr/bin/nvidia-persistenced --user nvidia-persistenced
```

Check the column with the “Persistence-M” heading for Persistence-M and that the GPUs are in On mode.

```
nvidia@slate8:~$ nvidia-smi
Tue Feb 24 20:05:26 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 590.48.01              Driver Version: 590.48.01      CUDA Version: 13.1     |
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H200                    On  |   00000000:03:00.0 Off |                    0 |
| N/A   28C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA H200                    On  |   00000000:23:00.0 Off |                    0 |
| N/A   25C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA H200                    On  |   00000000:43:00.0 Off |                    0 |
| N/A   27C    P0             72W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
~~ snip~~
```

6.2.2.4 NVIDIA NCCL

NVIDIA Collective Communications Library (NCCL) is a package from NVIDIA that is optimized for multi-GPU and multi-node collective operations such as AllReduce, Broadcast, ReduceScatter, and so on. It leverages available interconnects such as NVLink, PCIe, and RDMA-based networking (RoCE) to maximize bandwidth and minimize latency.

Installation

1. Clone the repository.

```
Git clone https://github.com/NVIDIA/nccl-tests.git
```

2. Install the pre-requisites.

```
sudo apt-get install -y libnccl2 libnccl-dev
```

3. Build the MPI executable with `make`.

```
make clean
make -j MPI=1 CC=mpicc CXX=mpicxx
```

Verification

```
nvidia@slate8:~$ dpkg -l | grep nccl
ii  libnccl-dev                          2.29.2-1+cuda13.1                          amd64
NVIDIA Collective Communication Library (NCCL) Development Files

ii  libnccl2                             2.29.2-1+cuda13.1                          amd64
NVIDIA Collective Communication Library (NCCL) Runtime
```

```
nvidia@slate8:~$ dpkg -s libnccl2 | grep Version
Version: 2.29.2-1+cuda13.1
```

6.2.2.5 MPI Runtime (Open MPI)

Message Passing Interface (MPI) provides the runtime environment for executing parallel computing applications. An MPI runtime such as Open MPI provides the process orchestration and communication framework for distributed workloads across multiple nodes. It manages process launching, synchronization, and data exchange and works in conjunction with other transport layers to optimize the network performance.

In GPU clusters, MPI coordinates multi-node execution while enabling NCCL and RDMA transports to operate collectively.

Installation

```
sudo apt-get update
sudo apt install -y openmpi-bin libopenmpi-dev
```

Verification

```
nvidia@slate8:~$ mpirun --version
mpirun (Open MPI) 4.1.6

Report bugs to http://www.open-mpi.org/community/help/
```

6.2.2.6 NVIDIA Container Toolkit

The NVIDIA container Toolkit enables GPU acceleration within containerized environments such as Docker. It provides runtime hooks that expose GPU devices, drivers, and CUDA libraries inside containers without embedding drivers into container images.

Installation

NVIDIA Container Toolkit.

```
sudo apt update
sudo apt-get install -y nvidia-container-toolkit nvidia-container-toolkit-base libnvidia-
container1 libnvidia-container-tools
```

Configure the Docker runtime.

```
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

Verification

```
nvidia@slate7:~$ nvidia-container-toolkit --version
NVIDIA Container Runtime Hook version 1.18.2
```

6.2.3 BlueField-3 400G NIC installation and configuration

This validated design uses the 400 Gb BlueField-3 NICs as GPU-attached NICs. Other GPU NIC models are supported but require different configuration steps, so these instructions apply only to BlueField-3 NICs. This section describes how to install the necessary RDMA drivers and configure QoS on the NICs using Data Center Infrastructure-on-a-Chip Architecture (DOCA), which automatically installs all required packages.

DOCA installation

Instructions for installing DOCA version 3.2.1 can be found at:

https://developer.nvidia.com/doca-downloads?deployment_platform=Host-Server

```
wget https://www.mellanox.com/downloads/DOCA/DOCA_v3.2.1/host/doca-host_3.2.1-044000-
25.10-ubuntu2404_amd64.deb
sudo dpkg -i doca-host_3.2.1-044000-25.10-ubuntu2404_amd64.deb
sudo apt-get update
sudo apt-get -y install doca-all
```

Verify driver installation

1. Verify if DOCA has been installed. Search for “doca-ofed”, as this package installs the RDMA drivers (for example, mlx5_core, and mlx5_ib).

```
nvidia@slate5:~$ dpkg -l | grep doca
ii  doca-all                              3.2.1-044000
amd64        doca-all meta-package
ii  doca-apsh-config                      3.2.1025-1
amd64        Data Center on a Chip Architecture (DOCA) Tool
ii  doca-bench                            3.2.1025-1
amd64        Data Center on a Chip Architecture (DOCA) Tool
ii  doca-caps                             3.2.1025-1
amd64        Data Center on a Chip Architecture (DOCA) Tool
ii  doca-comm-channel-admin               3.2.1025-1
amd64        Data Center on a Chip Architecture (DOCA) Tool
<snipped>
```

```
ii  doca-host                              3.2.1-044000-25.10-ubuntu2404
amd64        Software package including drivers, libraries and tools installed on the host
server to support NVIDIA networking platforms
```

2. Verify that RDMA drivers are loaded and functional. Check that ib_uverbs depends on the nvidia_peermem kernel module (for GPU peer memory access in RDMA/NCCL) as shown in **`lsmod`** output. Ensure all kernel modules are running the same version on all servers by using the **`modinfo`** command.

```
nvidia@slate5:~$ lsmod | grep mlx5
mlx5_ib               552960  0
mlx5_fwctl             16384  0
macsec                 77824  1 mlx5_ib
fwctl                  12288  1 mlx5_fwctl
mlx5_core            3129344  2 mlx5_fwctl,mlx5_ib
mlxdevm               540672  1 mlx5_core
mlxfw                  36864  1 mlx5_core
psample                16384  1 mlx5_core
tls                   155648  2 bonding,mlx5_core
pci_hyperv_intf        12288  1 mlx5_core
ib_uverbs             200704  3 nvidia_peermem,rdma_ucm,mlx5_ib
ib_core               520192  6 rdma_cm,iw_cm,rdma_ucm,ib_uverbs,mlx5_ib,ib_cm
```

3. Verify that the RDMA driver detects all eight NIC interfaces, each showing an active state, link UP, and 400G port speed. The following output maps the RDMA interfaces (displayed as mlx5_x) to their corresponding physical interfaces (ensxf0np0)

```
nvidia@slate5:~$ rdma link show
link mlx5_0/1 state ACTIVE physical_state LINK_UP netdev ens6f0np0
link mlx5_1/1 state ACTIVE physical_state LINK_UP netdev ens5f0np0
link mlx5_2/1 state ACTIVE physical_state LINK_UP netdev ens8f0np0
link mlx5_3/1 state ACTIVE physical_state LINK_UP netdev ens7f0np0
link mlx5_4/1 state ACTIVE physical_state LINK_UP netdev ens2f0np0
link mlx5_5/1 state ACTIVE physical_state LINK_UP netdev ens1f0np0
link mlx5_6/1 state ACTIVE physical_state LINK_UP netdev ens4f0np0
link mlx5_9/1 state ACTIVE physical_state LINK_UP netdev ens3f0np0
link mlx5_bond_0/1 state ACTIVE physical_state LINK_UP netdev ens10f0np0
nvidia@slate5:~$
nvidia@slate5:~$ ibstat -d mlx5_0
CA 'mlx5_0'
        CA type: MT41692
        Number of ports: 1
        Firmware version: 32.43.2026
        Hardware version: 1
        Node GUID: 0xb8e9240300b3acee
        System image GUID: 0xb8e9240300b3acee
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 400
                Base lid: 0
                LMC: 0
```

```
                SM lid: 0
                Capability mask: 0x00010000
                Port GUID: 0xbae924fffeb3acee
                Link layer: Ethernet
```

6.2.3.1 GPU NIC mapping and configuration

To route traffic efficiently, GPUs use the nearest NIC within the PCIe topology. To understand how each GPU is mapped to its corresponding NIC, mlx5_x interface, and NUMA node, refer to the following example, which illustrates the configuration for the NIC in slot 1.

Mapping the PCI address to the interface name: in this example, the NIC in slot 1 corresponds to ens1f0np0, though the specific interface name may vary depending on the system.

```
nvidia@slate5:/$ ethtool -i ens1f0np0 | grep -i bus
bus-info: 0000:a4:00.0
```

1. Map the interfaces to the NUMA nodes using either interface names or PCI addresses.
   - Based on interface name

```
nvidia@slate5:/$ cat /sys/class/net/ens1f0np0/device/numa_node
1
```

   - Based on interface PCI address

```
nvidia@slate5:/$ cat /sys/bus/pci/devices/0000\:a4\:00.0/numa_node
1
```

2. Map the physical interfaces to the RDMA interfaces.

```
nvidia@slate5:/$ rdma link show | grep ens1f0np0
link mlx5_5/1 state ACTIVE physical_state LINK_UP netdev ens1f0np0 link bnxt_re1/1 state
```

3. List all GPU PCI addresses. The same information can be retrieved with the **`nvidia-smi`** command.

```
nvidia@slate5:/$ sudo lspci -d 10de: | grep H200
03:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
23:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
43:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
63:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
83:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
a3:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
c3:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
e3:00.0 3D controller: NVIDIA Corporation GH100 [H200 SXM 141GB] (rev a1)
```

4. Match the GPU PCI addresses to the GPU-ID.

```
nvidia@slate5:/$ nvidia-smi
Tue Feb 24 12:12:30 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 590.48.01              Driver Version: 590.48.01      CUDA Version: 13.1     |
```

```
+-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |                  MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA H200                    On  |   00000000:03:00.0 Off |                    0 |
| N/A   25C    P0             72W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA H200                    On  |   00000000:23:00.0 Off |                    0 |
| N/A   24C    P0             72W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA H200                    On  |   00000000:43:00.0 Off |                    0 |
| N/A   27C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   3  NVIDIA H200                    On  |   00000000:63:00.0 Off |                    0 |
| N/A   26C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   4  NVIDIA H200                    On  |   00000000:83:00.0 Off |                    0 |
| N/A   27C    P0             71W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   5  NVIDIA H200                    On  |   00000000:A3:00.0 Off |                    0 |
| N/A   25C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   6  NVIDIA H200                    On  |   00000000:C3:00.0 Off |                    0 |
| N/A   27C    P0             74W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
|   7  NVIDIA H200                    On  |   00000000:E3:00.0 Off |                    0 |
| N/A   25C    P0             73W /  700W |       0MiB / 143771MiB |      0%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+
```

5. Match the PCI address of the NIC in slot 1 (identified above) with the GPU's PCI address. The GPU whose PCI path aligns most closely with the NIC shares the same PCIe bridge, indicating it is the nearest NIC and provides the shortest communication path. In the output below, we can see that GPU with PCI address 03:00.0 is closest to NIC ens1f0np0, which has PCI address a4:00.0.

```
nvidia@slate5:~$ lstopo

<snipped>

HostBridge
      PCIBridge
        PCIBridge
          PCIBridge
            PCI a3:00.0 (3D)
              CoProc(OpenCL) "opencl0d5"
          PCIBridge
            PCI a4:00.0 (Ethernet)
```

```
                Net "ens1f0np0"
                OpenFabrics "mlx5_5"
<snipped>
```

6. Repeat these steps for each NIC interface and collate the results.

NIC NUMA	NIC slot	PCI address	NIC BIOS name	RoCE NIC name	Closest GPU-ID	GPU PCI address
1	1	a4:00.0	ens1f0np0	mlx5_5	GPU5	a3:00.0
1	2	84:00.0	ens2f0np0	mlx5_4	GPU4	83:00.0
1	3	e4:00.0	ens3f0np0	mlx5_9	GPU7	e3:00.0
1	4	c4:00.0	ens4f0np0	mlx5_6	GPU6	c3:00.0
0	5	24:00.0	ens5f0np0	mlx5_1	GPU1	23:00.0
0	6	04:00.0	ens6f0np0	mlx5_0	GPU0	03:00.0
0	7	6a:00.0	ens7f0np0	mlx5_3	GPU3	63:00.0
0	8	44:00.0	ens8f0np0	mlx5_2	GPU2	43:00.0

6.3 BlueField-3 NIC QoS configuration

This section explains how to configure PFC, ECN, and traffic classification on Bluefield-3 NIC cards. These settings must match the configuration of the backend network. By default, the NIC cards don't have PFC or scheduling configured, so they must be explicitly configured. The values shown in the following table are applied to the NICs.

Type of traffic	Classification DSCP	Queue	PFC	Scheduling
All other	0	0	-	ETS 10%
RoCEv2	26	3	Enabled	ETS 90%
Control traffic (CNP)	48	6	-	Strict priority

The BlueField-3 NICs can be configured using the **`mlnx_qos`** and **`cma_roce_tos`** commands, which are included in the DOCA package installation described in section 6.2.3.
The following script demonstrates how to configure the DSCP mapping, PFC settings, and traffic scheduling for the ens1f0np0 interface. Apply the same configuration to all remaining NIC interfaces in the system.

```
# Trust DSCP, cable length between GPU NIC and switch port (measure yours)
sudo mlnx_qos -i ens1f0np0 --trust dscp --cable_len=5

# DSCP-to-prio (26→3 RoCE, 48→6 CNP; others→0)
sudo mlnx_qos -i ens1f0np0 --dscp2prio set,26,3
sudo mlnx_qos -i ens1f0np0 --dscp2prio set,48,6

# PFC only on prio3/TC3
```

```
sudo mlnx_qos -i ens1f0np0 --pfc 0,0,0,1,0,0,0,0

# Prio-to-TC (TC0 bulk, TC3 RoCE, TC6 CNP)
sudo mlnx_qos -i ens1f0np0 --prio_tc 0,0,0,3,0,0,6,0

# ETS scheduler: prio6=strict SP, TCs 0/3 10%/90%
sudo mlnx_qos -i ens1f0np0 --tsa ets,ets,ets,ets,ets,ets,strict,ets –tcbw 10,0,0,90,0,0,0,0
```

Configure DSCP value 26 for RoCEv2 traffic on RDMA interface mlx5_5.

```
# Set ToS=104 (DSCP 26) for RoCE traffic on RDMA interface level
sudo cma_roce_tos -d mlx5_5 -t 104

# Set ToS=104 (DSCP 26) for RoCE traffic on egress
echo 104 | sudo tee /sys/class/infiniband/mlx5_5/tc/1/traffic_class > /dev/null
```

ECN is already enabled by default for all interfaces, but can be verified as follows.

```
# Shows the DSCP marking used for CNP (Congestion Notification Packet) traffic.
nvidia@slate5:/$ cat /sys/class/net/ens1f0np0/ecn/roce_np/cnp_dscp
48

# Displays the Layer-2 802.1p priority assigned to CNP packets.
nvidia@slate5:/$ cat /sys/class/net/ens1f0np0/ecn/roce_np/cnp_802p_prio
6
# Confirms that ECN is enabled for traffic with priority class 3
nvidia@slate5:/$ cat /sys/class/net/ens1f0np0/ecn/roce_np/enable/3
1
```

Verify if all configuration has been applied.

```
nvidia@slate5:~ $ sudo mlnx_qos -i ens1f0np0
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
        prio:0 dscp:07,06,05,04,03,02,01,00,
        prio:1 dscp:15,14,13,12,11,10,09,08,
        prio:2 dscp:23,22,21,20,19,18,17,16,
        prio:3 dscp:26,31,30,29,28,27,25,24,
        prio:4 dscp:39,38,37,36,35,34,33,32,
        prio:5 dscp:47,46,45,44,43,42,41,40,
        prio:6 dscp:48,55,54,53,52,51,50,49,
        prio:7 dscp:63,62,61,60,59,58,57,56,
Receive buffer size (bytes): 19872,588672,0,0,0,0,0,0,max_buffer_size=4151520
Cable len: 5
PFC configuration:
        priority    0   1   2   3   4   5   6   7
        enabled     0   0   0   1   0   0   0   0
        buffer      0   0   0   1   0   0   0   0
tc: 0 ratelimit: unlimited, tsa: ets, bw: 10%
         priority:  0
         priority:  1
         priority:  2
         priority:  4
         priority:  5
         priority:  7
tc: 3 ratelimit: unlimited, tsa: ets, bw: 90%
         priority:  3
```

```
tc: 6 ratelimit: unlimited, tsa: strict
        priority:  6
```

6.4 IP addressing

The IPv6 addresses for the GPU interfaces are configured through a shared Netplan YAML file. Each interface is assigned its own routing table with a default route pointing to the appropriate gateway. Source-based routing rules are added so that traffic originating from a specific interface uses only its corresponding routing table. This setup is necessary because a single default route in the main routing table would not allow the system to determine which interface outgoing packets should use. By separating the routing tables, each interface has a clearly defined path for its traffic, and all eight links can operate as independent network paths.

Both frontend and storage facing interfaces are configured in a LACP bond0 each with their own VLAN 100 and 200.

The following output shows the GPU server1 Netplan file for GPU NICs. Note that, for brevity, not all interfaces are shown.

```
network:
  version: 2
  ethernets:

<snipped>

    ens10f0np0: { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens10f1np1: { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens1f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens2f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens3f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens4f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens5f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens6f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens7f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
    ens8f0np0:  { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }

  bonds:
    bond0:
      interfaces:
        - ens10f0np0
        - ens10f1np1
      mtu: 9000
      parameters:
        mode: 802.3ad
        lacp-rate: fast
        mii-monitor-interval: 100
  vlans:
    ens1f0np0.1000: # GPU 1 NIC interface
      id: 1000
      link: ens1f0np0
      mtu: 9000
      accept-ra: false
```

```
      dhcp4: no
      dhcp6: no
      optional: true
      addresses:
        - "fd00:2:9:1:2:1:0:2/96"
      routes:
        - to: "::/0"
          via: "fd00:2:9:1:2:1:0:1"
          on-link: true
          metric: 10
          table: 101
      routing-policy:
        - from: "fd00:2:9:1:2:1:0:2/96"
          table: 101
    ens2f0np0.1000: # GPU 2 NIC interface
      id: 1000
      link: ens2f0np0
      mtu: 9000
      accept-ra: false
      dhcp4: no
      dhcp6: no
      optional: true
      addresses:
        - "fd00:2:10:1:2:1:0:2/96"
      routes:
        - to: "::/0"
          via: "fd00:2:10:1:2:1:0:1"
          on-link: true
          metric: 10
          table: 102
      routing-policy:
        - from: "fd00:2:10:1:2:1:0:2/96"
          table: 102

<snipped> # GPU3 to GPU8 have been left out for brevity

    bond0.100: # storage network
      id: 100
      link: bond0
      mtu: 9000
      addresses:
        - 192.168.1.68/24
    bond0.200: # frontend network
      id: 200
      link: bond0
      mtu: 9000
      addresses:
        - 192.168.2.68/24
```

7 WEKA storage orchestration

7.1 Server resource specifications

Component	Specification
CPU	1 x AMD Genoa 9454 (48 cores, 96 threads each)
Memory	12 x 32GB DDR5-5600 (total 384 GB per server)
Disk storage	File system: Micron 7450 PRO 960GB NVMe. 2 x 960 GB = 1.92 TB per server Cluster storage: Micron 7500 PRO 15.3 TB NVMe. 61.2 TB per server. 489.6 TB cluster-wide
Networking	2 x 200G ConnectX-7 2 x 25G Mellanox ConnectX-6 2 x 10G Intel X550

7.2 WEKA server optimization

By default, Supermicro storage servers do not ship with performance-optimized BIOS settings. WEKA provides a built-in utility, **bios_tool**, that allows you to view and configure BIOS settings on these servers. This tool performs the same actions you would carry out if you configured the BIOS manually.

The bios_tool uses two configuration files:

- host_config.yml — Specifies the list of hosts and their BMC login credentials.
- bios_setting.yml — Defines the BIOS settings you want to apply to the servers listed in host_config.yml. These settings are organized by server type.

Both configuration files, along with the bios_tool binary, are located in /opt/tools/bios_tool.

Note: You might need to run `git pull` under /opt/tools to pull the latest software.

Perform the following steps to optimize the WEKA server.

1. Define the hosts and credentials in host_config.yml.

```
hosts:
  - name: 192.168.1.197
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.198
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.199
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.200
```

```
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.201
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.202
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.203
    user: ADMIN
    password: MyPassword
  - name: 192.168.1.204
    user: ADMIN
    password: MyPassword
```

The default BIOS optimization settings can be found in bios_settings.yml. No changes are required, but you may review them for our server type, Supermicro AMD AS-1115CS-TNR.

```
<snipped>
Supermicro:
AMD:
     AS -1115CS-TNR:
     ACPISRATL3CacheAsNUMADomain_0099: Disabled
     DFCstates_7103: Disabled
     DeterminismControl_00F4: Manual
     GlobalC_stateControl_00CD: Disabled
     IOMMU_00EA: Disabled
     NUMANodesPerSocket_703E: NPS1
     PowerProfileSelection_00F8: Maximum IO Performance Mode
     SMTControl_00CB: Disabled
     '*':
     ACPISRATL3CacheAsNUMADomain: Disabled
     IOMMU: Disabled
     NUMANodesPerSocket: NPS1
     SMTControl: Disabled
     DFCstates: Disabled
     GlobalC_stateControl: Disabled
     DeterminismControl: Manual
     PowerProfileSelection: "Maximum IO Performance Mode"
<snipped>
```

2. Execute the following command to apply the new BIOS settings defined in bios_settings.yml.

Note: A reboot is required for the configuration to take effect.

```
[root@weka01 bios_tool]# ./bios_tool -c host_config.yml -b bios_settings.yml --fix
Opening sessions to hosts:
Connected to 100.116.161.197
```

```
Connected to 100.116.161.203
Connected to 100.116.161.202
Connected to 100.116.161.201
Connected to 100.116.161.200
Connected to 100.116.161.199
Connected to 100.116.161.198
Connected to 100.116.161.204
Checking BIOS settings on 100.116.161.197
100.116.161.197: BIOS setting ACPISRATL3CacheAsNUMADomain is Auto, but should be Disabled
100.116.161.197: BIOS setting DFCstates is Auto, but should be Disabled
100.116.161.197: BIOS setting DeterminismControl is Auto, but should be Manual
100.116.161.197: BIOS setting GlobalC_stateControl is Auto, but should be Disabled
100.116.161.197: BIOS setting IOMMU is Auto, but should be Disabled
100.116.161.197: BIOS setting NUMANodesPerSocket is Auto, but should be NPS1
100.116.161.197: BIOS setting PowerProfileSelection is High Performance Mode, but should
be Maximum IO Performance Mode
100.116.161.197: BIOS setting SMTControl is Auto, but should be Disabled
8 changes are needed on 100.116.161.197
Successfully set settings on host 100.116.161.197; System reboot required

Checking BIOS settings on 100.116.161.198
100.116.161.198: BIOS setting ACPISRATL3CacheAsNUMADomain is Auto, but should be Disabled
100.116.161.198: BIOS setting DFCstates is Auto, but should be Disabled
100.116.161.198: BIOS setting DeterminismControl is Auto, but should be Manual
100.116.161.198: BIOS setting GlobalC_stateControl is Auto, but should be Disabled
100.116.161.198: BIOS setting IOMMU is Auto, but should be Disabled
100.116.161.198: BIOS setting NUMANodesPerSocket is Auto, but should be NPS1
100.116.161.198: BIOS setting PowerProfileSelection is High Performance Mode, but should
be Maximum IO Performance Mode
100.116.161.198: BIOS setting SMTControl is Auto, but should be Disabled
8 changes are needed on 100.116.161.198
Successfully set settings on host 100.116.161.198; System reboot required
```

3. After the reboot, verify that all nodes have the same BIOS configuration, as follows. The output should indicate “The servers have identical BIOS settings”. If this is not the case, you must set the BIOS manually.

```
[root@weka01 bios_tool]# ./bios_tool --diff 100.116.161.197 100.116.161.204 -c
host_config.yml
Opening sessions to hosts:
Connected to 100.116.161.197
Connected to 100.116.161.204
The servers have identical BIOS settings
```

7.3 Network settings

It is crucial to define the storage network prior to configuring the storage cluster, as the cluster does not operate without reliable connectivity between the storage NICs.

1. Execute the following script to create a bond0 interface in active-active LACP mode using both 200G NICs and assign an IP address with an MTU of 9000.

```
[root@weka01 ~]# cat 00_bond_no_vlan.sh
#!/bin/bash
# Script to delete and recreate bond0 with static IP (no VLAN)

BOND_NAME="bond0"
BOND_SLAVES=("ens1f0np0" "ens1f1np1")
IP_ADDR="192.168.1.189/24"
GATEWAY="192.168.1.1"

echo "=== Deleting existing bond and related connections if they exist ==="
# Delete bond slaves if exist
for slave in "${BOND_SLAVES[@]}"; do
nmcli connection delete bond-slave-$slave 2>/dev/null
done

# Delete bond if exists
nmcli connection delete bond-$BOND_NAME 2>/dev/null

echo "=== Creating bond interface ==="
nmcli connection add type bond ifname $BOND_NAME bond.options
"mode=802.3ad,miimon=100,downdelay=5,updelay=10,xmit_hash_policy=layer3+4"

echo "=== Configuring bond IP and settings ==="
nmcli connection modify bond-$BOND_NAME ipv4.addresses $IP_ADDR
nmcli connection modify bond-$BOND_NAME ipv4.gateway $GATEWAY
nmcli connection modify bond-$BOND_NAME ipv4.method manual
nmcli connection modify bond-$BOND_NAME ipv6.method ignore
nmcli connection modify bond-$BOND_NAME 802-3-ethernet.mtu 9000

echo "=== Adding slave interfaces to bond ==="
for slave in "${BOND_SLAVES[@]}"; do
nmcli connection add type ethernet ifname $slave master $BOND_NAME mtu 9000
done

echo "=== Bringing up bond and slave interfaces ==="
nmcli connection up bond-$BOND_NAME
for slave in "${BOND_SLAVES[@]}"; do
nmcli connection up bond-slave-$slave
done

echo "=== Setup complete ==="
```

2. Configure all bond0 interfaces on all WEKA storage servers and verify connectivity of the storage network.

```
root@weka01 ~]# ping weka02.storage.ncse.io -c 5
PING weka02.storage.ncse.io (192.168.1.190) 56(84) bytes of data.
64 bytes from weka02.storage.ncse.io (192.168.1.190): icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from weka02.storage.ncse.io (192.168.1.190): icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from weka02.storage.ncse.io (192.168.1.190): icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from weka02.storage.ncse.io (192.168.1.190): icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from weka02.storage.ncse.io (192.168.1.190): icmp_seq=5 ttl=64 time=0.036 ms
```

```
--- weka02.storage.ncse.io ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4120ms
rtt min/avg/max/mdev = 0.033/0.042/0.067/0.013 ms
```

7.4 Building the storage cluster

Before deploying the WEKA storage cluster, verify that the storage network has been configured and that all storage NICs have proper connectivity.

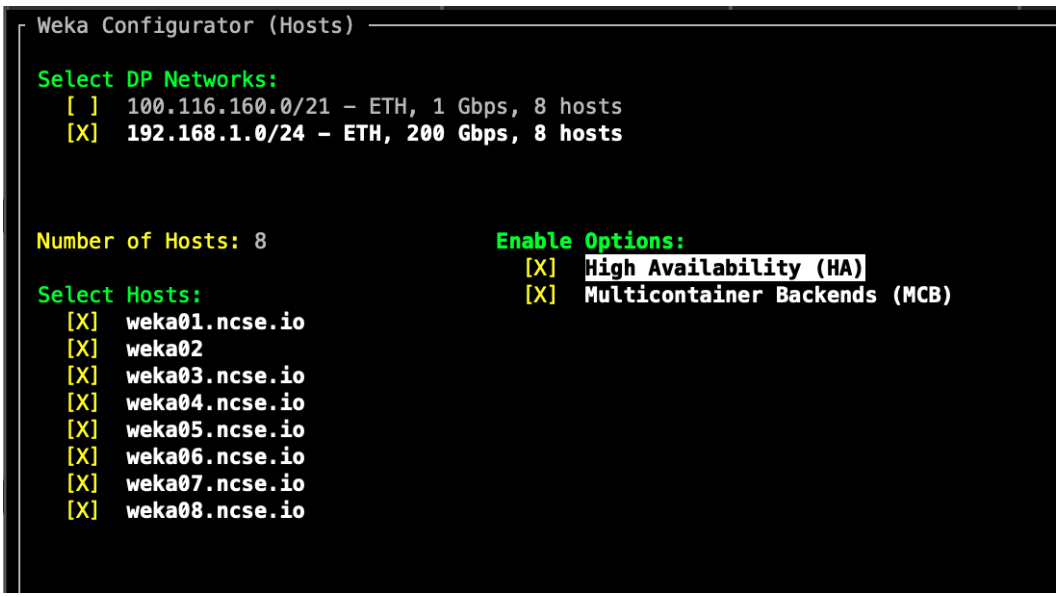
1. To initiate the cluster build process, execute the wekaconfig script found in the /opt/tools/install directory. This wrapper script creates the config.sh file, which is ultimately responsible for staging and deploying the storage cluster.
2. Execute the wekaconfig script and press Enter to continue to the GUI prompt.
3. After the GUI prompt appears, select the storage network interface.

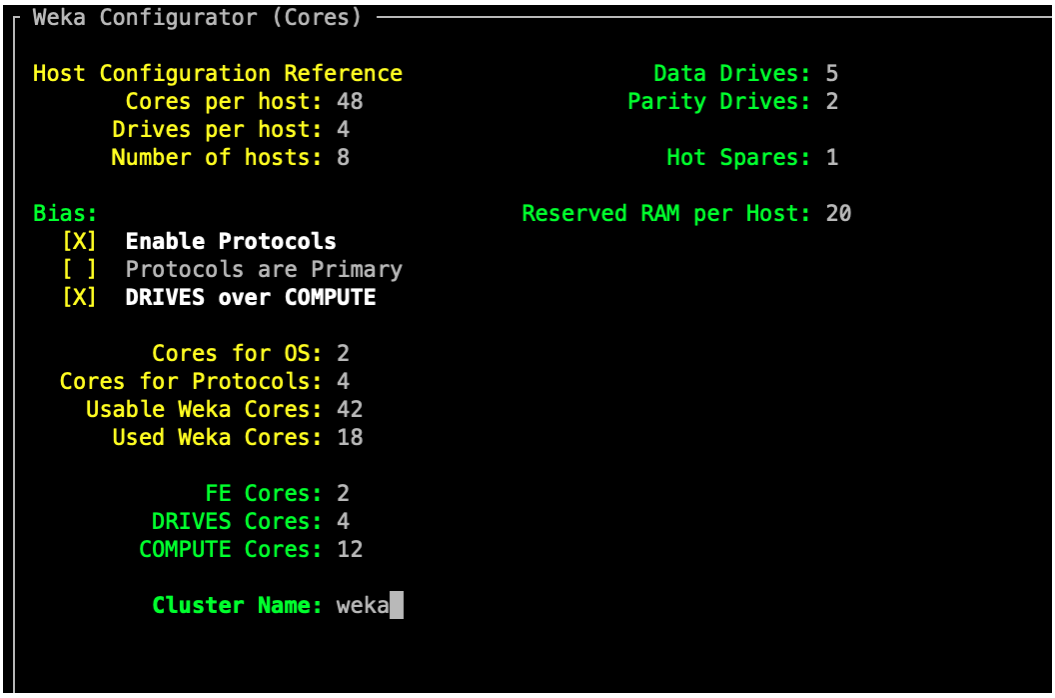
Note: This interface will not be shown if it is not yet configured.

```
[root@weka01 install]# ./wekaconfig
Setting TERMINFO to /tmp/_MEIj6JPdF/terminfo
collecting host data... please wait...
INFO:****************** Starting Weka Configurator ******************
INFO:looking for WEKA beacons on localhost
INFO:finding hosts...
INFO:Beacons found:

<snipped>

INFO:All 8 hosts are ping-able via dataplane
INFO:There appear to be 8 usable hosts - ['weka07.ncse.io', 'weka05.ncse.io',
'weka08.ncse.io', 'weka04.ncse.io', 'weka03.ncse.io', 'weka02', 'weka01.ncse.io',
'weka06.ncse.io']
INFO:Opening ssh to hosts
INFO:Probing for gateways
INFO:probing gateway for weka07.ncse.io/enp129s0f0
INFO: weka07.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka07.ncse.io/ens1f1np1
INFO: weka07.ncse.io/ens1f1np1 has gateway 192.168.1.1
INFO:probing gateway for weka05.ncse.io/enp129s0f0
INFO: weka05.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka05.ncse.io/ens1f1np1
INFO: weka05.ncse.io/ens1f1np1 has gateway 192.168.1.1
INFO:probing gateway for weka08.ncse.io/enp129s0f0
INFO: weka08.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka08.ncse.io/ens1f1np1
INFO: weka08.ncse.io/ens1f1np1 has gateway 192.168.1.1
INFO:probing gateway for weka04.ncse.io/enp129s0f0
INFO: weka04.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka04.ncse.io/ens1f1np1
INFO: weka04.ncse.io/ens1f1np1 has gateway 192.168.1.1
```

```
INFO:probing gateway for weka03.ncse.io/enp129s0f0
INFO: weka03.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka03.ncse.io/ens1f1np1
INFO: weka03.ncse.io/ens1f1np1 has gateway 192.168.1.1
INFO:probing gateway for weka02/enp129s0f0
INFO: weka02/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka02/ens1f1np1
INFO: weka02/ens1f1np1 has gateway 192.168.1.1
INFO:probing gateway for weka01.ncse.io/enp129s0f0
INFO: weka01.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka01.ncse.io/ens1f1np1
INFO: weka01.ncse.io/ens1f1np1 has gateway 192.168.1.1
INFO:probing gateway for weka06.ncse.io/enp129s0f0
INFO: weka06.ncse.io/enp129s0f0 has gateway 100.116.160.1
INFO:probing gateway for weka06.ncse.io/ens1f1np1
INFO: weka06.ncse.io/ens1f1np1 has gateway 192.168.1.1
INFO:************************* Analysis *************************
INFO:Host group is Homogeneous.
Scanning Complete. Press Enter to continue:
INFO:starting UI...
```

4. Select all eight servers and the storage network where these servers will synchronize with each other. This will create the config.sh file that will be used to stage the storage cluster for deployment.





5. Run the script config.sh until you see “Configuration process complete”.

Note: Ensure you have enabled passwordless SSH between all WEKA servers before running the script.

```
root@weka01 install]# ./config.sh
starting - PARA is TRUE
Stopping weka on weka01.ncse.io
cp ./resources_generator.py /tmp/
sudo weka local stop
No container names or types provided; applying action to all Weka containers on the
server.
Attempting to remove container default
Removing container default
error: Can't delete a still running container
Stopping weka on weka02
scp -p ./resources_generator.py weka02:/tmp/
ssh weka02 sudo weka local stop; sudo weka local rm -f default
Stopping weka on weka03.ncse.io
scp -p ./resources_generator.py weka03.ncse.io:/tmp/
ssh weka03.ncse.io sudo weka local stop; sudo weka local rm -f default

<snipped>

Container "frontend0" is ready (pid = 2634893)
frontend0: Allocated network device "0000:01:00.1" (with identifier "0000:01:00.1") to
slots [1,2] on "weka08.ncse.io":"frontend0" (1/1)
frontend0: Allocated core 3 to slot 2 on "weka08.ncse.io":"frontend0" (1/2)
frontend0: Allocated core 35 to slot 1 on "weka08.ncse.io":"frontend0" (2/2)
frontend0: Starting hugepages allocation for "weka08.ncse.io":"frontend0"
```

```
frontend0: Container "weka08.ncse.io":"frontend0" allocated 1408 out of 1408 required
hugepages after 1 retries
frontend0: Allocated 2816MB hugepages memory from 1 NUMA nodes for
"weka08.ncse.io":"frontend0"
frontend0: Bandwidth of "weka08.ncse.io":"frontend0" set to unlimited
frontend0: Disabled NUMA Balancing on dedicated host "weka08.ncse.io":"frontend0"
Container "frontend0" is ready (pid = 2616104)
Configuration process complete
```

6. At this stage, the WEKA cluster is not yet operational; it is only staged and configured for startup. Use the following command to verify that the WEKA storage cluster is currently not running.

```
[root@weka01 install]# weka status
WekaIO v5.0.1.101 (CLI build 4.4.8.53)

cluster: weka (5f6172d1-0ecf-4326-9d65-f02280b46b3b)
status: UNINITIALIZED (24 backend containers UP, 0/32 drives UP)
protection: 5+2 (N/A)
hot spare: 1 failure domain(s) (35.92 TiB)
drive storage: 251.46 TiB total, 287.38 TiB unavailable, 251.46 TiB unprovisioned
cloud: disabled
license: Unlicensed

io status: STOPPED (run 'weka cluster start-io' to start IO service)
link layer: Ethernet
clients: 0 connected
alerts: 3 active alerts, use `weka alerts` to list them
```

7. Start the WEKA storage cluster.

```
[root@weka01 install]# weka cluster start-io
```

8. Verify that the cluster is up and running.

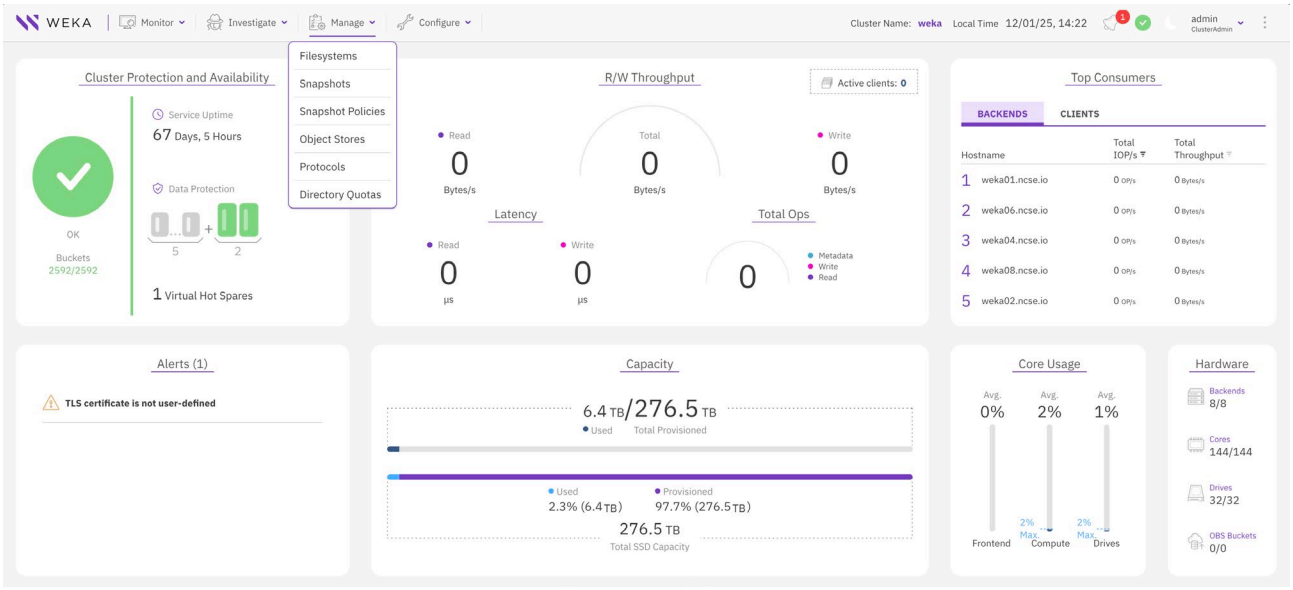
```
[root@weka01 install]# weka status
WekaIO v5.0.1.101 (CLI build 4.4.8.53)

cluster: weka (5f6172d1-0ecf-4326-9d65-f02280b46b3b)
status: OK (24 backend containers UP, 32 drives UP)
protection: 5+2 (Fully protected)
hot spare: 1 failure domain(s) (35.92 TiB)
drive storage: 251.46 TiB total, 251.46 TiB unprovisioned
cloud: disabled
license: Unlicensed

io status: STARTED 7 seconds ago (144 io-nodes UP, 2592 Buckets UP)
link layer: Ethernet
clients: 0 connected
reads: 0 B/s (0 IO/s)
writes: 0 B/s (0 IO/s)
```

```
operations: 0 ops/s
alerts: 3 active alerts, use `weka alerts` to list them
```

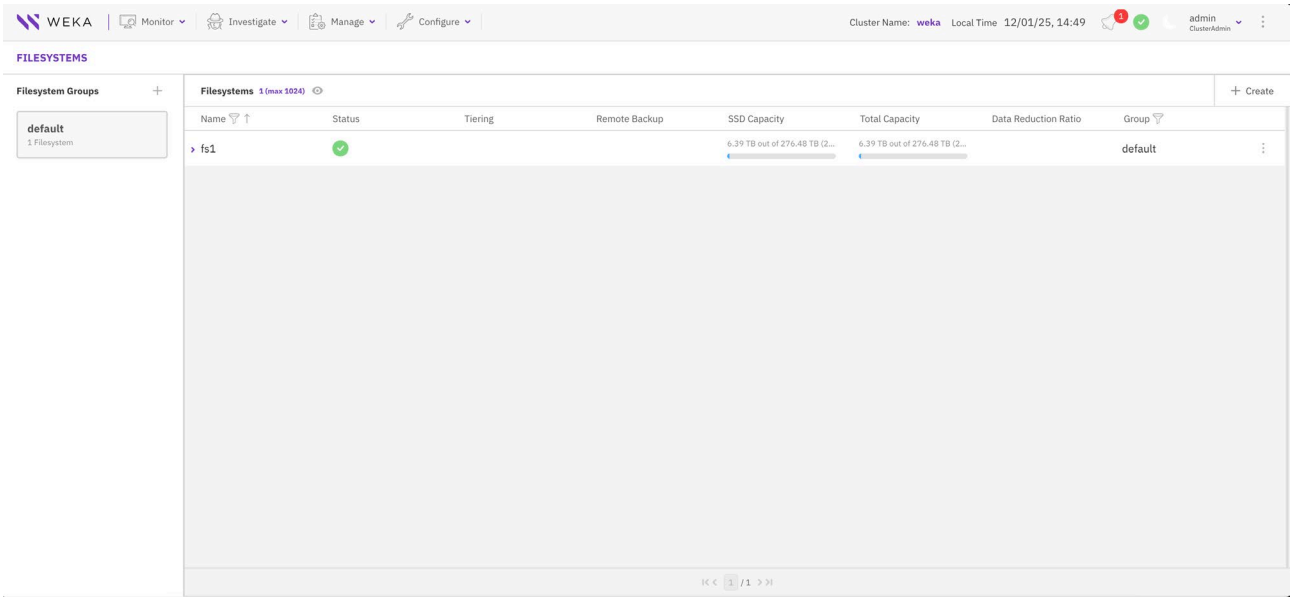
7.5 Creating the storage file system

To create a file system that spans all eight WEKA storage nodes, perform the following steps.

1. Open the WEKA GUI and, from the main dashboard, navigate to **Manage** → **Filesystems**.

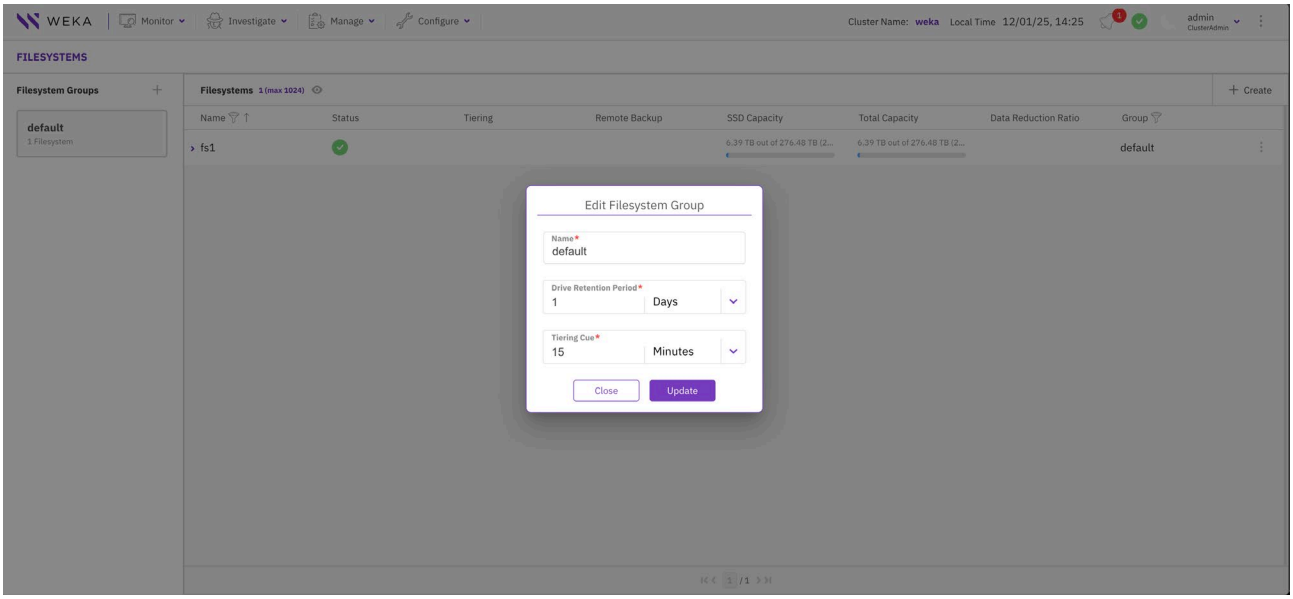


2. Create a **Filesystem Group**. Every file system must belong to a group. A group is a collection of file systems that share the same tiering rules.

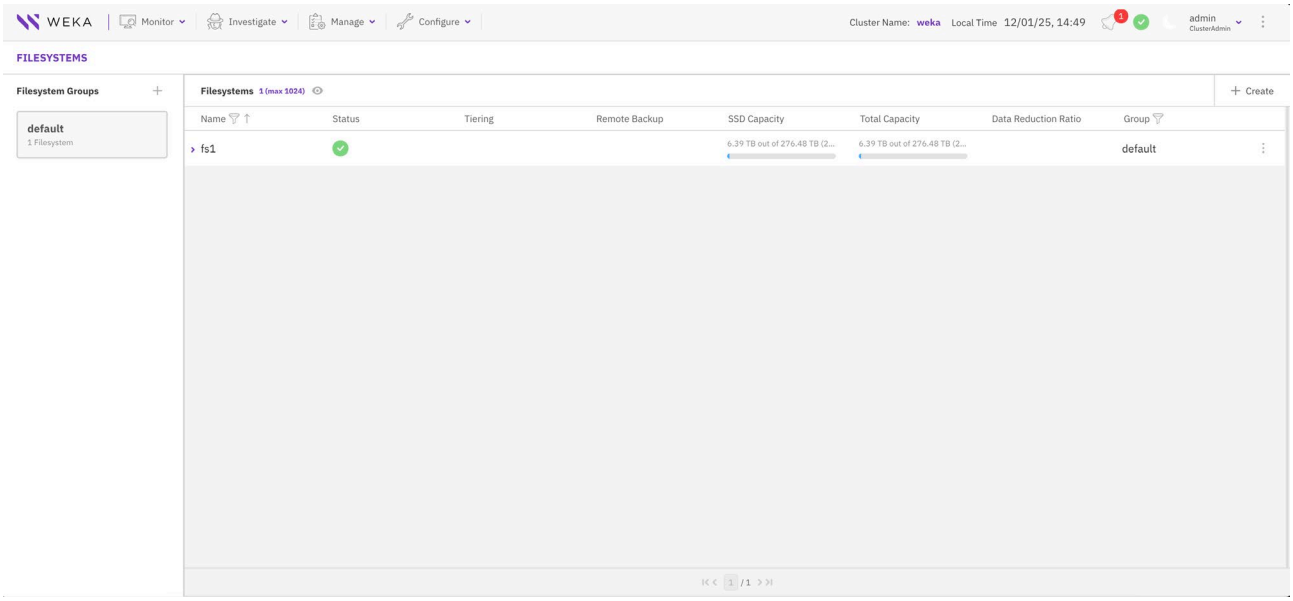


3. Name the Filesystem Group. You can leave the **Drive Retention Period** and **Tiering Cue** at their default values.

    - Drive Retention Period: defines how long WEKA keeps a copy of data on SSD after it has been tiered (that is, copied) to the object store
    - Tiering Cue: the minimum amount of time data must remain unchanged after being created or modified before WEKA moves it from SSD to the object store on a tiered file system

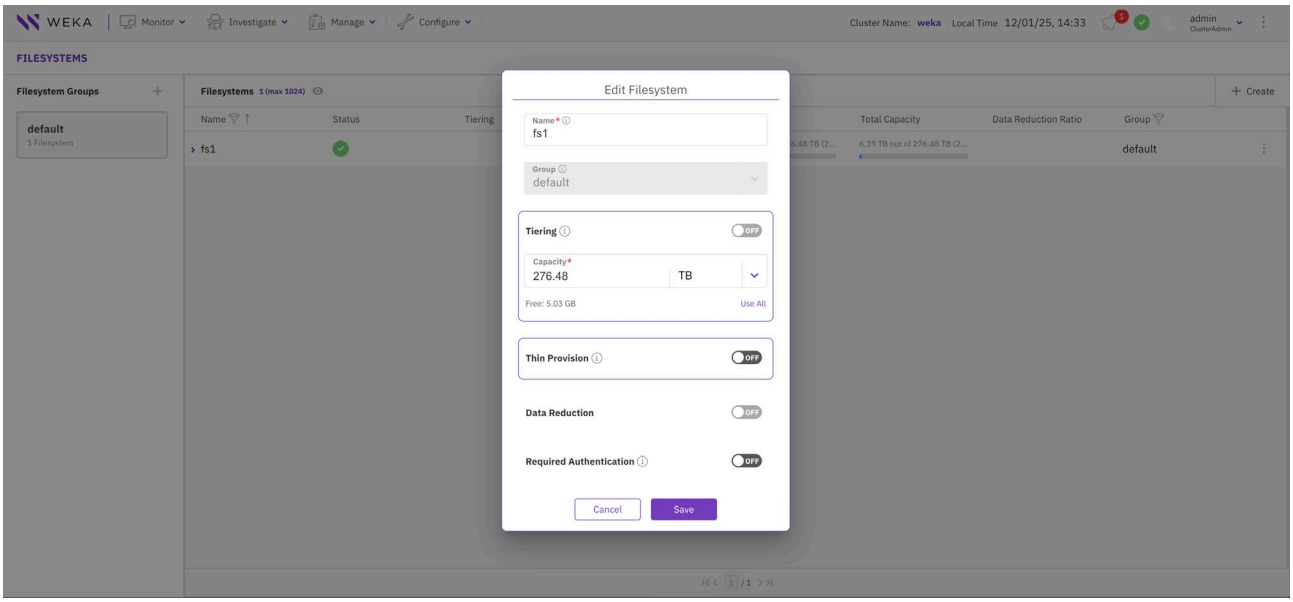


4. Click the Create button at the top right to create a file system.



5. Name the file system and assign it to the group you created earlier.

6. Specify the amount of storage for the file system — in this example, we use all available disk space. Leave **Thin Provisioning**, **Data Reduction**, and **Required Authentication** at their default (off) settings.



7.6 Mounting the WEKA system

The WEKA file system created in section 7.5 can be mounted on the GPU servers. Each node that mounts the WEKA file system is called a WEKA client. WEKA clients can be of two types: stateful and stateless:

- **stateless WEKA client**: A client that does not store any persistent metadata or configuration locally. All metadata is managed by the WEKA cluster. These clients are lightweight and can be easily added or replaced.
- **stateful WEKA client**: A client that stores some metadata or state locally, such as caching or configuration data. It may have slightly faster access to some operations but requires careful management if the client fails or is replaced.

For simplicity, we will use stateless clients on the GPU servers, as described here: https://docs.weka.io/planning-and-installation/bare-metal/adding-clients-bare-metal#add-a-stateless-client-to-the-cluster

1. Install the WEKA agent client on the GPU server (one-time setup). This step prepares the client to interact with the WEKA system. Provide the IP address of one of the WEKA backend servers.

```
curl http://192.168.1.189:14000/dist/v1/install | sh
```

2. Create a mount point. Create a directory on the client system where the WEKA filesystem will be mounted.

```
mkdir –p /mnt/weka
```

3. Mount the WEKA filesystem to the client using the **`mount`** command. In the following example, “fs1” is the filesystem name we created in section 7.5.

Note: UDP mode is required for mounting, as WEKA version 5.0.3-19 does not support DPDK mode for Thor/Thor2 BCM 57508 and 57608 NIC cards.

```
mount -t wekafs -o net=udp 192.168.1.189,192.168.1.190/fs1 /mnt/weka -o num_cores=8
```

4. After the WEKA file system is mounted, log in to verify cluster connectivity. Use the WEKA **`login`** command and provide the same username and password you use for the Dashboard GUI.

```
nvidia@slate5:~$ weka user login
Username: admin
Password:
+------------------------------+
| Login completed successfully |
+------------------------------+
Default profile updated successfully
nvidia@slate5:~$
nvidia@slate5:~$ weka status
WekaIO v5.0.3.19 (CLI build 5.0.3.19)

       cluster: weka (5f6172d1-0ecf-4326-9d65-f02280b46b3b)
        status: OK (24 backend containers UP, 32 drives UP)
    protection: 5+2 (Fully protected)
     hot spare: 1 failure domain(s) (35.92 TiB)
 drive storage: 251.46 TiB total, 4.68 GiB unprovisioned
         cloud: disabled
       license: OK, valid thru 2028-04-28T11:27:59Z

     io status: STARTED 70 days ago (144 io-nodes UP, 2592 Buckets UP)
    link layer: Ethernet
       clients: 4 connected
         reads: 0 B/s (0 IO/s)
        writes: 0 B/s (0 IO/s)
    operations: 0 ops/s
        alerts: 5 active alerts, use `weka alerts` to list them
```

8 Network feature configuration

8.1 Backend network

The backend network infrastructure, as described in section 5, provides communication paths between GPUs for distributed computing. This design includes IPv6 point-to-point interfaces between

the leafs of a stripe and the spines (using IPv6 unnumbered), along with /96 IPv6 addresses to the GPUs, which are aggregated into /94s and then advertised to the spine layer, and eventually, to other leafs. This section dives deeper into the network configuration for different aspects of this backend fabric.

8.1.1 Point-to-point interfaces between leafs and spines

The interfaces between the leafs of a stripe and the spines are enabled for IPv6 Router Advertisement (RA), which automatically generates an IPv6 link-local address for the interface.

```
A:admin@backend-stripe1-leaf1# show system lldp neighbor
  +---------------+-------------------+----------------------+---------------------+------------------------+---------------------+----------------+
  |     Name      |     Neighbor      | Neighbor System Name | Neighbor Chassis ID | Neighbor First Message | Neighbor Last Update | Neighbor Port  |
  +===============+===================+======================+=====================+========================+=====================+================+
  | ethernet-1/31 | 30:00:FC:2E:1C:81 | backend-spine1       | 30:00:FC:2E:1C:81   | a day ago              | 16 seconds ago       | ethernet-1/1   |
  | ethernet-1/32 | 30:00:FC:FF:16:B0 | backend-spine2       | 30:00:FC:FF:16:B0   | a day ago              | 20 seconds ago       | ethernet-1/1/1 |
  | mgmt0         | 00:21:05:A1:B8:24 | ebc210-sr12-mgmt     | 00:21:05:A1:B8:24   | a day ago              | 7 seconds ago        | esat-12/1/1    |
  +---------------+-------------------+----------------------+---------------------+------------------------+---------------------+----------------+
```

```
A:admin@backend-stripe1-leaf1# info interface ethernet-1/{31,32}
    interface ethernet-1/31 {
        admin-state enable
        ethernet {
            flow-control {
                receive false
            }
        }
        subinterface 0 {
            admin-state enable
            ipv6 {
                admin-state enable
                router-advertisement {
                    router-role {
                        admin-state enable
                        max-advertisement-interval 10
                        min-advertisement-interval 4
                    }
                }
            }
        }
    }
    interface ethernet-1/32 {
        admin-state enable
        ethernet {
            flow-control {
                receive false
            }
        }
        subinterface 0 {
            admin-state enable
            ipv6 {
                admin-state enable
                router-advertisement {
                    router-role {
```

```
                        admin-state enable
                        max-advertisement-interval 10
                        min-advertisement-interval 4
                    }
                }
            }
        }
    }
```

8.1.2 Point-to-point interfaces between leafs and GPUs

On the leafs, the interfaces to the GPUs are configured with /96 IPv6 addresses using a VLAN ID for 802.1Q tagging. These IPv6 addresses are encoded in a special way to include various parameters such as the leaf ID, the port number, the stripe number, and so on.

```
A:admin@backend-stripe1-leaf1# info interface ethernet-1/3
    admin-state enable
    vlan-tagging true
    ethernet {
        port-speed 400G
        flow-control {
            receive false
        }
    }
    subinterface 1000 {
        type routed
        description "Backend: backend-fabric Stripe: stripe1"
        admin-state enable
        ip-mtu 9000
        ipv6 {
            admin-state enable
            address fd00:1:1:1:0:3:0:1/96 {
                primary
            }
            neighbor-discovery {
                learn-unsolicited both
                host-route {
                    populate dynamic {
                        datapath-programming true
                    }
                }
            }
            router-advertisement {
                router-role {
                    admin-state enable
                    current-hop-limit 64
                    managed-configuration-flag false
                    other-configuration-flag false
                    max-advertisement-interval 600
                    min-advertisement-interval 200
                    reachable-time 0
                    retransmit-time 0
```

```
                    router-lifetime 1800
                }
            }
        }
        vlan {
            encap {
                single-tagged {
                    vlan-id 1000
                }
            }
        }
    }
```

8.1.3 BGP in the fabric for route distribution

The backend network is configured as an IP fabric using BGP for route distribution between the leafs and spines of the fabric. The BGP peering is established using BGP dynamic neighbors, leveraging the underlying IPv6 link-local addressing per-interface.

```
A:admin@backend-stripe1-leaf1# info network-instance default protocols bgp
    admin-state enable
    autonomous-system 4200000014
    router-id 192.0.2.10
    dynamic-neighbors {
        interface ethernet-1/31.0 {
            peer-group bgpgroup-ebgp-stripe-connector
            allowed-peer-as [
                4200000016
            ]
        }
        interface ethernet-1/32.0 {
            peer-group bgpgroup-ebgp-stripe-connector
            allowed-peer-as [
                4200000016
            ]
        }
    }
    ebgp-default-policy {
        import-reject-all true
        export-reject-all true
    }
    afi-safi evpn {
        admin-state disable
    }
    afi-safi ipv4-unicast {
        admin-state enable
        multipath {
            allow-multiple-as true
            ebgp {
                maximum-paths 32
            }
            ibgp {
```

```
                maximum-paths 32
            }
        }
        ipv4-unicast {
            advertise-ipv6-next-hops true
            receive-ipv6-next-hops true
        }
    }
    afi-safi ipv6-unicast {
        admin-state enable
        multipath {
            allow-multiple-as true
            ebgp {
                maximum-paths 2
            }
            ibgp {
                maximum-paths 32
            }
        }
    }
    preference {
        ebgp 170
        ibgp 170
    }
    route-advertisement {
        rapid-withdrawal true
        wait-for-fib-install false
    }
    group bgpgroup-ebgp-stripe-connector {
        admin-state enable
        export-policy [
            ebgp-isl-export-stripeconnector-backend-backend-fabric
        ]
        import-policy [
            ebgp-isl-import-policy-stripe-connector
        ]
        afi-safi evpn {
            admin-state disable
        }
        afi-safi ipv4-unicast {
            admin-state enable
            ipv4-unicast {
                advertise-ipv6-next-hops true
                receive-ipv6-next-hops true
            }
        }
        afi-safi ipv6-unicast {
            admin-state enable
        }
    }
```

8.1.4 Dynamic load balancing

For AI/HPC clusters, static ECMP is not optimal for load-balancing traffic across available paths. In the backend fabric for AI clusters, traffic patterns include long-lived, elephant flows that are bursty in nature—static load-balancing pins flows to a specific hash index (and next-hop), not considering any feedback from the fabric itself (congestion).

Dynamic load balancing (DLB) is a next-generation adaptive load-balancing technique that can react to link utilization and dynamically shift flows into lower utilized paths. This technique is better-suited for AI clusters than static ECMP.

```
A:admin@backend-stripe1-leaf1# info system load-balancing
    dynamic {
        flowset-size 256
        inactivity-timer 50
        mode flow-dynamic
        link-quality-sampling-interval 5
        weighting-factor {
            port-utilization 70
            queue-utilization 20
            itm-utilization 10
        }
    }
A:admin@backend-stripe1-leaf1# info network-instance default ip-load-balancing
    dynamic-load-balancing {
        prefix ::/0 {
        }
    }
A:admin@backend-stripe1-leaf1# info network-instance nvidia-gpus ip-load-balancing
    dynamic-load-balancing {
        prefix ::/0 {
        }
    }
```

8.1.5 Route aggregation for GPU subnets

When advertising the GPU subnets to the spines, each leaf aggregates its locally configured /96 IPv6 subnets into a broader /94 subnet, encompassing all possible /96 allocations it might have for its connected GPUs. For example, stripe1-leaf1 has two interfaces to GPU0s of two NVIDIA H200 servers, with fd00:1:1:1:0:1:0:1/96 and fd00:1:1:1:0:2:0:1/96, respectively. These are aggregated into a fd00:1:1:1::/94 IPv6 subnet, with a BGP policy allowing only the /94 IPv6 subnet to be advertised, suppressing the individual /96 IPv6 addresses.

```
A:admin@backend-stripe1-leaf1# info network-instance default aggregate-routes
    route fd00:1:1:1::/94 {
        admin-state enable
        summary-only true
        aggregator {
            address 192.0.2.10
        }
```

```
    }

A:admin@backend-stripe1-leaf1# info network-instance default protocols bgp group bgpgroup-
ebgp-stripe-connector
    admin-state enable
    export-policy [
        ebgp-isl-export-stripeconnector-backend-backend-fabric
    ]
    import-policy [
        ebgp-isl-import-policy-stripe-connector
    ]
    afi-safi evpn {
        admin-state disable
    }
    afi-safi ipv4-unicast {
        admin-state enable
        ipv4-unicast {
            advertise-ipv6-next-hops true
            receive-ipv6-next-hops true
        }
    }
    afi-safi ipv6-unicast {
        admin-state enable
    }

A:admin@backend-stripe1-leaf1# info routing-policy policy ebgp-isl-export-stripeconnector-
backend-backend-fabric
    default-action {
        policy-result reject
    }
    statement 10 {
        match {
            protocol local
            prefix {
                prefix-set prefixset-backend-backend-fabric-stripeconnector-stripe-
connector
            }
        }
        action {
            policy-result accept
            bgp {
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 15 {
        match {
            protocol bgp
        }
        action {
            policy-result accept
            bgp {
```

```
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 20 {
        match {
            protocol aggregate
        }
        action {
            policy-result accept
            bgp {
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 25 {
        match {
            bgp {
                evpn {
                    route-type [
                        1
                    ]
                }
            }
        }
        action {
            policy-result accept
            bgp {
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 30 {
        match {
            bgp {
                evpn {
                    route-type [
                        2
                    ]
                }
            }
        }
        action {
            policy-result accept
            bgp {
                local-preference {
                    set 100
                }
```

```
            }
        }
    }
    statement 35 {
        match {
            bgp {
                evpn {
                    route-type [
                        3
                    ]
                }
            }
        }
        action {
            policy-result accept
            bgp {
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 40 {
        match {
            bgp {
                evpn {
                    route-type [
                        4
                    ]
                }
            }
        }
        action {
            policy-result accept
            bgp {
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 45 {
        match {
            bgp {
                evpn {
                    route-type [
                        5
                    ]
                }
            }
        }
        action {
            policy-result accept
            bgp {
```

```
                local-preference {
                    set 100
                }
            }
        }
    }
    statement 60 {
        match {
            prefix {
                prefix-set prefixset-routeleak-import-nvidia-gpus-backend-fabric
            }
        }
        action {
            policy-result accept
        }
    }

A:admin@backend-stripe1-leaf1# info routing-policy prefix-set prefixset-routeleak-import-
nvidia-gpus-backend-fabric
    prefix fd00:1:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:2:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:3:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:4:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:5:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:6:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:7:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:8:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:9:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:10:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:11:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:12:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:13:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:14:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:15:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:16:1:1::/94 mask-length-range 94..94 {
    }
```

8.1.6 IP VRFs for tenant isolation in the fabric

For single-tenant AI fabrics, all GPUs are mapped to the same IP VRF across all stripes. Because the leaf-spine BGP peering exists in the default routing instance, the GPU subnets must be leaked into the default instance. In addition, any remote GPU subnet received via BGP in the default routing instance must also be leaked into the IP VRF. This route leaking is achieved via inter-instance routing policies.

For example, an export inter-instance policy, within the IP VRF, defines what routes are allowed to be leaked out of the IP VRF while an import inter-instance policy in the default network instance can match these leaked routes and import them.

```
A:admin@backend-stripe1-leaf1# info network-instance nvidia-gpus
    type ip-vrf
    admin-state enable
    description "Backend: backend-fabric"
    router-id 192.0.2.15
    ip-load-balancing {
        dynamic-load-balancing {
            prefix ::/0 {
            }
        }
    }
    interface ethernet-1/3.1000 {
    }
    interface ethernet-1/4.1000 {
    }
    inter-instance-policies {
        apply-policy {
            import-policy import-routeleak-nvidia-gpus-backend-fabric
            export-policy export-routeleak-nvidia-gpus-backend-fabric
        }
    }

A:admin@backend-stripe1-leaf1# info routing-policy policy export-routeleak-nvidia-gpus-
backend-fabric
    default-action {
        policy-result reject
    }
    statement 20 {
        match {
            protocol arp-nd
        }
        action {
            policy-result accept
        }
    }
    statement 10 {
        match {
            protocol local
            prefix {
                prefix-set prefixset-routeleak-export-nvidia-gpus-backend-fabric
            }
        }
```

```
        action {
            policy-result accept
        }
    }

A:admin@backend-stripe1-leaf1# info routing-policy policy import-routeleak-nvidia-gpus-
backend-fabric
    default-action {
        policy-result reject
    }
    statement 10 {
        match {
            protocol bgp
            prefix {
                prefix-set prefixset-routeleak-import-nvidia-gpus-backend-fabric
            }
        }
        action {
            policy-result accept
        }
    }

A:admin@backend-stripe1-leaf1# info routing-policy prefix-set prefixset-routeleak-export-
nvidia-gpus-backend-fabric
    prefix fd00:1:1:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:1:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:2:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:2:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:3:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:3:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:4:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:4:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:5:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:5:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:6:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:6:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:7:1:0:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:7:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:1:8:1:0:1::/96 mask-length-range 96..96 {
    }
```

```
    prefix fd00:1:8:1:0:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:9:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:9:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:10:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:10:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:11:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:11:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:12:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:12:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:13:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:13:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:14:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:14:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:15:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:15:1:1:2::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:16:1:1:1::/96 mask-length-range 96..96 {
    }
    prefix fd00:2:16:1:1:2::/96 mask-length-range 96..96 {
    }

A:admin@backend-stripe1-leaf1# info routing-policy prefix-set prefixset-routeleak-import-
nvidia-gpus-backend-fabric
    prefix fd00:1:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:2:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:3:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:4:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:5:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:6:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:7:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:1:8:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:9:1:1::/94 mask-length-range 94..94 {
```

```
    }
    prefix fd00:2:10:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:11:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:12:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:13:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:14:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:15:1:1::/94 mask-length-range 94..94 {
    }
    prefix fd00:2:16:1:1::/94 mask-length-range 94..94 {
    }

A:admin@backend-stripe1-leaf1# info network-instance default inter-instance-policies
    apply-policy {
        import-policy default-import-routeleak-backend-fabric
        export-policy default-export-routeleak-backend-fabric
    }

A:admin@backend-stripe1-leaf1# info routing-policy policy default-*-routeleak-backend-
fabric
    policy default-export-routeleak-backend-fabric {
        default-action {
            policy-result accept
        }
    }
    policy default-import-routeleak-backend-fabric {
        default-action {
            policy-result accept
        }
    }
```

8.1.7 Quality of service for RoCEv2 traffic

Quality of service (QoS) is crucial in building a lossless fabric for AI clusters, using ECN and PFC to slow down traffic instead of dropping packets during congestion. In SR Linux, this configuration includes:

- Mapping queues to queue indices and mapping a forwarding class to the queues.
- Creating a PFC mapping profile that enables PFC for a specific PFC priority (priority 3, in our case) and configuring the appropriate PFC deadlock detection and recovery timers.
- Creating a classification policy that maps packets with a specified DSCP value into a specific forwarding class. In our design, this includes mapping RoCEv2 traffic into forwarding class of FC3 and CNP into a forwarding class of FC6, both with a low drop probability.

- Creating a scheduler policy for the CNP and RoCEv2 queues—the CNP queue is marked as a strict priority queue, while the RoCEv2 queue uses weighted round-robin.
- Creating a queue management profile that enables ECN and defines the minimum and maximum thresholds for the same.
- Creating a buffer allocation profile that assigns the maximum burst size across different queues and PFC priorities.
- Defining the PFC headroom buffer allocation.

```
A:admin@backend-stripe1-leaf1# info qos
    explicit-congestion-notification {
    }
    queues {
        queue unicast-0 {
            queue-index 0
        }
        queue unicast-3 {
            queue-index 3
        }
        queue unicast-6 {
            queue-index 6
        }
        pfc-queue pfc-3 {
            queue-index 3
        }
    }
    forwarding-classes {
        forwarding-class fc0 {
            output {
                unicast-queue unicast-0
            }
        }
        forwarding-class fc3 {
            output {
                unicast-queue unicast-3
            }
        }
        forwarding-class fc6 {
            output {
                unicast-queue unicast-6
            }
        }
    }
    pfc-mapping-profile fabric-edge-interfaces {
        pfc-priority 3 {
            pfc-enable true
        }
        received-traffic {
            unicast-mapping {
                pfc-queue pfc-3 {
                    forwarding-class fc3
```

```
                    pfc-pause-frame-priority 3
                }
            }
        }
        received-pfc-pause-frames {
            deadlock {
                enable true
                detection-timer 750
                recovery-timer 750
            }
            queue unicast-3 {
                pfc-pause-frame-priority 3
            }
        }
    }
    pfc-mapping-profile fabric-stripe-connector {
        pfc-priority 3 {
            pfc-enable true
        }
        received-traffic {
            unicast-mapping {
                pfc-queue pfc-3 {
                    forwarding-class fc3
                    pfc-pause-frame-priority 3
                }
            }
        }
        received-pfc-pause-frames {
            deadlock {
                enable true
                detection-timer 750
                recovery-timer 750
            }
            queue unicast-3 {
                pfc-pause-frame-priority 3
            }
        }
    }
    classifiers {
        dscp-policy ingress-backend-backend-fabric {
            dscp 0 {
                forwarding-class fc0
                drop-probability low
            }
            dscp 26 {
                forwarding-class fc3
                drop-probability low
            }
            dscp 48 {
                forwarding-class fc6
                drop-probability low
            }
        }
    }
```

```
    scheduler-policies {
        scheduler-policy egress-backend-backend-fabric {
            scheduler 0 {
                priority strict
                input unicast-6 {
                    queue-name unicast-6
                    peak-rate-percent 80
                }
            }
            scheduler 1 {
                input unicast-0 {
                    queue-name unicast-0
                    peak-rate-percent 10
                    weight 10
                }
                input unicast-3 {
                    queue-name unicast-3
                    peak-rate-percent 100
                    weight 50
                }
            }
        }
    }
    interfaces {
        interface ethernet-1/1 {
            interface-ref {
                interface ethernet-1/1
            }
            pfc {
                pfc-mapping-profile fabric-edge-interfaces
                pfc-enable true
            }
            input {
                pfc-buffer-allocation-profile ingress-backend-backend-fabric
            }
            output {
                buffer-allocation-profile egress-backend-backend-fabric
                queues {
                    queue unicast-3 {
                        queue-management-profile egress-backend-backend-fabric-2
                    }
                }
                scheduler {
                    scheduler-policy egress-backend-backend-fabric
                }
            }
        }
        interface ethernet-1/2 {
            interface-ref {
                interface ethernet-1/2
            }
            pfc {
                pfc-mapping-profile fabric-edge-interfaces
                pfc-enable true
```

```
            }
            input {
                pfc-buffer-allocation-profile ingress-backend-backend-fabric
            }
            output {
                buffer-allocation-profile egress-backend-backend-fabric
                queues {
                    queue unicast-3 {
                        queue-management-profile egress-backend-backend-fabric-2
                    }
                }
                scheduler {
                    scheduler-policy egress-backend-backend-fabric
                }
            }
        }
        interface ethernet-1/31 {
            interface-ref {
                interface ethernet-1/31
            }
            pfc {
                pfc-mapping-profile fabric-stripe-connector
                pfc-enable true
            }
            input {
                pfc-buffer-allocation-profile ingress-backend-backend-fabric
            }
            output {
                buffer-allocation-profile egress-backend-backend-fabric
                queues {
                    queue unicast-3 {
                        queue-management-profile egress-backend-backend-fabric-2
                    }
                }
                scheduler {
                    scheduler-policy egress-backend-backend-fabric
                }
            }
        }
        interface ethernet-1/31.0 {
            interface-ref {
                interface ethernet-1/31
                subinterface 0
            }
            input {
                classifiers {
                    dscp-policy ingress-backend-backend-fabric
                }
            }
        }
        interface ethernet-1/32 {
            interface-ref {
                interface ethernet-1/32
            }
```

```
            pfc {
                pfc-mapping-profile fabric-stripe-connector
                pfc-enable true
            }
            input {
                pfc-buffer-allocation-profile ingress-backend-backend-fabric
            }
            output {
                buffer-allocation-profile egress-backend-backend-fabric
                queues {
                    queue unicast-3 {
                        queue-management-profile egress-backend-backend-fabric-2
                    }
                }
                scheduler {
                    scheduler-policy egress-backend-backend-fabric
                }
            }
        }
        interface ethernet-1/32.0 {
            interface-ref {
                interface ethernet-1/32
                subinterface 0
            }
            input {
                classifiers {
                    dscp-policy ingress-backend-backend-fabric
                }
            }
        }
    }
    buffer-management {
        queue-management-profile egress-backend-backend-fabric-2 {
            weight-factor 0
            wred {
                wred-slope all drop-probability all enable-ecn true {
                    min-threshold-percent 30
                    max-threshold-percent 85
                    slope-enabled true
                    max-drop-probability-percent 100
                }
            }
        }
        buffer-allocation-profile egress-backend-backend-fabric {
            queues {
                queue unicast-0 {
                    maximum-burst-size 5211064
                }
                queue unicast-3 {
                    maximum-burst-size 52110640
                }
                queue unicast-6 {
                    maximum-burst-size 52110640
                }
```

```
            }
        }
        buffer-allocation-profile ingress-backend-backend-fabric {
            queues {
                pfc-queue pfc-3 {
                    maximum-burst-size 52110640
                }
            }
        }
    }
    linecard 1 {
        forwarding-complex 0 {
            input {
                pfc-buffer-reservation 10
            }
        }
    }
```

8.2 Converged frontend and storage network

In this design, the frontend and storage network is converged into a common two-leaf, two-spine fabric, catering to both services. This is a 3-stage IPv6 unnumbered EVPN VXLAN fabric that enables multihoming to the storage cluster, the GPU storage NICs, and the frontend server using Ethernet segments and MAC VRFs.

8.2.1 Point-to-point interfaces between leafs and spines for underlay

As for the backend fabric, this converged frontend and storage fabric also includes IPv6 unnumbered interfaces between the leafs and the spines.

```
A:admin@frontend-leaf1# info interface ethernet-1/{31,32}
    interface ethernet-1/31 {
        description "to frontend-spine1 eth-1/1/1"
        admin-state enable
        subinterface 0 {
            admin-state enable
            ipv6 {
                admin-state enable
                router-advertisement {
                    router-role {
                        admin-state enable
                        max-advertisement-interval 10
                        min-advertisement-interval 4
                    }
                }
            }
        }
    }
    interface ethernet-1/32 {
        description "to frontend-spine2 eth-1/1/1"
        admin-state enable
```

```
        subinterface 0 {
            admin-state enable
            ipv6 {
                admin-state enable
                router-advertisement {
                    router-role {
                        admin-state enable
                        max-advertisement-interval 10
                        min-advertisement-interval 4
                    }
                }
            }
        }
    }
```

8.2.2 Default network instance

The leaf-spine interfaces are added in the default network instance for BGP peering using BGP dynamic neighbors.

```
A:admin@frontend-leaf1# info network-instance default
    type default
    admin-state enable
    description "fabric: frontend-fabric role: leaf"
    router-id 192.0.2.2
    ip-forwarding {
        receive-ipv4-check false
    }
    interface ethernet-1/31.0 {
    }
    interface ethernet-1/32.0 {
    }
    interface system0.0 {
    }

*snip*
```

8.2.3 BGP configuration for underlay and overlay

BGP is configured for dynamic neighbors, leveraging the underlying IPv6 unnumbered interfaces (with IPv6 link-local addresses).

```
A:admin@frontend-leaf1# info network-instance default protocols bgp
    admin-state enable
    autonomous-system 101
    router-id 192.0.2.2
    dynamic-neighbors {
        interface ethernet-1/31.0 {
            peer-group bgpgroup-ebgp-frontend-fabric
            allowed-peer-as [
                100
```

```
            ]
        }
        interface ethernet-1/32.0 {
            peer-group bgpgroup-ebgp-frontend-fabric
            allowed-peer-as [
                100
            ]
        }
    }
    ebgp-default-policy {
        import-reject-all true
        export-reject-all true
    }
    afi-safi evpn {
        admin-state enable
        multipath {
            allow-multiple-as true
            ebgp {
                maximum-paths 64
            }
            ibgp {
                maximum-paths 64
            }
        }
        evpn {
            inter-as-vpn true
            rapid-update true
        }
    }
    afi-safi ipv4-unicast {
        admin-state enable
        multipath {
            allow-multiple-as true
            ebgp {
                maximum-paths 2
            }
            ibgp {
                maximum-paths 2
            }
        }
        ipv4-unicast {
            advertise-ipv6-next-hops true
            receive-ipv6-next-hops true
        }
        evpn {
            rapid-update true
        }
    }
    afi-safi ipv6-unicast {
        admin-state enable
        multipath {
            allow-multiple-as true
            ebgp {
                maximum-paths 2
```

```
            }
            ibgp {
                maximum-paths 2
            }
        }
        evpn {
            rapid-update true
        }
    }
    preference {
        ebgp 170
        ibgp 170
    }
    route-advertisement {
        rapid-withdrawal true
        wait-for-fib-install false
    }
    group bgpgroup-ebgp-frontend-fabric {
        admin-state enable
        export-policy [
            ebgp-isl-export-policy-frontend-fabric
        ]
        import-policy [
            ebgp-isl-import-policy-frontend-fabric
        ]
        failure-detection {
            enable-bfd true
            fast-failover true
        }
        afi-safi evpn {
            admin-state enable
        }
        afi-safi ipv4-unicast {
            admin-state enable
            ipv4-unicast {
                advertise-ipv6-next-hops true
                receive-ipv6-next-hops true
            }
        }
        afi-safi ipv6-unicast {
            admin-state enable
        }
    }
```

8.2.4 Ethernet segments to GPU servers

Links to the GPU storage NICs are configured to be a part of a Link Aggregation Group (LAG) mapped to an Ethernet segment for EVPN-based active/active multihoming (shown from the perspective of one leaf only).

```
A:admin@frontend-leaf1# info interface ethernet-1/1/1
    description lag-gpu-server1
```

```
    admin-state enable
    ethernet {
        aggregate-id lag1
        lacp-port-priority 32768
    }

A:admin@frontend-leaf1# info interface lag1
    description lag-gpu-server1
    admin-state enable
    vlan-tagging true
    subinterface 100 {
        type bridged
        description gpu
        admin-state enable
        vlan {
            encap {
                single-tagged {
                    vlan-id 100
                }
            }
        }
    }
    subinterface 200 {
        type bridged
        description gpu-frontend
        admin-state enable
        vlan {
            encap {
                single-tagged {
                    vlan-id 200
                }
            }
        }
    }
    lag {
        lag-type lacp
        min-links 1
        lacp-fallback-mode static
        lacp-fallback-timeout 60
        lacp {
            interval FAST
            lacp-mode ACTIVE
            admin-key 3
            system-id-mac 00:00:00:00:00:11
            system-priority 32768
        }
    }

A:admin@frontend-leaf1# info system network-instance protocols evpn ethernet-segments bgp-
instance 1 ethernet-segment lag-gpu-server1
    admin-state enable
    esi 00:00:00:00:00:00:11:00:00:00
    multi-homing-mode all-active
    interface lag1 {
```

```
    }
    df-election {
        timers {
            activation-timer 0
        }
        algorithm {
            type default
        }
    }
```

8.2.5 Ethernet segments to WEKA storage nodes

Links to the WEKA storage NICs are configured to be a part of a LAG mapped to an Ethernet segment for EVPN-based active/active multihoming (shown from the perspective of one leaf only). Every WEKA node in our eight-node cluster connects to each leaf.

```
A:admin@frontend-leaf1# info interface ethernet-1/5/1
    description lag-storage-server1
    admin-state enable
    ethernet {
        aggregate-id lag5
        lacp-port-priority 32768
    }

A:admin@frontend-leaf1# info interface lag5
    description lag-storage-server1
    admin-state enable
    vlan-tagging true
    subinterface 4096 {
        type bridged
        description storage
        admin-state enable
        vlan {
            encap {
                untagged {
                }
            }
        }
    }
    lag {
        lag-type lacp
        min-links 1
        lacp-fallback-mode static
        lacp-fallback-timeout 60
        lacp {
            interval FAST
            lacp-mode ACTIVE
            admin-key 6
            system-id-mac 00:00:00:00:00:15
            system-priority 32768
        }
    }
```

```
A:admin@frontend-leaf1# info system network-instance protocols evpn ethernet-segments bgp-
instance 1 ethernet-segment lag-storage-server1
    admin-state enable
    esi 00:00:00:00:00:00:15:00:00:00
    multi-homing-mode all-active
    interface lag5 {
    }
    df-election {
        timers {
            activation-timer 0
        }
        algorithm {
            type default
        }
    }
```

8.2.6 Ethernet segments to frontend server

Links to the frontend server are configured to be a part of a LAG mapped to an Ethernet segment for EVPN-based active/active multihoming (shown from the perspective of one leaf only).

```
A:admin@frontend-leaf1# info interface ethernet-1/29
    description lag-frontend-server
    admin-state enable
    ethernet {
        aggregate-id lag13
        lacp-port-priority 32768
    }

A:admin@frontend-leaf1# info interface lag13
    description lag-frontend-server
    admin-state enable
    vlan-tagging true
    subinterface 200 {
        type bridged
        description frontend-server
        admin-state enable
        vlan {
            encap {
                single-tagged {
                    vlan-id 200
                }
            }
        }
    }
    lag {
        lag-type lacp
        min-links 1
        lacp-fallback-mode static
        lacp-fallback-timeout 60
        lacp {
            interval FAST
```

```
            lacp-mode ACTIVE
            admin-key 1
            system-id-mac 00:00:00:00:00:23
            system-priority 32768
        }
    }

A:admin@frontend-leaf1# info system network-instance protocols evpn ethernet-segments bgp-
instance 1 ethernet-segment lag-frontend-server
    admin-state enable
    esi 00:00:00:00:00:00:23:00:00:00
    multi-homing-mode all-active
    interface lag13 {
    }
    df-election {
        timers {
            activation-timer 0
        }
        algorithm {
            type default
        }
    }
```

8.2.7 MAC VRFs for GPU storage NICs, storage cluster, and frontend server

MAC VRFs are configured for GPU storage NICs, the storage servers, and frontend server connectivity using Layer 2 subinterfaces with specific VLANs. One MAC VRF connects the storage nodes and the GPU frontend/storage NICs while another MAC VRF connects the GPU frontend/storage NICs and the frontend server.

```
A:admin@frontend-leaf1# info network-instance mac-vrf-frontend
    type mac-vrf
    admin-state enable
    description mac-vrf-frontend
    interface lag1.200 {
    }
    interface lag13.200 {
    }
    interface lag2.200 {
    }
    interface lag3.200 {
    }
    interface lag4.200 {
    }
    vxlan-interface vxlan0.501 {
    }
    protocols {
        bgp-evpn {
            bgp-instance 1 {
                vxlan-interface vxlan0.501
                evi 200
```

```
                ecmp 8
                routes {
                    bridge-table {
                        mac-ip {
                            advertise true
                            advertise-arp-nd-only-with-mac-table-entry true
                        }
                        inclusive-mcast {
                            advertise true
                        }
                    }
                }
            }
        }
        bgp-vpn {
            bgp-instance 1 {
                route-target {
                    export-rt target:1:200
                    import-rt target:1:200
                }
            }
        }
    }
    bridge-table {
        mac-learning {
            admin-state enable
            aging {
                admin-state enable
                age-time 300
            }
        }
        proxy-arp {
            admin-state enable
            table-size 250
            dynamic-learning {
                admin-state enable
                age-time 2000
                send-refresh 2000
            }
            ip-duplication {
                monitoring-window 10
                num-moves 4
                hold-down-time 10
            }
        }
    }

A:admin@frontend-leaf1# info network-instance mac-vrf-storage
    type mac-vrf
    admin-state enable
    description mac-vrf-storage
    interface lag1.100 {
    }
    interface lag10.4096 {
```

```
    }
    interface lag11.4096 {
    }
    interface lag12.4096 {
    }
    interface lag2.100 {
    }
    interface lag3.100 {
    }
    interface lag4.100 {
    }
    interface lag5.4096 {
    }
    interface lag6.4096 {
    }
    interface lag7.4096 {
    }
    interface lag8.4096 {
    }
    interface lag9.4096 {
    }
    vxlan-interface vxlan0.500 {
    }
    protocols {
        bgp-evpn {
            bgp-instance 1 {
                vxlan-interface vxlan0.500
                evi 100
                ecmp 8
                routes {
                    bridge-table {
                        mac-ip {
                            advertise true
                            advertise-arp-nd-only-with-mac-table-entry true
                        }
                        inclusive-mcast {
                            advertise true
                        }
                    }
                }
            }
        }
        bgp-vpn {
            bgp-instance 1 {
                route-target {
                    export-rt target:1:100
                    import-rt target:1:100
                }
            }
        }
    }
    bridge-table {
        mac-learning {
            admin-state enable
```

```
            aging {
                admin-state enable
                age-time 300
            }
        }
        proxy-arp {
            admin-state enable
            table-size 250
            dynamic-learning {
                admin-state enable
                age-time 2000
                send-refresh 2000
            }
            ip-duplication {
                monitoring-window 10
                num-moves 4
                hold-down-time 10
            }
        }
    }
```

9 EDA integration

Nokia's Event Driven Automation (EDA) platform is a cloud-native platform deployed on top of Kubernetes, leveraging the Kubernetes-provided declarative API, tooling, and the ecosystem around it. EDA can be deployed as a single or multimode cluster. The various components of the EDA/K8s tech stack are shown below, instantiated as Kubernetes pods.

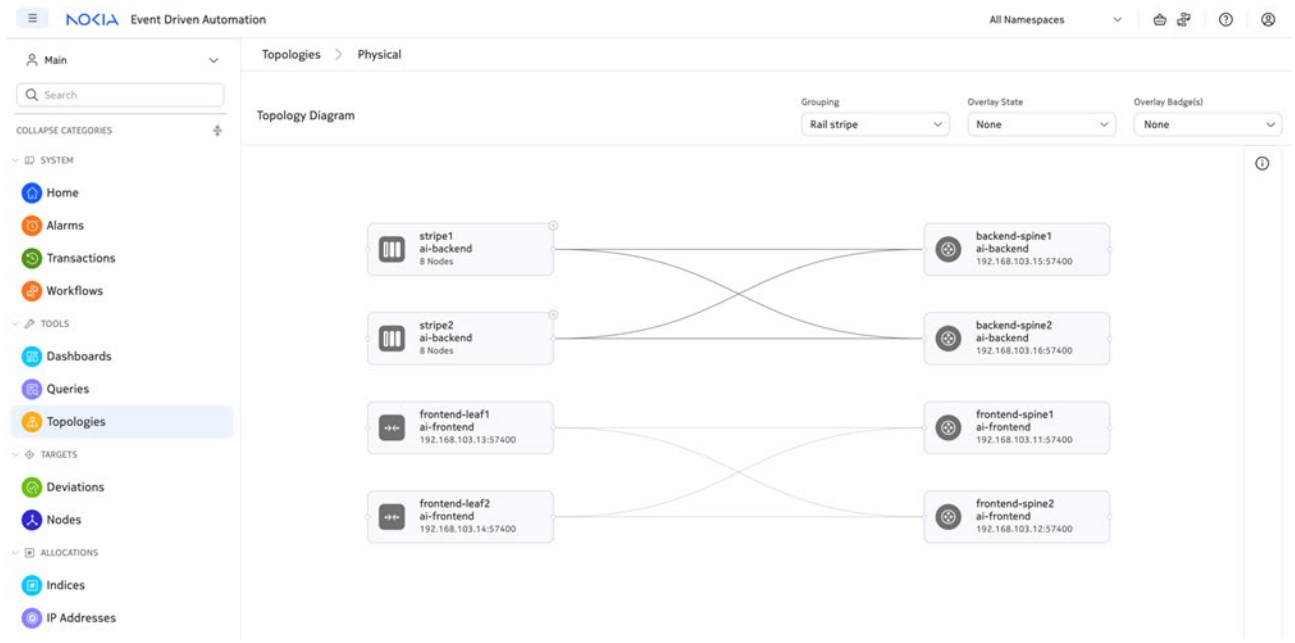
```
admin@server:~/nvd-hw/ai-nvd$ kubectl get pods -A

NAMESPACE            NAME                                          READY   STATUS    RESTARTS   AGE
cert-manager         cert-manager-777c6f8ff4-njdtn                 1/1     Running   0          6d4h
cert-manager         cert-manager-cainjector-6558fc6578-rbbft      1/1     Running   0          6d4h
cert-manager         cert-manager-webhook-6964489477-mn4x7         1/1     Running   0          6d4h
eda-system           cert-manager-csi-driver-pr6tr                 3/3     Running   0          6d4h
eda-system           eda-api-8584c74dc9-5fd8s                      1/1     Running   0          6d4h
eda-system           eda-appstore-6c8647f66-zmpzq                  1/1     Running   0          6d4h
eda-system           eda-asvr-766b5c79f4-v65r5                     1/1     Running   0          6d4h
eda-system           eda-bsvr-6d79d7d5cc-j54pt                     1/1     Running   0          6d4h
eda-system           eda-ce-94f86887d-ql9ng                        1/1     Running   0          6d4h
eda-system           eda-cert-checker-6b9b6f466b-fnjtd             1/1     Running   0          6d4h
eda-system           eda-fe-65d556bf64-kt5p6                       1/1     Running   0          6d4h
eda-system           eda-fluentbit-69pht                           1/1     Running   0          6d4h
eda-system           eda-fluentd-9b78f4c9f-hj4nj                   1/1     Running   0          6d4h
eda-system           eda-git-7487f97b5f-78c2v                      1/1     Running   0          6d4h
eda-system           eda-git-replica-6799f7bccb-xxw79              1/1     Running   0          6d4h
eda-system           eda-keycloak-579b449c96-ncd7t                 1/1     Running   0          6d4h
eda-system           eda-kx-59d79fff69-gkv8t                       1/1     Running   0          4d13h
eda-system           eda-metrics-server-788b466b77-hz7ct           1/1     Running   0          6d4h
eda-system           eda-npp-0                                     1/1     Running   0          5d8h
eda-system           eda-npp-1                                     1/1     Running   0          5d6h
eda-system           eda-npp-2                                     1/1     Running   0          5d5h
eda-system           eda-npp-3                                     1/1     Running   0          5d5h
eda-system           eda-npp-4                                     1/1     Running   0          5d5h
eda-system           eda-postgres-bb4c86cc9-k446j                  1/1     Running   0          6d4h
eda-system           eda-prw-76497cbcdc-b8gth                      1/1     Running   0          4d13h
```

```
eda-system           eda-px-5f5c66bc55-mkrxk                          1/1     Running   0          4d13h
eda-system           eda-sa-5f8c677f97-5b7hn                          1/1     Running   0          6d4h
eda-system           eda-sc-6778dbb78f-5vstb                          1/1     Running   0          6d4h
eda-system           eda-se-559f8894d6-l97hw                          1/1     Running   0          6d4h
eda-system           eda-toolbox-76886bc564-ftkvg                     1/1     Running   0          6d4h
eda-system           trust-manager-849b644bdf-v25b9                   1/1     Running   0          6d4h
eda-telemetry        alloy-b5648665-tqpcx                             1/1     Running   0          4d17h
eda-telemetry        grafana-f9dc9b4d7-brg7q                          1/1     Running   0          4d17h
eda-telemetry        kafka-6fbf94cbcb-dfj95                           1/1     Running   0          4d17h
eda-telemetry        loki-7449c899b8-pcmcw                            1/1     Running   0          4d17h
eda-telemetry        vms-victoria-metrics-single-server-0             1/1     Running   0          4d17h
kube-system          coredns-674b8bbfcf-dmxj4                         1/1     Running   0          6d4h
kube-system          coredns-674b8bbfcf-m7l7g                         1/1     Running   0          6d4h
kube-system          etcd-eda-demo-control-plane                      1/1     Running   0          6d4h
kube-system          kindnet-tnhhm                                    1/1     Running   0          6d4h
kube-system          kube-apiserver-eda-demo-control-plane            1/1     Running   0          6d4h
kube-system          kube-controller-manager-eda-demo-control-plane   1/1     Running   0          6d4h
kube-system          kube-proxy-mxhpq                                 1/1     Running   0          6d4h
kube-system          kube-scheduler-eda-demo-control-plane            1/1     Running   0          6d4h
local-path-storage   local-path-provisioner-7dc846544d-qr8th          1/1     Running   0          6d4h
metallb-system       controller-5cbffbc46b-v2wh4                      1/1     Running   0          6d4h
metallb-system       speaker-l7q4b                                    1/1     Running   0          6d4h
```

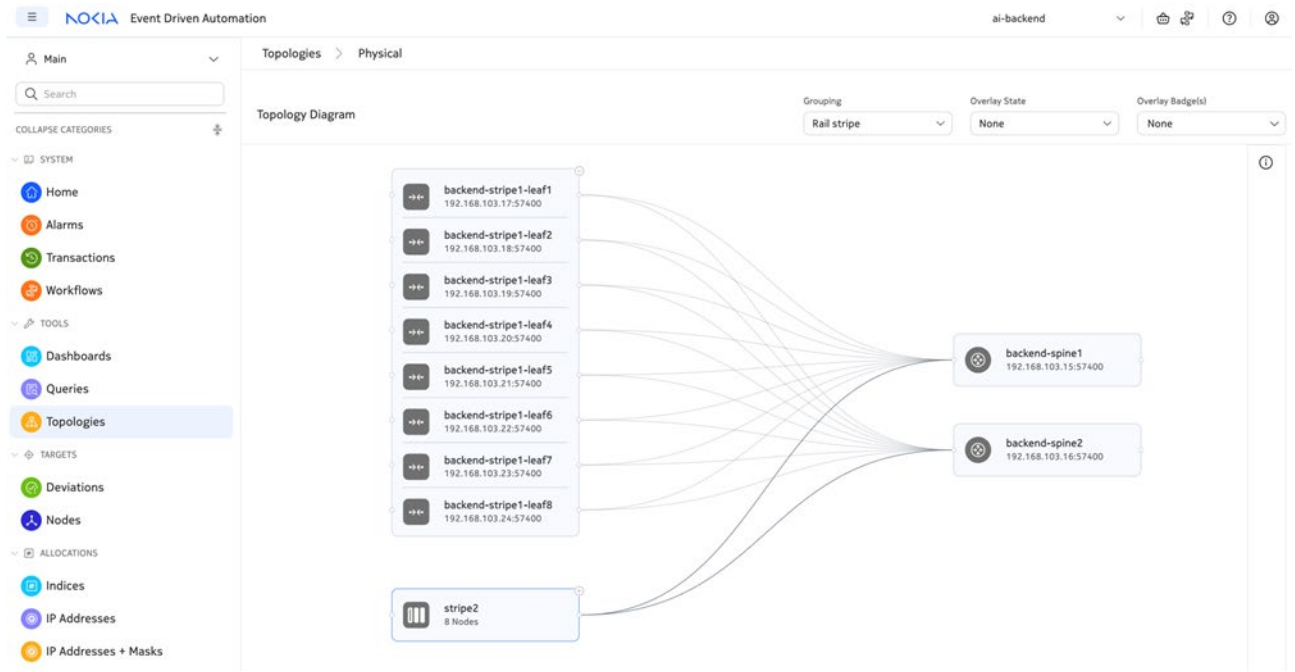
Some commonly used pods and their functionalities are:

- **eda-asvr** – The artifact server stores common artifacts used in EDA functionality. Examples include SR Linux image, SR Linux MD5 hash, YANG path .zip file, and so on. The availability of an artifact can be verified with **`kubectl get artifacts -A`**.
- **eda-bsvr** – The bootstrap server is responsible for all onboarding of nodes (virtual or hardware). Onboarding involves gNMI discovery, gNMI management, and instantiation of NPP pods for node lifecycle management.
- **eda-ce** – The configuration engine keeps track of all the dependencies among the application resources and runs the application intents when needed.
- **eda-npp** – The eda-npp pod is responsible for schema validation of the generated configuration. Additionally, it is responsible for all communications to the devices for both setting configuration and retrieving state.
- **eda-api** – The eda-api pod is the REST API server, which is accessible to end users and is consumed by the GUI.
- **eda-cx** – The sandbox controller spins up simulated nodes for building digital twins of the fabric (as the example above has the mode set to physical hardware only, the EDA CX functionality has been disabled).
- **eda-toolbox** – The toolbox provides tools such as edactl for insight into EDA transactions and an EDA topology generator that can generate a topology from a YAML file.

A consolidated view of the topology across both AI backend and converged frontend/storage fabrics (across both namespaces) is shown below.



An expanded view of just the AI backend fabric is shown below by selecting only the AI backend namespace. Each stripe is collapsed, shown only as an abstracted stripe view. However, you can expand each stripe by clicking on the + button, which shows all leafs that are a part of the stripe.



9.1 Backend fabric

9.1.1 Creating a namespace

A dedicated namespace is created for the backend fabric. All resources and fabric constructs related to the backend fabric exist within this namespace.

```
apiVersion: core.eda.nokia.com/v1
kind: Namespace
metadata:
  name: ai-backend
  namespace: "eda-system"
spec:
  description: namespace for Nokia AI fabric backend
```

9.1.2 Onboarding TopoNodes using ZTP

TopoNodes are the network switches onboarded into EDA and used for fabric deployment. In the case of the backend fabric, this includes two stripes of eight leafs each with two spines acting as stripe connectors.

```
// example of a spine being onboarded

apiVersion: core.eda.nokia.com/v1
kind: TopoNode
metadata:
  labels:
    eda.nokia.com/name: backend-spine1
    eda.nokia.com/role: spine
    eda.nokia.com/security-profile: managed
  name: backend-spine1
  namespace: ai-backend
spec:
  nodeProfile: real-srlinux-25.10.1
  npp:
    mode: normal
  onBoarded: true
  operatingSystem: srl
  platform: 7220 IXR-H5-32D
  productionAddress: {}
  serialNumber: [serial-number]
  version: 25.10.1

// example of a stripe leaf being onboarded

apiVersion: core.eda.nokia.com/v1
kind: TopoNode
metadata:
  labels:
    eda.nokia.com/name: backend-stripe1-leaf1
    eda.nokia.com/role: leaf
```

```
    eda.nokia.com/security-profile: managed
    eda.nokia.com/stripe: stripe1
  name: backend-stripe1-leaf1
  namespace: ai-backend
spec:
  nodeProfile: real-srlinux-25.10.1
  npp:
    mode: normal
  onBoarded: true
  operatingSystem: srl
  platform: 7220 IXR-H4-32D
  productionAddress: {}
  serialNumber: [serial-number]
  version: 25.10.1
```

9.1.3 Creating interfaces

All interfaces to be used in the fabric deployment must be first defined. This includes all leaf-to-spine, leaf-to-GPU, and LAG interfaces (shown in section 8.2).

```
// point-to-point link between leaf and spine

apiVersion: interfaces.eda.nokia.com/v1alpha1
kind: Interface
metadata:
  labels:
    eda.nokia.com/role: interSwitch
  name: backend-spine1-ethernet-1-1
  namespace: ai-backend
spec:
  enabled: true
  encapType: 'null'
  ethernet:
    speed: 400G
  lldp: true
  members:
    - enabled: true
      interface: ethernet-1-1
      lacpPortPriority: 32768
      node: backend-spine1
  type: interface
```

9.1.4 Creating TopoLinks

TopoLinks are used to define topological links between TopoNodes in the fabric. Thus, all links between the stripe leafs and the stripe connectors (spines) are defined as TopoLinks.

```
apiVersion: core.eda.nokia.com/v1
kind: TopoLink
metadata:
  labels:
```

```
    eda.nokia.com/role: interSwitch
  name: backend-spine1-e1-1-backend-stripe1-leaf1-e1-31
  namespace: ai-backend
spec:
  links:
    - local:
        interface: ethernet-1-1
        interfaceResource: backend-spine1-ethernet-1-1
        node: backend-spine1
      remote:
        interface: ethernet-1-31
        interfaceResource: backend-stripe1-leaf1-ethernet-1-31
        node: backend-stripe1-leaf1
      type: interSwitch
```

9.1.5 AI backend fabric deployment

The backend fabric is deployed using the AI backend app in EDA, which provisions multiple stripes connected using spines (Stripe Connector in the app). In this app, GPU isolation groups are used to enable segmentation and multitenancy through the instantiation of IP VRFs. The app itself requires:

- an ASN pool for BGP ASN assignment
- an IP pool for system0 assignment
- a link selector label for leaf-to-spine links
- a node selector label for the stripe connectors (which are the spines in this fabric)
- a stripe definition, including a GPU VLAN ID, a stripe ID, and a node selector label for the leafs that will constitute this stripe

```
// ASN index allocation pool manifest

apiVersion: core.eda.nokia.com/v1
kind: IndexAllocationPool
metadata:
  labels:
    eda.nokia.com/bootstrap: 'true'
  name: asn-pool-4byte
  namespace: ai-backend
spec:
  segments:
    - size: 3000
      start: 4200000000

// system0 IP allocation pool manifest

apiVersion: core.eda.nokia.com/v1
kind: IPAllocationPool
metadata:
  name: system0
  namespace: ai-backend
```

```
spec:
  segments:
    - subnet: 192.0.2.0/24

// AI backend app manifest

apiVersion: aifabrics.eda.nokia.com/v1alpha1
kind: Backend
metadata:
  name: backend-fabric
  namespace: ai-backend
spec:
  asnPool: asn-pool-4byte
  gpuIsolationGroups:
    - interfaceSelector:
        - eda.nokia.com/role=edge
      name: nvidia-gpus
  rocev2QoS:
    ecnMaxDropProbabilityPercent: 100
    ecnSlopeMaxThresholdPercent: 85
    ecnSlopeMinThresholdPercent: 30
    pfcDeadlockDetectionTimer: 750
    pfcDeadlockRecoveryTimer: 750
    queueMaximumBurstSize: 52110640
  stripeConnector:
    linkSelector:
      - eda.nokia.com/role=interSwitch
    name: stripe-connector
    nodeSelector:
      - eda.nokia.com/role=spine
  stripes:
    - gpuVlan: 1000
      name: stripe1
      nodeSelector:
        - eda.nokia.com/stripe=stripe1
      stripeID: 1
    - gpuVlan: 1000
      name: stripe2
      nodeSelector:
        - eda.nokia.com/stripe=stripe2
      stripeID: 2
  systemPoolIPV4: system0
```

9.2 Converged frontend and storage fabric

9.2.1 Creating a namespace

A new namespace is created specifically for the converged frontend/storage fabric.

```
apiVersion: core.eda.nokia.com/v1
kind: Namespace
metadata:
```

```
  name: ai-frontend
  namespace: "eda-system"
spec:
  description: namespace for Nokia AI fabric frontend
```

9.2.2 Onboarding TopoNodes

The nodes in the frontend fabric connect to the storage (WEKA) cluster and the storage/frontend NICs of the NVIDIA H200 GPUs.

```
apiVersion: core.eda.nokia.com/v1
kind: TopoNode
metadata:
  labels:
    eda.nokia.com/name: frontend-leaf1
    eda.nokia.com/role: leaf
    eda.nokia.com/security-profile: managed
  name: frontend-leaf1
  namespace: ai-frontend
spec:
  nodeProfile: real-srlinux-25.10.1
  npp:
    mode: normal
  onBoarded: true
  operatingSystem: srl
  platform: 7220 IXR-D5
  productionAddress: {}
  serialNumber: [serial-number]
  version: 25.10.1
```

9.2.3 Creating interfaces

All interfaces to be used in the fabric deployment must be first defined. This includes all leaf-to-spine and LAG interfaces for the storage cluster and the storage NICs of the GPUs.

```
// spine-facing interface on frontend-leaf1

apiVersion: interfaces.eda.nokia.com/v1alpha1
kind: Interface
metadata:
  labels:
    eda.nokia.com/role: interSwitch
  name: frontend-leaf1-ethernet-1-31
  namespace: ai-frontend
spec:
  description: to frontend-spine1 eth-1/1/1
  enabled: true
  encapType: 'null'
  lldp: true
  members:
    - enabled: true
```

```
      interface: ethernet-1-31
      lacpPortPriority: 32768
      node: frontend-leaf1
  type: interface

// LAG interface to storage NICs of GPUs

apiVersion: interfaces.eda.nokia.com/v1alpha1
kind: Interface
metadata:
  labels:
    eda.nokia.com/role: edge
    eda.nokia.com/type: gpu
  name: lag-gpu-server1
  namespace: ai-frontend
spec:
  enabled: true
  encapType: dot1q
  lag:
    lacp:
      interval: fast
      lacpFallback:
        mode: static
        timeout: 60
      mode: active
      systemIdMac: '00:00:00:00:00:11'
      systemPriority: 32768
    minLinks: 1
    multihoming:
      esi: auto
      mode: all-active
      reloadDelayTimer: 100
    type: lacp
  lldp: true
  members:
    - aggregateId: '1'
      enabled: true
      interface: ethernet-1-1-1
      lacpPortPriority: 32768
      node: frontend-leaf1
    - aggregateId: '1'
      enabled: true
      interface: ethernet-1-1-1
      lacpPortPriority: 32768
      node: frontend-leaf2
  type: lag
```

9.2.4 Creating TopoLinks

TopoLinks are used to define topological links between TopoNodes in the fabric. Thus, all links between the stripe leafs and the stripe connectors (spines) are defined as TopoLinks.

```
apiVersion: core.eda.nokia.com/v1
kind: TopoLink
metadata:
  labels:
    eda.nokia.com/role: interSwitch
  name: frontend-spine1-e1-1-1-frontend-leaf1-e1-31
  namespace: ai-frontend
spec:
  links:
    - local:
        interface: ethernet-1-1-1
        interfaceResource: frontend-spine1-ethernet-1-1-1
        node: frontend-spine1
      remote:
        interface: ethernet-1-31
        interfaceResource: frontend-leaf1-ethernet-1-31
        node: frontend-leaf1
      type: interSwitch
```

9.2.5 Fabric deployment

The fabric deployed for the converged frontend/storage network follows the 3-stage EVPN VXLAN Nokia Validated Design, found at the following location, provisioned with point-to-point interfaces between the leafs and the spines using IPv6 unnumbered (IPv6 link-local addressing) and a single MP-BGP session between the leafs and the spines for IPv4, IPv6, and EVPN address families and subsequent address families:

https://documentation.nokia.com/cgi-bin/dbaccessfilename.cgi/3HE21632AAAATQZZA_V1_Nokia%20Validated%20Design:%203-stage%20EVPN%20VXLAN%20Fabric%20.pdf

```
apiVersion: fabrics.eda.nokia.com/v1alpha1
kind: Fabric
metadata:
  name: frontend-fabric
  namespace: ai-frontend
spec:
  interSwitchLinks:
    linkSelector:
      - eda.nokia.com/role=interSwitch
    unnumbered: IPV6
  leafs:
    leafNodeSelector:
      - eda.nokia.com/role=leaf
  overlayProtocol:
    protocol: EBGP
  spines:
    spineNodeSelector:
      - eda.nokia.com/role=spine
  systemPoolIPV4: system0
  underlayProtocol:
```

```
    bfd:
      desiredMinTransmitInt: 250000
      detectionMultiplier: 3
      enabled: true
      minEchoReceiveInterval: 250000
      requiredMinReceive: 250000
    bgp:
      asnPool: asn-pool
    protocol:
      - EBGP
```

9.2.6 Virtual networks for bridge domains (MAC VRFs) and VLANs

Virtual Networks (VNETs) are created to deploy bridge domains and VLANs in the frontend/storage fabric for communication between the GPU frontend/storage NICs and the storage cluster (for access to storage and remote file systems) and between the frontend server and the GPU frontend/storage NICs (for dispatching jobs from the frontend server to the GPUs).

```
// storage VNET

apiVersion: services.eda.nokia.com/v1
kind: VirtualNetwork
metadata:
  name: storage
  namespace: ai-frontend
spec:
  bridgeDomains:
    - name: mac-vrf-storage
      spec:
        evi: 100
        eviPool: evi-pool
        l2proxyARPND:
          dynamicLearning:
            ageTime: 2000
            enabled: true
            sendRefresh: 2000
          ipDuplication:
            enabled: true
            holdDownTime: 10
            monitoringWindow: 10
            numMoves: 4
          proxyARP: true
          proxyND: false
          tableSize: 250
        macAging: 300
        macLearning: true
        tunnelIndexPool: tunnel-index-pool
        type: EVPNVXLAN
        vni: 100
        vniPool: vni-pool
  vlans:
    - name: storage
```

```
      spec:
        bridgeDomain: mac-vrf-storage
        interfaceSelector:
          - eda.nokia.com/type=storage
        vlanID: untagged
    - name: gpu
      spec:
        bridgeDomain: mac-vrf-storage
        interfaceSelector:
          - eda.nokia.com/type=gpu
        vlanID: '100'

// frontend VNET

apiVersion: services.eda.nokia.com/v1
kind: VirtualNetwork
metadata:
  name: frontend
  namespace: ai-frontend
spec:
  bridgeDomains:
    - name: mac-vrf-frontend
      spec:
        evi: 200
        eviPool: evi-pool
        l2proxyARPND:
          dynamicLearning:
            ageTime: 2000
            enabled: true
            sendRefresh: 2000
          ipDuplication:
            enabled: true
            holdDownTime: 10
            monitoringWindow: 10
            numMoves: 4
          proxyARP: true
          proxyND: false
          tableSize: 250
        macAging: 300
        macLearning: true
        tunnelIndexPool: tunnel-index-pool
        type: EVPNVXLAN
        vni: 200
        vniPool: vni-pool
  vlans:
    - name: gpu-frontend
      spec:
        bridgeDomain: mac-vrf-frontend
        interfaceSelector:
          - eda.nokia.com/type=gpu
        vlanID: '200'
    - name: frontend-server
      spec:
        bridgeDomain: mac-vrf-frontend
```

```
        interfaceSelector:
          - eda.nokia.com/type=frontend-server
        vlanID: '200'
```

9.2.7 EDA configlets

This AI NVD requires two configlets:

- configlet to enable DLB
- configlet to modify QoS

```
// DLB configlet

apiVersion: config.eda.nokia.com/v1alpha1
kind: Configlet
metadata:
  name: dlb
  namespace: ai-backend
spec:
  endpointSelector:
    - eda.nokia.com/role=leaf
  operatingSystem: srl
  priority: 100
  configs:
    - path: .system
      operation: Create
      config: |-
        {
            "load-balancing": {
            "dynamic": {
              "flowset-size": "256",
              "inactivity-timer": 50,
              "mode": "flow-dynamic",
              "link-quality-sampling-interval": 5,
              "weighting-factor": {
                "port-utilization": 70,
                "queue-utilization": 20,
                "itm-utilization": 10
              }
            }
          }
        }
---
apiVersion: config.eda.nokia.com/v1alpha1
kind: Configlet
metadata:
  name: ip-load-balance-network-instance
  namespace: ai-backend
spec:
  endpointSelector:
    - eda.nokia.com/role=leaf
  operatingSystem: srl
```

```
  priority: 100
  configs:
    - path: .network-instance{.name=="default"}
      operation: Create
      config: |-
        {
            "ip-load-balancing": {
            "dynamic-load-balancing": {
              "prefix": [
                {
                  "ip-prefix": "::/0"
                }
              ]
            }
          }
        }
    - path: .network-instance{.name=="nvidia-gpus"}
      operation: Create
      config: |-
        {
            "ip-load-balancing": {
            "dynamic-load-balancing": {
              "prefix": [
                {
                  "ip-prefix": "::/0"
                }
              ]
            }
          }
        }

// QOS configlet

apiVersion: config.eda.nokia.com/v1alpha1
kind: Configlet
metadata:
  name: qos-mbs-q0
  namespace: ai-backend
spec:
  endpointSelector:
    - eda.nokia.com/role=leaf
    - eda.nokia.com/role=spine
  operatingSystem: srl
  priority: 100
  configs:
    - path: .qos.buffer-management.buffer-allocation-profile{.name=="egress-backend-
backend-fabric"}.queues
      operation: Update
      config: |-
        {
          "queue": [
            {
              "queue-name": "unicast-0",
              "maximum-burst-size": 5211064
```

```
            },
            {
              "queue-name": "unicast-6",
              "maximum-burst-size": 52110640
            }

          ]
        }
    - path: .qos.buffer-management.buffer-allocation-profile{.name=="ingress-backend-
backend-fabric"}.queues
      operation: Update
      config: |-
        {
          "pfc-queue": [
            {
              "pfc-queue-name": "pfc-3",
              "maximum-burst-size": 52110640
            }
          ]
        }

---
```

10 Test summary

#	Feature	SRL v25.10.1	EDA 25.12.2	
		Validated	Validated	Configlet
1	RA for IPv6 link-local communication	Yes	Yes	No
2	Disable IPv4 check on IPv6 only interfaces	Yes	Yes	No
3	BGP for IPv6 unicast AFI/SAFI	Yes	Yes	No
4	BGP dynamic neighbors using IPv6 ND	Yes	Yes	No
5	BGP multipath for IPv6 unicast	Yes	Yes	No
6	BGP peer groups	Yes	Yes	No
7	BGP export and import policies	Yes	Yes	No
8	Export routing policies for IPv4 system0 addresses	Yes	Yes	No
9	Import routing policies for IPv4 unicast BGP	Yes	Yes	No
10	Platform-specific jumbo MTUs – H5	Yes	Yes	No
11	Platform-specific jumbo MTUs – H4	Yes	Yes	No

12	Platform-specific jumbo MTUs – D5	Yes	Yes	No
13	BGP failure detection with BFD (frontend)	Yes	Yes	No
14	BFD fast failover for BGP of 300 ms (frontend)	Yes	Yes	No
15	LLDP for neighbor adjacency	Yes	Yes	No
16	Layer 3 interfaces for GPU connectivity (backend)	Yes	Yes	No
17	IP VRFs for Layer 3 isolation	Yes	Yes	No
18	DCQCN (backend)	Yes	Yes	Yes
19	DLB (backend)	Yes	Yes	Yes
20	QP hashing (backend)	Yes	Yes	No
21	EDA - onboard physical hardware nodes (7220 IXR spines and leafs) with ZTP	Yes	Yes	No
22	EDA - deploy backend fabric with AI Backends app	Yes	Yes	No
23	EDA - deploy BGP for AI backend fabric with IPv6 AFI/SAFI	Yes	Yes	No
24	EDA - deploy Layer 2 LAGs in active/active mode for frontend/storage network	Yes	Yes	No
25	EDA – deploy Layer 2 LAGs in active/active mode for GPU storage NICs	Yes	Yes	No
26	EDA - deploy Layer 3 GPU-facing interfaces with IPv6 addressing	Yes	Yes	No
27	Streaming telemetry - architecture	Yes	Yes	Not applicable
28	Installing streaming telemetry stack using Helm	Yes	Yes	Not applicable
29	Export metrics from EDA	Yes	Yes	Not applicable
30	Dashboard – frontend, backend networks, node level	Yes	Yes	Not applicable
31	Centralized syslogs	Not applicable	Yes	Yes
32	EDA and containerlab digital twin	Yes	Yes	Yes
Server validation				
33	GPU server Netplan IPv6 configuration	Yes		
34	Storage cluster IP configuration	Yes		
35	Frontend server IP configuration	Yes		

NCCL and ROCEv2 perf test		
36	ib_send_bw for different packet sizes 256 to 4096 bytes	Yes
37	ib_read_bw average for all interfaces – RoCEv2 4096-byte packets	Yes
38	Ib_write_bw average for all interfaces – RoCEv2 4096-byte packets	Yes
39	NCCL performance per algorithm	Yes
MLCommons benchmarking – training		
40	Llama 2 70B /32 accelerators	Yes
MLCommons benchmarking – inference		
41	BERT – 8– offline mode	Yes

11 Validation

11.1 Network validation

11.1.1 IPv6 unnumbered validation

The links between the leafs and the spines are configured for IPv6 RA, auto-generating IPv6 link-local addresses per interface and using IPv6 Neighbor Discovery (ND) to resolve a peer's address.

```
A:admin@backend-stripe1-leaf1# info from state interface ethernet-1/31 subinterface 0 ipv6
    admin-state enable
    address fe80::4e62:cdff:feb3:6d37/64 {
        type link-local-unicast
        origin link-layer
        status preferred
    }
    neighbor-discovery {
        duplicate-address-detection true
        reachable-time 30
        stale-time 14400
        learn-unsolicited none
        proxy-nd false
        neighbor fe80::3200:fcff:fe2e:1c82 {
            link-layer-address 30:00:FC:2E:1C:82
            origin dynamic
            is-router true
            current-state stale
            next-state-time "2025-12-01T13:02:15.624Z (3 hours from now)"
            datapath-programming {
```

```
                status success
            }
        }
        limit {
            log-only false
            warning-threshold-pct 90
        }
    }
    router-advertisement {
        router-role {
            admin-state enable
            current-hop-limit 64
            managed-configuration-flag false
            other-configuration-flag false
            max-advertisement-interval 10
            min-advertisement-interval 4
            reachable-time 0
            retransmit-time 0
            router-lifetime 1800
        }
    }
```

11.1.2 BGP validation

Both the backend and frontend/storage fabrics use a single MP-BGP session carrying multiple address families (and subsequent address families). For the frontend/storage fabric, the EVPN AFI/SAFI is also used.

```
// backend leaf BGP summary

A:admin@backend-stripe1-leaf1# show network-instance default protocols bgp neighbor
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
BGP neighbor summary for network-instance "default"
Flags: S static, D dynamic, L discovered by LLDP, B BFD enabled, - disabled, * slow
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
+--------------------------+-----------------------------------------+--------------------------+---------+--------------+----------------------+-----
----------------+------------------+-----------------------------------------+
|         Net-Inst         |                  Peer                   |           Group          |  Flags  |   Peer-AS    |        State         |
Uptime        |     AFI/SAFI     |              [Rx/Active/Tx]             |
+==========================+=========================================+==========================+=========+==============+======================+=====
================+==================+=========================================+
| default                  | fe80::3200:fcff:fe2e:1c82%ethernet-     | bgpgroup-ebgp-stripe-    | D       | 4200000016   | established          |
5d:7h:38m:53s        | ipv4-unicast     | [16/16/2]                               |
|                          | 1/31.0                                  | connector                |         |              |                      |
| ipv6-unicast     | [15/15/1]                               |
| default                  | fe80::3200:fcff:feff:16b1%ethernet-     | bgpgroup-ebgp-stripe-    | D       | 4200000016   | established          |
5d:7h:38m:53s        | ipv4-unicast     | [16/1/17]                               |
|                          | 1/32.0                                  | connector                |         |              |                      |
| ipv6-unicast     | [15/15/16]                              |
+--------------------------+-----------------------------------------+--------------------------+---------+--------------+----------------------+-----
----------------+------------------+-----------------------------------------+
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
Summary:
0 configured neighbors, 0 configured sessions are established, 0 disabled peers
2 dynamic peers

// frontend leaf BGP summary

A:admin@frontend-leaf1# show network-instance default protocols bgp neighbor
---------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
BGP neighbor summary for network-instance "default"
Flags: S static, D dynamic, L discovered by LLDP, B BFD enabled, - disabled, * slow
```

```
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------
+-------------------------+-----------------------------------------+--------------------------+---------+--------------+----------------------+-----
---------------+------------------+--------------------------------------+
|         Net-Inst        |                  Peer                   |          Group           |  Flags  |   Peer-AS    |        State         |
Uptime        |     AFI/SAFI      |            [Rx/Active/Tx]            |
+=========================+=========================================+==========================+=========+==============+======================+=====
===============+==================+======================================+
| default                 | fe80::3200:fcff:feff:12b1%ethernet-     | bgpgroup-ebgp-frontend-  | DB      | 100          | established          |
2d:19h:20m:13s       | evpn             | [75/0/141]                           |
|                         | 1/32.0                                  | fabric                   |         |              |                      |
| ipv4-unicast      | [2/2/3]                                 |
|                         |                                         |                          |         |              |                      |
| ipv6-unicast      | [0/0/0]                                 |
| default                 | fe80::8669:91ff:fe59:7e25%ethernet-     | bgpgroup-ebgp-frontend-  | DB      | 100          | established          |
2d:19h:20m:13s       | evpn             | [75/75/66]                           |
|                         | 1/31.0                                  | fabric                   |         |              |                      |
| ipv4-unicast      | [2/2/2]                                 |
|                         |                                         |                          |         |              |                      |
| ipv6-unicast      | [0/0/0]                                 |
+-------------------------+-----------------------------------------+--------------------------+---------+--------------+----------------------+-----
---------------+------------------+--------------------------------------+
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------
Summary:
0 configured neighbors, 0 configured sessions are established, 0 disabled peers
2 dynamic peers
```

11.1.3 BFD validation

BFD is used for fast-failover with sub-second convergence in the frontend/storage fabric.

```
A:admin@frontend-leaf1# info from state bfd network-instance default peer *
    peer 16387 {
        oper-state up
        ipv6-link-local-interface ethernet-1/31.0
        local-address fe80::a6ff:95ff:fe59:74a6
        remote-address fe80::8669:91ff:fe59:7e25
        remote-discriminator 16387
        subscribed-protocols BGP
        session-state UP
        remote-session-state UP
        last-state-transition "2025-11-28T13:07:01.295Z (2 days ago)"
        failure-transitions 0
        local-diagnostic-code NO_DIAGNOSTIC
        remote-diagnostic-code NO_DIAGNOSTIC
        remote-minimum-receive-interval 250000
        remote-control-plane-independent false
        active-transmit-interval 250000
        active-receive-interval 250000
        remote-multiplier 3
        async {
            last-packet-transmitted "2025-12-01T09:10:13.950Z (now)"
            last-packet-received "2025-12-01T09:10:13.884Z (now)"
            transmitted-packets 1236986
            received-packets 1237038
            up-transitions 1
        }
    }
    peer 16388 {
        oper-state up
        ipv6-link-local-interface ethernet-1/32.0
```

```
        local-address fe80::a6ff:95ff:fe59:74a7
        remote-address fe80::3200:fcff:feff:12b1
        remote-discriminator 16387
        subscribed-protocols BGP
        session-state UP
        remote-session-state UP
        last-state-transition "2025-11-28T13:07:01.695Z (2 days ago)"
        failure-transitions 0
        local-diagnostic-code NO_DIAGNOSTIC
        remote-diagnostic-code NO_DIAGNOSTIC
        remote-minimum-receive-interval 250000
        remote-control-plane-independent false
        active-transmit-interval 250000
        active-receive-interval 250000
        remote-multiplier 3
        async {
            last-packet-transmitted "2025-12-01T09:10:13.915Z (now)"
            last-packet-received "2025-12-01T09:10:13.958Z (now)"
            transmitted-packets 1237030
            received-packets 1237037
            up-transitions 1
        }
    }
```

11.1.4 GPU subnets route validation

The per-leaf GPU-facing interface subnets are aggregated into a /94 route that is advertised to the fabric via BGP in the backend network. As a result, every leaf in a stripe receives a /94 from every other leaf in the fabric. In this NVD, with 16 leafs, every leaf receives 15/94 routes for GPU-to-GPU connectivity via the fabric.

```
A:admin@backend-stripe1-leaf1# show network-instance default protocols bgp neighbor fe80::3200:fcff:fe2e:1c82%ethernet-1/31.0 advertised-routes ipv6
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
Peer        : fe80::3200:fcff:fe2e:1c82%ethernet-1/31.0, remote AS: 4200000016, local AS: 4200000014
Type        : dynamic
Description : None
Group       : bgpgroup-ebgp-stripe-connector
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
Origin codes: i=IGP, e=EGP, ?=incomplete
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------+
|                  Network                                Path-id                      Next Hop                  MED
LocPref                                AsPath                Origin       |
+===========================================================================================================================================================
===========================================+
| fd00:1:1:1::/94                             0                                  fe80::4e62:cdff:feb3:6            -
[4200000014]                    ?         |
|                                                                                d37
|
+-----------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------+
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
1 advertised BGP routes
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------

A:admin@backend-stripe1-leaf1# show network-instance default protocols bgp neighbor fe80::3200:fcff:fe2e:1c82%ethernet-1/31.0 received-routes ipv6
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
Peer        : fe80::3200:fcff:fe2e:1c82%ethernet-1/31.0, remote AS: 4200000016, local AS: 4200000014
Type        : dynamic
Description : None
Group       : bgpgroup-ebgp-stripe-connector
```

```
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------
Status codes: u=used, *=valid, >=best, x=stale, b=backup, w=unused-weight-only
Origin codes: i=IGP, e=EGP, ?=incomplete
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------+
|   Status                  Network                           Path-id                     Next Hop                  MED
LocPref                                    AsPath                Origin       |
+==============================================================================================================================================================================
==============================================================================================================+
|    u*>       fd00:1:2:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000013]                                |
|    u*>       fd00:1:3:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000012]                                |
|    u*>       fd00:1:4:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000011]                                |
|    u*>       fd00:1:5:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000010]                                |
|    u*>       fd00:1:6:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000009]                                |
|    u*>       fd00:1:7:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000008]                                |
|    u*>       fd00:1:8:1::/94                    0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000015]                                |
|    u*>       fd00:2:9:1:1::/94                  0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000005]                                |
|    u*>       fd00:2:10:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000006]                                |
|    u*>       fd00:2:11:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000007]                                |
|    u*>       fd00:2:12:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000000]                                |
|    u*>       fd00:2:13:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000001]                                |
|    u*>       fd00:2:14:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000002]                                |
|    u*>       fd00:2:15:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000003]                                |
|    u*>       fd00:2:16:1:1::/94                 0                                fe80::3200:fcff:fe2e:1            -
100                                  [4200000016,                      ?          |
|                                                                                  c82%ethernet-1/31.0
4200000004]                                |
+------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------+
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------
15 received BGP routes : 15 used 15 valid
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------
```

11.1.5 IP VRFs import and export validation

Every GPU is mapped to a GPU isolation group (IP VRF) depending on the use case—for a single-tenant environment, all GPUs are mapped to the same IP VRF. However, the backend fabric is an IP

fabric where the IP VRFs are not extended to the spines. Consequently, routes must be leaked from the IP VRF table to the global table and then advertised via BGP to its peers.

```
A:admin@backend-stripe1-leaf1# show network-instance nvidia-gpus route-table ipv6-unicast summary
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
IPv6 unicast route table of network instance nvidia-gpus
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
+-------------------------------------+-------+------------+--------------------+----------+----------+---------+------------+--------------------
---+----------------------+----------------------+-----------------------+
|               Prefix                |  ID   | Route Type |     Route Owner    |  Active  |  Origin  | Metric  |    Pref    |    Next-hop (Type)
|   Next-hop Interface   | Backup Next-hop (Type) |    Backup Next-hop     |
|                                     |       |            |                    |          | Network  |         |            |
|                        |                      |       Interface        |
|                                     |       |            |                    |          | Instance |         |            |
|                        |                      |                        |
+=====================================+=======+============+====================+==========+==========+=========+============+====================
===+======================+======================+=======================+
| fd00:1:1:1:0:1::/96                 | 7     | local      | net_inst_mgr       | True     | nvidia-gpus | 0       | 0          |
fd00:1:1:1:0:1:0:1     | ethernet-1/1.1000    |                        |
|                                     |       |            |                    |          |          |         |            | (direct)
|                        |                      |                        |
| fd00:1:1:1:0:1:0:1/128              | 7     | host       | net_inst_mgr       | True     | nvidia-gpus | 0       | 0          | None
| None                   |                      |                        |
| fd00:1:1:1:0:1:0:2/128              | 7     | arp-nd     | arp_nd_mgr         | True     | nvidia-gpus | 0       | 1          |
fd00:1:1:1:0:1:0:2     | ethernet-1/1.1000    |                        |
|                                     |       |            |                    |          |          |         |            | (direct)
|                        |                      |                        |
| fd00:1:1:1:0:2::/96                 | 8     | local      | net_inst_mgr       | True     | nvidia-gpus | 0       | 0          |
fd00:1:1:1:0:2:0:1     | ethernet-1/2.1000    |                        |
|                                     |       |            |                    |          |          |         |            | (direct)
|                        |                      |                        |
| fd00:1:1:1:0:2:0:1/128              | 8     | host       | net_inst_mgr       | True     | nvidia-gpus | 0       | 0          | None
| None                   |                      |                        |
| fd00:1:1:1:0:2:0:2/128              | 8     | arp-nd     | arp_nd_mgr         | True     | nvidia-gpus | 0       | 1          |
fd00:1:1:1:0:2:0:2     | ethernet-1/2.1000    |                        |
|                                     |       |            |                    |          |          |         |            | (direct)
|                        |                      |                        |
| fd00:1:2:1::/94                     | 0     | bgp        | bgp_mgr            | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0     |                        |
|                                     |       |            |                    |          |          |         |            | c82 (direct)
| ethernet-1/32.0        |                      |                        |
|                                     |       |            |                    |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                        |
|                                     |       |            |                    |          |          |         |            | 6b1 (direct)
|                        |                      |                        |
| fd00:1:3:1::/94                     | 0     | bgp        | bgp_mgr            | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0     |                        |
|                                     |       |            |                    |          |          |         |            | c82 (direct)
| ethernet-1/32.0        |                      |                        |
|                                     |       |            |                    |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                        |
|                                     |       |            |                    |          |          |         |            | 6b1 (direct)
|                        |                      |                        |
| fd00:1:4:1::/94                     | 0     | bgp        | bgp_mgr            | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0     |                        |
|                                     |       |            |                    |          |          |         |            | c82 (direct)
| ethernet-1/32.0        |                      |                        |
|                                     |       |            |                    |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                        |
|                                     |       |            |                    |          |          |         |            | 6b1 (direct)
|                        |                      |                        |
| fd00:1:5:1::/94                     | 0     | bgp        | bgp_mgr            | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0     |                        |
|                                     |       |            |                    |          |          |         |            | c82 (direct)
| ethernet-1/32.0        |                      |                        |
|                                     |       |            |                    |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                        |
|                                     |       |            |                    |          |          |         |            | 6b1 (direct)
|                        |                      |                        |
| fd00:1:6:1::/94                     | 0     | bgp        | bgp_mgr            | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0     |                        |
|                                     |       |            |                    |          |          |         |            | c82 (direct)
| ethernet-1/32.0        |                      |                        |
|                                     |       |            |                    |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                        |
|                                     |       |            |                    |          |          |         |            | 6b1 (direct)
|                        |                      |                        |
| fd00:1:7:1::/94                     | 0     | bgp        | bgp_mgr            | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0     |                        |
|                                     |       |            |                    |          |          |         |            | c82 (direct)
| ethernet-1/32.0        |                      |                        |
|                                     |       |            |                    |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                        |
|                                     |       |            |                    |          |          |         |            | 6b1 (direct)
|                        |                      |                        |
```

```
| fd00:1:8:1::/94                      | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:9:1:1::/94                    | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:10:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:11:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:12:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:13:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:14:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:15:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
| fd00:2:16:1:1::/94                   | 0     | bgp        | bgp_mgr             | True     | default  | 0       | 170        |
fe80::3200:fcff:fe2e:1 | ethernet-1/31.0       |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | c82 (direct)
| ethernet-1/32.0       |                      |                       |
|                                      |       |            |                     |          |          |         |            |
fe80::3200:fcff:feff:1 |                      |                       |                     |          |          |         |            |
|                                      |       |            |                     |          |          |         |            | 6b1 (direct)
|                       |              |                       |
+--------------------------------------+-------+------------+---------------------+----------+----------+---------+------------+---------------------
---+----------------------+----------------------+-----------------------+
-------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
IPv6 routes total                   : 21
IPv6 prefixes with active routes    : 21
IPv6 prefixes with active ECMP routes: 15
```

11.1.6 MAC VRFs validation for frontend/storage fabric

MAC VRFs are used for communication between the GPU frontend/storage NICs and the storage cluster and frontend server, providing a Layer 2 path between them.

```
A:admin@frontend-leaf1# show network-instance mac-vrf-* summary
+-------------------------------------------------+-------------------------+-------------------------+-------------------------+----------------
-----------------------------------+-----------------------------------------------------------------------------+
|                      Name                       |          Type           |       Admin state       |        Oper state       |
Router id                    |                          Description                          |
+=================================================+=========================+=========================+=========================+================
===================================+=============================================================================+
| mac-vrf-frontend                                | mac-vrf                 | enable                  | up                      | N/A
| mac-vrf-frontend                                                          |
| mac-vrf-storage                                 | mac-vrf                 | enable                  | up                      | N/A
| mac-vrf-storage                                                           |
+-------------------------------------------------+-------------------------+-------------------------+-------------------------+----------------
-----------------------------------+-----------------------------------------------------------------------------+

A:admin@frontend-leaf1# show network-instance mac-vrf-* bridge-table mac-table all
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
Mac-table of network instance mac-vrf-frontend
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
+-------------------+-------------------------------------------------------------------------------+------------+----------------------+---------+----
----+-------------------------------------------------------------------------+
|      Address      |                                 Destination                                   | Dest Index |         Type         | Active  |
Aging  |                              Last Update                                |
+===================+===============================================================================+============+======================+=========+====
====+=========================================================================+
| 5E:B4:EB:6D:5B:86  | lag4.200                                                                      | 22         | learnt               | true    | 180
| 2025-12-01T09:37:26.000Z                                                |
| A4:FF:95:59:62:87  | vxlan-interface:vxlan0.501 vtep:192.0.2.1 vni:200                             | 4414668    | evpn-static          | true    | N/A
| 2025-11-28T13:07:05.000Z                                                |
| A4:FF:95:59:74:87  | reserved                                                                      | 0          | reserved             | false   | N/A
| 2025-11-26T00:04:44.000Z                                                |
+-------------------+-------------------------------------------------------------------------------+------------+----------------------+---------+----
----+-------------------------------------------------------------------------+
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
Mac-table of network instance mac-vrf-storage
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
+-------------------+-------------------------------------------------------------------------------+------------+----------------------+---------+----
----+-------------------------------------------------------------------------+
|      Address      |                                 Destination                                   | Dest Index |         Type         | Active  |
Aging  |                              Last Update                                |
+===================+===============================================================================+============+======================+=========+====
====+=========================================================================+
| 5E:B4:EB:6D:5B:86  | lag4.100                                                                      | 9          | learnt               | true    | 265
| 2025-11-27T00:32:34.000Z                                                |
| 82:0C:97:65:F8:6E  | lag1.100                                                                      | 6          | learnt               | true    | 265
| 2025-11-27T01:07:08.000Z                                                |
| A4:FF:95:59:62:87  | vxlan-interface:vxlan0.500 vtep:192.0.2.1 vni:100                             | 4414667    | evpn-static          | true    | N/A
| 2025-11-28T13:07:05.000Z                                                |
| A4:FF:95:59:74:87  | reserved                                                                      | 0          | reserved             | false   | N/A
| 2025-11-26T00:04:43.000Z                                                |
| A6:2F:05:7F:C9:F2  | lag3.100                                                                      | 8          | evpn                 | true    | N/A
| 2025-12-01T09:18:53.000Z                                                |
| D6:12:E4:AB:4B:B8  | lag2.100                                                                      | 7          | learnt               | true    | 265
| 2025-11-27T01:11:08.000Z                                                |
| E0:9D:73:BD:86:08  | lag11.4096                                                                    | 16         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:BD:89:F8  | lag5.4096                                                                     | 10         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:BD:8A:88  | lag6.4096                                                                     | 11         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:BD:98:E8  | lag8.4096                                                                     | 13         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:C1:1C:0C  | lag9.4096                                                                     | 14         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:C1:1D:3C  | lag12.4096                                                                    | 17         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:C1:25:C4  | lag10.4096                                                                    | 15         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
| E0:9D:73:C1:27:A4  | lag7.4096                                                                     | 12         | learnt               | true    | 265
| 2025-11-26T00:04:44.000Z                                                |
+-------------------+-------------------------------------------------------------------------------+------------+----------------------+---------+----
----+-------------------------------------------------------------------------+
Total Irb Macs                :    0 Total    0 Active
Total Static Macs             :    0 Total    0 Active
Total Duplicate Macs          :    0 Total    0 Active
Total Learnt Macs             :   12 Total   12 Active
Total Evpn Macs               :    1 Total    1 Active
Total Evpn static Macs        :    2 Total    2 Active
```

```
Total Irb anycast Macs         :    0 Total    0 Active
Total Proxy Antispoof Macs     :    0 Total    0 Active
Total Reserved Macs            :    2 Total    0 Active
Total Eth-cfm Macs             :    0 Total    0 Active
Total Irb Vrrps                :    0 Total    0 Active
```

11.1.7 LAGs and Ethernet segments validation for frontend/storage fabric

LAGs are used to connect to the GPU storage NICs, the WEKA cluster, and the frontend server. All these systems have dual connectivity: one link to frontend-leaf1 and another to frontend-leaf2.

```
// LACP state to NVIDIA H200 server1, Weka node1 and frontend server respectively

A:admin@frontend-leaf1# show lag lag{1,5,13} lacp-state
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
LACP State for lag1
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
Lag Id         : lag1
Interval       : FAST
Mode           : ACTIVE
System Id      : 00:00:00:00:00:11
System Priority: 32768
+----------------+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+-----------------
----+---------------------+---------------------+---------------------+---------------------+---------------------+
|    Members     |      Oper state     |       Activity      |       Timeout       |        State        |      System Id      |       Oper key
|      Partner Id      |     Partner Key     |       Port No       |   Partner Port No    |
+================+=====================+=====================+=====================+=====================+=====================+=================
====+=====================+=====================+=====================+=====================+=====================+
| ethernet-1/1/1 | up                  | ACTIVE              | SHORT               | IN_SYNC/True/True/Tr | 00:00:00:00:00:11   | 3
| 82:0C:97:65:F8:6E   | 31                  | 2                   | 2                   |
|                |                     |                     |                     | ue                  |                     |
|                |                     |                     |                     |                     |
+----------------+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+-----------------
----+---------------------+---------------------+---------------------+---------------------+---------------------+
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
LACP State for lag5
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
Lag Id         : lag5
Interval       : FAST
Mode           : ACTIVE
System Id      : 00:00:00:00:00:15
System Priority: 32768
+----------------+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+-----------------
----+---------------------+---------------------+---------------------+---------------------+---------------------+
|    Members     |      Oper state     |       Activity      |       Timeout       |        State        |      System Id      |       Oper key
|      Partner Id      |     Partner Key     |       Port No       |   Partner Port No    |
+================+=====================+=====================+=====================+=====================+=====================+=================
====+=====================+=====================+=====================+=====================+=====================+
| ethernet-1/5/1 | up                  | ACTIVE              | SHORT               | IN_SYNC/True/True/Tr | 00:00:00:00:00:15   | 6
| E0:9D:73:BD:89:F8   | 31                  | 6                   | 2                   |
|                |                     |                     |                     | ue                  |                     |
|                |                     |                     |                     |                     |
+----------------+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+-----------------
----+---------------------+---------------------+---------------------+---------------------+---------------------+
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
LACP State for lag13
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
Lag Id         : lag13
Interval       : FAST
Mode           : ACTIVE
System Id      : 00:00:00:00:00:23
System Priority: 32768
+----------------+---------------------+---------------------+---------------------+---------------------+---------------------+---------------------+-----------------
----+---------------------+---------------------+---------------------+---------------------+---------------------+
|    Members     |      Oper state     |       Activity      |       Timeout       |        State        |      System Id      |       Oper key
|      Partner Id      |     Partner Key     |       Port No       |   Partner Port No    |
+================+=====================+=====================+=====================+=====================+=====================+=================
====+=====================+=====================+=====================+=====================+=====================+
| ethernet-1/29  | up                  | ACTIVE              | SHORT               | OUT_SYNC/True/False/ | 00:00:00:00:00:23   | 1
| 00:00:00:00:00:00   | 32768               | 1                   | 0                   |
```

```
|             |                   |                   |                   | False              |                   |
|               |                     |                     |                     |                    |
+---------------+-------------------+-------------------+-------------------+--------------------+-------------------+--------------------
----+--------------------+--------------------+--------------------+--------------------+
--------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
```

11.1.8 QoS validation

Quality of service (QoS) is critical to ensure lossless fabrics for RoCEv2 traffic. Per-queue statistics help monitor any drops in specific queues and confirm if traffic is flowing in the correct queues while statistics for every priority-group indicates if any pause frames are being generated or received.

```
A:admin@backend-stripe1-leaf1# info from state qos interfaces interface ethernet
-1/3 output queues queue unicast-3 queue-statistics
    aggregate-statistics {
        last-clear "2026-02-11T19:28:31.023Z (15 days ago)"
        transmitted-packets 42149927273
        transmitted-octets 151191486256542
        dropped-packets 0
        dropped-octets 0
    }

--{ + running }--[  ]--
A:admin@backend-stripe1-leaf1# info from state qos interfaces interface ethernet
-1/3 pfc statistics pfc-priority 3
    pfc-pause-frames-received 0
    pfc-pause-frames-generated 74
    pfc-transitions 0
    deadlock-recovery-occurrences 0
```

Interface statistics show any input or output discarded packets, indicating possible congestion points in the fabric.

```
A:admin@backend-stripe1-leaf1# info from state interface ethernet-1/3 statistics

    in-packets 45486178944
    in-octets 163249771225736
    in-unicast-packets 45486118459
    in-broadcast-packets 0
    in-multicast-packets 60485
    in-discarded-packets 5727553
    in-error-packets 0
    in-fcs-error-packets 0
    out-packets 45388466623
    out-octets 163204277276180
    out-unicast-packets 45385387829
    out-broadcast-packets 0
    out-multicast-packets 3078794
    out-discarded-packets 1
    out-error-packets 0
    carrier-transitions 3
    last-clear "2026-02-10T03:50:08.370Z (16 days ago)"
```

11.2 EDA validation

11.2.1 EDA pods validation

Use **`kubectl get pods -A`** to ensure all pods are in a running state.

```
cse@d2vm-4~ kubectl get pods -A
NAMESPACE            NAME                                                READY   STATUS    RESTARTS   AGE
cert-manager         cert-manager-777c6f8ff4-njdtn                       1/1     Running   0          8d
cert-manager         cert-manager-cainjector-6558fc6578-rbbft            1/1     Running   0          8d
cert-manager         cert-manager-webhook-6964489477-mn4x7               1/1     Running   0          8d
eda-system           cert-manager-csi-driver-pr6tr                       3/3     Running   0          8d
eda-system           eda-api-8584c74dc9-5fd8s                            1/1     Running   0          8d
eda-system           eda-appstore-6c8647f66-zmpzq                        1/1     Running   0          8d
eda-system           eda-asvr-766b5c79f4-v65r5                           1/1     Running   0          8d
eda-system           eda-bsvr-6d79d7d5cc-j54pt                           1/1     Running   0          8d
eda-system           eda-ce-94f86887d-ql9ng                              1/1     Running   0          8d
eda-system           eda-cert-checker-6b9b6f466b-fnjtd                   1/1     Running   0          8d
eda-system           eda-fe-65d556bf64-kt5p6                             1/1     Running   0          8d
eda-system           eda-fluentbit-69pht                                 1/1     Running   0          8d
eda-system           eda-fluentd-9b78f4c9f-hj4nj                         1/1     Running   0          8d
eda-system           eda-git-7487f97b5f-78c2v                            1/1     Running   0          8d
eda-system           eda-git-replica-6799f7bccb-xxw79                    1/1     Running   0          8d
eda-system           eda-keycloak-579b449c96-ncd7t                       1/1     Running   0          8d
eda-system           eda-kx-59d79fff69-gkv8t                             1/1     Running   0          6d13h
eda-system           eda-metrics-server-788b466b77-hz7ct                 1/1     Running   0          8d
eda-system           eda-npp-0                                           1/1     Running   0          7d8h
eda-system           eda-npp-1                                           1/1     Running   0          7d6h
eda-system           eda-npp-2                                           1/1     Running   0          7d5h
eda-system           eda-npp-3                                           1/1     Running   0          7d5h
eda-system           eda-npp-4                                           1/1     Running   0          7d5h
eda-system           eda-postgres-bb4c86cc9-k446j                        1/1     Running   0          8d
eda-system           eda-prw-76497cbcdc-b8gth                            1/1     Running   0          6d13h
eda-system           eda-px-5f5c66bc55-mkrxk                             1/1     Running   0          6d13h
eda-system           eda-sa-5f8c677f97-5b7hn                             1/1     Running   0          8d
eda-system           eda-sc-6778dbb78f-5vstb                             1/1     Running   0          8d
eda-system           eda-se-559f8894d6-l97hw                             1/1     Running   0          8d
eda-system           eda-toolbox-76886bc564-ftkvg                        1/1     Running   0          8d
eda-system           trust-manager-849b644bdf-v25b9                      1/1     Running   0          8d
eda-telemetry        alloy-b5648665-tqpcx                                1/1     Running   0          6d17h
eda-telemetry        grafana-f9dc9b4d7-brg7q                             1/1     Running   0          6d17h
eda-telemetry        kafka-6fbf94cbcb-dfj95                              1/1     Running   0          6d17h
eda-telemetry        loki-7449c899b8-pcmcw                               1/1     Running   0          6d17h
eda-telemetry        vms-victoria-metrics-single-server-0                1/1     Running   0          6d17h
kube-system          coredns-674b8bbfcf-dmxj4                            1/1     Running   0          8d
kube-system          coredns-674b8bbfcf-m7l7g                            1/1     Running   0          8d
kube-system          etcd-eda-demo-control-plane                         1/1     Running   0          8d
kube-system          kindnet-tnhhm                                       1/1     Running   0          8d
kube-system          kube-apiserver-eda-demo-control-plane               1/1     Running   0          8d
kube-system          kube-controller-manager-eda-demo-control-plane      1/1     Running   0          8d
kube-system          kube-proxy-mxhpq                                    1/1     Running   0          8d
kube-system          kube-scheduler-eda-demo-control-plane               1/1     Running   0          8d
local-path-storage   local-path-provisioner-7dc846544d-qr8th             1/1     Running   0          8d
metallb-system       controller-5cbffbc46b-v2wh4                         1/1     Running   0          8d
metallb-system       speaker-l7q4b                                       1/1     Running   0          8d
```

11.2.2 TopoNode validation

Use **`kubectl get toponodes -A`** to ensure that all the nodes are onboarded and synced in EDA.

```
cse@d2vm-4~ kubectl get toponodes -A
NAMESPACE     NAME                  PLATFORM          VERSION   OS    ONBOARDED   MODE     NPP         NODE     AGE
ai-backend    backend-spine1        7220 IXR-H5-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-spine2        7220 IXR-H5-64O   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf1 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf2 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf3 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf4 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf5 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf6 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf7 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe1-leaf8 7220 IXR-H4-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf1 7220 IXR-H5-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf2 7220 IXR-H5-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf3 7220 IXR-H5-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf4 7220 IXR-H5-32D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf5 7220 IXR-H5-64D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf6 7220 IXR-H5-64D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf7 7220 IXR-H5-64D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-backend    backend-stripe2-leaf8 7220 IXR-H5-64D   25.10.1   srl   true        normal   Connected   Synced   7d6h
ai-frontend   frontend-leaf1        7220 IXR-D5       25.10.1   srl   true        normal   Connected   Synced   7d8h
ai-frontend   frontend-leaf2        7220 IXR-D5       25.10.1   srl   true        normal   Connected   Synced   7d8h
ai-frontend   frontend-spine1       7220 IXR-H5-64O   25.10.1   srl   true        normal   Connected   Synced   7d8h
ai-frontend   frontend-spine2       7220 IXR-H5-64O   25.10.1   srl   true        normal   Connected   Synced   7d8h
```

11.2.3 AI backend network validation

The following output shows the backend network with its associated nodes.

```
$kubectl describe Backend backend-fabric --namespace ai-backend
Name:         backend-fabric
Namespace:    ai-backend
Labels:       <none>
Annotations:  <none>
API Version:  aifabrics.eda.nokia.com/v1alpha1
Kind:         Backend
Metadata:
  Creation Timestamp:  2025-12-15T21:07:32Z
  Generation:          19
  Resource Version:    17408164
  UID:                 67c3aa36-a731-4b97-98e0-6b1b5c78e42f
Spec:
  Asn Pool:  asn-pool-4byte
  Gpu Isolation Groups:
    Interface Selector:
      eda.nokia.com/type=gpu-nvidia
    Name:  nvidia-gpus
  Ip MTU:  9000
  rocev2QoS:
    Ecn Max Drop Probability Percent:  100
    Ecn Slope Max Threshold Percent:   85
    Ecn Slope Min Threshold Percent:   30
    Pfc Deadlock Detection Timer:      750
    Pfc Deadlock Recovery Timer:       750
    Queue Maximum Burst Size:          52110640
```

```
  Stripe Connector:
    Link Selector:
      eda.nokia.com/role=interSwitch
    Name:  stripe-connector
    Node Selector:
      eda.nokia.com/role=spine
  Stripes:
    Gpu Vlan:  1000
    Name:      stripe1
    Node Selector:
      eda.nokia.com/stripe=stripe1
    Stripe ID:  1
    Gpu Vlan:   1000
    Name:       stripe2
    Node Selector:
      eda.nokia.com/stripe=stripe2
    Stripe ID:     2
  systemPoolIPV4:  system0
Status:
  Last Change:        2026-02-21T07:12:23.000Z
  Operational State:  up
  Stripe Connector:
    Name:  stripe-connector
    Stripe Connector Nodes:
      Node:                     backend-spine1
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-spine2
      Operating System:         srl
      Operating System Version:  25.10.1
  Stripes:
    Leaf Nodes:
      Node:                     backend-stripe2-leaf1
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf2
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf5
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf6
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf7
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf8
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf4
      Operating System:         srl
      Operating System Version:  25.10.1
      Node:                     backend-stripe2-leaf3
```

```
      Operating System:          srl
      Operating System Version:  25.10.1
    Name:                        stripe2
    Leaf Nodes:
      Node:                      backend-stripe1-leaf8
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf7
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf6
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf5
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf4
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf3
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf2
      Operating System:          srl
      Operating System Version:  25.10.1
      Node:                      backend-stripe1-leaf1
      Operating System:          srl
      Operating System Version:  25.10.1
    Name:                        stripe1
Events:                          <none>
```

11.2.4 Frontend network validation

Frontend fabric validation shows the nodes associated with the converged frontend/storage fabric, as well as the underlay and overlay protocols.

```
kubectl describe fabrics frontend-fabric --namespace ai-frontend
Name:         frontend-fabric
Namespace:    ai-frontend
Labels:       <none>
Annotations:  <none>
API Version:  fabrics.eda.nokia.com/v1alpha1
Kind:         Fabric
Metadata:
  Creation Timestamp:  2025-11-26T00:02:54Z
  Generation:          2
  Resource Version:    3106175
  UID:                 9e2ccf4d-34c5-4bb3-ab2a-0af2b3997d40
Spec:
  Inter Switch Links:
    Link Selector:
      eda.nokia.com/role=interSwitch
    Unnumbered:  IPV6
```

```
  Leafs:
    Leaf Node Selector:
      eda.nokia.com/role=leaf
  Overlay Protocol:
    Bgp:
    Protocol:  EBGP
  Spines:
    Spine Node Selector:
      eda.nokia.com/role=spine
  systemPoolIPV4:  system0
  Underlay Protocol:
    Bfd:
      Desired Min Transmit Int:   250000
      Detection Multiplier:       3
      Enabled:                    true
      Min Echo Receive Interval:  250000
      Required Min Receive:       250000
    Bgp:
      Asn Pool:  asn-pool
    Protocol:
      EBGP
Status:
  Border Leaf Nodes:
  Health:               100
  Health Score Reason:  Breakdown:
Metric "ISL Health", weight: 1, score: 100, calculation method: divide
Metric "DefaultRouter Health", weight: 1, score: 100, calculation method: divide

  Last Change:  2025-12-02T17:50:24.000Z
  Leaf Nodes:
    Node:                         frontend-leaf2
    Operating System:             srl
    Operating System Version:     25.10.1
    Underlay Autonomous System:  102
    Node:                         frontend-leaf1
    Operating System:             srl
    Operating System Version:     25.10.1
    Underlay Autonomous System:  101
  Operational State:              up
  Spine Nodes:
    Node:                         frontend-spine1
    Operating System:             srl
    Operating System Version:     25.10.1
    Underlay Autonomous System:  100
    Node:                         frontend-spine2
    Operating System:             srl
    Operating System Version:     25.10.1
    Underlay Autonomous System:  100
  Super Spine Nodes:
Events:  <none>
```

11.2.5 Data center network interfaces status

EDA provides a centralized view of data center network interface statuses. The interfaces are represented as Kubernetes objects of kind **`interface`**, spread across multiple namespaces. It provides a unified view of interface operational status, speed, and recent changes.

11.2.5.1 Frontend interfaces

This output shows the operational status of frontend network interfaces. It provides information about the interfaces enabled, link status, and speed.

```
kubectl get interfaces –namespace ai-frontend
NAMESPACE     NAME                                 ENABLED   OPERATIONAL STATE   SPEED   LAST CHANGE   AGE
ai-frontend   frontend-leaf1-ethernet-1-31         true      up                  400G    6d12h         14d
ai-frontend   frontend-leaf1-ethernet-1-32         true      up                  400G    13d           14d
ai-frontend   frontend-leaf2-ethernet-1-31         true      up                  400G    13d           14d
ai-frontend   frontend-leaf2-ethernet-1-32         true      up                  400G    13d           14d
ai-frontend   frontend-spine1-ethernet-1-1-1       true      up                  400G    6d12h         14d
ai-frontend   frontend-spine1-ethernet-1-1-2       true      up                  400G    13d           14d
ai-frontend   frontend-spine2-ethernet-1-1-1       true      up                  400G    13d           14d
ai-frontend   frontend-spine2-ethernet-1-1-2       true      up                  400G    13d           14d
ai-frontend   lag-frontend-server                  true      up                  40G     13d           14d
ai-frontend   lag-gpu-server1                      true      up                  200G    83m           14d
ai-frontend   lag-gpu-server2                      true      up                  200G    13d           14d
ai-frontend   lag-gpu-server3                      true      up                  200G    3d10h         14d
ai-frontend   lag-gpu-server4                      true      up                  200G    3d10h         14d
ai-frontend   lag-storage-server1                  true      up                  200G    6d12h         14d
ai-frontend   lag-storage-server2                  true      up                  200G    6d12h         14d
ai-frontend   lag-storage-server3                  true      up                  200G    13d           14d
ai-frontend   lag-storage-server4                  true      up                  200G    13d           14d
ai-frontend   lag-storage-server5                  true      up                  200G    13d           14d
ai-frontend   lag-storage-server6                  true      up                  200G    13d           14d
ai-frontend   lag-storage-server7                  true      up                  200G    13d           14d
ai-frontend   lag-storage-server8                  true      up                  200G    13d           14d
```

11.2.5.2 Backend interfaces

This output displays the status of the AI backend interfaces used within the AI cluster including GPU and storage paths. This output helps validate the status of interfaces and speed.

```
kubectl get interfaces –namespace ai-backend
NAMESPACE     NAME                                 ENABLED   OPERATIONAL STATE   SPEED   LAST CHANGE   AGE
ai-backend    backend-spine1-ethernet-1-1          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-2          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-25         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-26         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-27         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-28         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-29         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-3          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-30         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-31         true      up                  800G    3d12h         13d
ai-backend    backend-spine1-ethernet-1-32         true      up                  800G    12d           13d
ai-backend    backend-spine1-ethernet-1-4          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-5          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-6          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-7          true      up                  400G    13d           13d
ai-backend    backend-spine1-ethernet-1-8          true      up                  400G    13d           13d
```

ai-backend	backend-spine2-ethernet-1-1-1	true	up	400G	13d	13d
ai-backend	backend-spine2-ethernet-1-1-2	true	up	400G	13d	13d
ai-backend	backend-spine2-ethernet-1-2-1	true	up	400G	13d	13d
ai-backend	backend-spine2-ethernet-1-2-2	true	up	400G	13d	13d
ai-backend	backend-spine2-ethernet-1-25	true	up	800G	6d13h	13d
ai-backend	backend-spine2-ethernet-1-26	true	up	800G	12d	13d
ai-backend	backend-spine2-ethernet-1-27	true	up	800G	12d	13d
ai-backend	backend-spine2-ethernet-1-28	true	up	800G	12d	13d
ai-backend	backend-spine2-ethernet-1-29	true	up	800G	12d	13d
ai-backend	backend-spine2-ethernet-1-3-1	true	up	400G	13d	13d
ai-backend	backend-spine2-ethernet-1-3-2	true	up	400G	5d17h	13d
ai-backend	backend-spine2-ethernet-1-30	true	up	800G	12d	13d
ai-backend	backend-spine2-ethernet-1-31	true	up	800G	3d12h	13d
ai-backend	backend-spine2-ethernet-1-32	true	up	800G	12d	13d
ai-backend	backend-spine2-ethernet-1-4-1	true	up	400G	13d	13d
ai-backend	backend-spine2-ethernet-1-4-2	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf1-ethernet-1-1	true	up	400G	3d11h	13d
ai-backend	backend-stripe1-leaf1-ethernet-1-2	true	up	400G	3d12h	13d
ai-backend	backend-stripe1-leaf1-ethernet-1-3	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf1-ethernet-1-31	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf1-ethernet-1-32	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf1-ethernet-1-4	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf2-ethernet-1-1	true	up	400G	3d11h	13d
ai-backend	backend-stripe1-leaf2-ethernet-1-2	true	up	400G	3d12h	13d
ai-backend	backend-stripe1-leaf2-ethernet-1-3	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf2-ethernet-1-31	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf2-ethernet-1-32	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf2-ethernet-1-4	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf3-ethernet-1-1	true	up	400G	3d11h	13d
ai-backend	backend-stripe1-leaf3-ethernet-1-2	true	up	400G	3d12h	13d
ai-backend	backend-stripe1-leaf3-ethernet-1-3	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf3-ethernet-1-31	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf3-ethernet-1-32	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf3-ethernet-1-4	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf4-ethernet-1-1	true	up	400G	3d11h	13d
ai-backend	backend-stripe1-leaf4-ethernet-1-2	true	up	400G	3d12h	13d
ai-backend	backend-stripe1-leaf4-ethernet-1-3	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf4-ethernet-1-31	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf4-ethernet-1-32	true	up	400G	13d	13d
ai-backend	backend-stripe1-leaf4-ethernet-1-4	true	up	400G	9h	4d12h
ai-backend	backend-stripe1-leaf5-ethernet-1-1	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf5-ethernet-1-2	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf5-ethernet-1-3	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf5-ethernet-1-31	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf5-ethernet-1-32	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf5-ethernet-1-4	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf6-ethernet-1-1	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf6-ethernet-1-2	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf6-ethernet-1-3	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf6-ethernet-1-31	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf6-ethernet-1-32	true	up	400G	5y265d	13d
ai-backend	backend-stripe1-leaf6-ethernet-1-4	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf7-ethernet-1-1	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf7-ethernet-1-2	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf7-ethernet-1-3	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf7-ethernet-1-31	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf7-ethernet-1-32	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf7-ethernet-1-4	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf8-ethernet-1-1	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf8-ethernet-1-2	true	up	400G	5y263d	13d
ai-backend	backend-stripe1-leaf8-ethernet-1-3	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe1-leaf8-ethernet-1-31	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf8-ethernet-1-32	true	up	400G	5y273d	13d
ai-backend	backend-stripe1-leaf8-ethernet-1-4	true	up	400G	5y260d	4d12h
ai-backend	backend-stripe2-leaf1-ethernet-1-1-1	true	up	400G	13d	13d
ai-backend	backend-stripe2-leaf1-ethernet-1-1-2	true	up	400G	13d	13d
ai-backend	backend-stripe2-leaf1-ethernet-1-2-1	true	up	400G	3d10h	4d12h
ai-backend	backend-stripe2-leaf1-ethernet-1-2-2	true	up	400G	10h	4d12h

```
ai-backend    backend-stripe2-leaf1-ethernet-1-31    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf1-ethernet-1-32    true      up                  800G    6d13h        13d
ai-backend    backend-stripe2-leaf2-ethernet-1-1-1   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf2-ethernet-1-1-2   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf2-ethernet-1-2-1   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf2-ethernet-1-2-2   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf2-ethernet-1-31    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf2-ethernet-1-32    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf3-ethernet-1-1-1   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf3-ethernet-1-1-2   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf3-ethernet-1-2-1   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf3-ethernet-1-2-2   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf3-ethernet-1-31    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf3-ethernet-1-32    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf4-ethernet-1-1-1   true      up                  400G    5y246d       13d
ai-backend    backend-stripe2-leaf4-ethernet-1-1-2   true      up                  400G    5y246d       13d
ai-backend    backend-stripe2-leaf4-ethernet-1-2-1   true      up                  400G    5y233d       4d12h
ai-backend    backend-stripe2-leaf4-ethernet-1-2-2   true      up                  400G    5y233d       4d12h
ai-backend    backend-stripe2-leaf4-ethernet-1-31    true      up                  800G    5y245d       13d
ai-backend    backend-stripe2-leaf4-ethernet-1-32    true      up                  800G    5y245d       13d
ai-backend    backend-stripe2-leaf5-ethernet-1-1-1   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf5-ethernet-1-1-2   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf5-ethernet-1-2-1   true      up                  400G    7h45m        4d12h
ai-backend    backend-stripe2-leaf5-ethernet-1-2-2   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf5-ethernet-1-31    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf5-ethernet-1-32    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf6-ethernet-1-1-1   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf6-ethernet-1-1-2   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf6-ethernet-1-2-1   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf6-ethernet-1-2-2   true      up                  400G    10h          4d12h
ai-backend    backend-stripe2-leaf6-ethernet-1-31    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf6-ethernet-1-32    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf7-ethernet-1-1-1   true      up                  400G    3d12h        13d
ai-backend    backend-stripe2-leaf7-ethernet-1-1-2   true      up                  400G    3d12h        13d
ai-backend    backend-stripe2-leaf7-ethernet-1-2-1   true      up                  400G    9h           4d12h
ai-backend    backend-stripe2-leaf7-ethernet-1-2-2   true      up                  400G    9h           4d12h
ai-backend    backend-stripe2-leaf7-ethernet-1-31    true      up                  800G    3d12h        13d
ai-backend    backend-stripe2-leaf7-ethernet-1-32    true      up                  800G    3d12h        13d
ai-backend    backend-stripe2-leaf8-ethernet-1-1-1   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf8-ethernet-1-1-2   true      up                  400G    13d          13d
ai-backend    backend-stripe2-leaf8-ethernet-1-2-1   true      up                  400G    9h           4d12h
ai-backend    backend-stripe2-leaf8-ethernet-1-2-2   true      up                  400G    9h           4d12h
ai-backend    backend-stripe2-leaf8-ethernet-1-31    true      up                  800G    12d          13d
ai-backend    backend-stripe2-leaf8-ethernet-1-32    true      up                  800G    12d          13d
```

11.2.6 Virtual networks validation

This section lists the VRFs/virtual networks across the data center network with their operational status and last change. The virtual networks are represented as Kubernetes objects of kind **`virtualnetworks`**.

```
kubectl get virtualnetworks -A
NAMESPACE     NAME       OPERATIONALSTATE   LASTCHANGE
ai-frontend   frontend   up                 2026-02-23T18:19:37.000Z
ai-frontend   storage    up                 2026-02-23T18:19:37.000Z
```

11.2.6.1 Detailed virtual network information

This validation provides details of EVI, MAC learning, associated VLAN, and the number of nodes associated with their status.

```
kubectl describe virtualnetworks storage -n ai-frontend
Name:         storage
Namespace:    ai-frontend
Labels:       <none>
Annotations:  <none>
API Version:  services.eda.nokia.com/v1
Kind:         VirtualNetwork
Metadata:
  Creation Timestamp:  2025-11-26T00:04:42Z
  Generation:          1
  Resource Version:    4259514
  UID:                 5dc45377-3296-4fb6-8016-1163826c1cc4
Spec:
  Bridge Domains:
    Name:  mac-vrf-storage
    Spec:
      Evi:       100
      Evi Pool:  evi-pool
      l2proxyARPND:
        Dynamic Learning:
          Age Time:      2000
          Enabled:       true
          Send Refresh:  2000
        Ip Duplication:
          Enabled:            true
          Hold Down Time:     10
          Monitoring Window:  10
          Num Moves:          4
        Proxy ARP:            true
        Proxy ND:             false
        Table Size:           250
      Mac Aging:              300
      Mac Learning:           true
      Tunnel Index Pool:      tunnel-index-pool
      Type:                   EVPNVXLAN
      Vni:                    100
      Vni Pool:               vni-pool
  Vlans:
    Name:  storage
    Spec:
      Bridge Domain:  mac-vrf-storage
      Interface Selector:
        eda.nokia.com/type=storage
      Vlan ID:  untagged
    Name:       gpu
    Spec:
      Bridge Domain:  mac-vrf-storage
      Interface Selector:
        eda.nokia.com/type=gpu
```

```
      Vlan ID:  100
Status:
  Last Change:  2025-12-04T19:08:32.000Z
  Nodes:
    frontend-leaf2
    frontend-leaf1
  Num BGP Peers:                    0
  Num BGP Peers Oper Down:          0
  Num IRB Interfaces:               0
  Num IRB Interfaces Oper Down:     0
  Num Nodes:                        2
  Num Routed Interfaces:            0
  Num Routed Interfaces Oper Down:  0
  Num Sub Interfaces:               32
  Num Sub Interfaces Oper Down:     6
  Operational State:                degraded
Events:                             <none>
```

11.3 Storage server validation

This section describes the commands used to validate the overall health, configuration, and operational status of the WEKA storage system, including cluster status, filesystem configuration, capacity usage, and alerts.

11.3.1 WEKA storage status

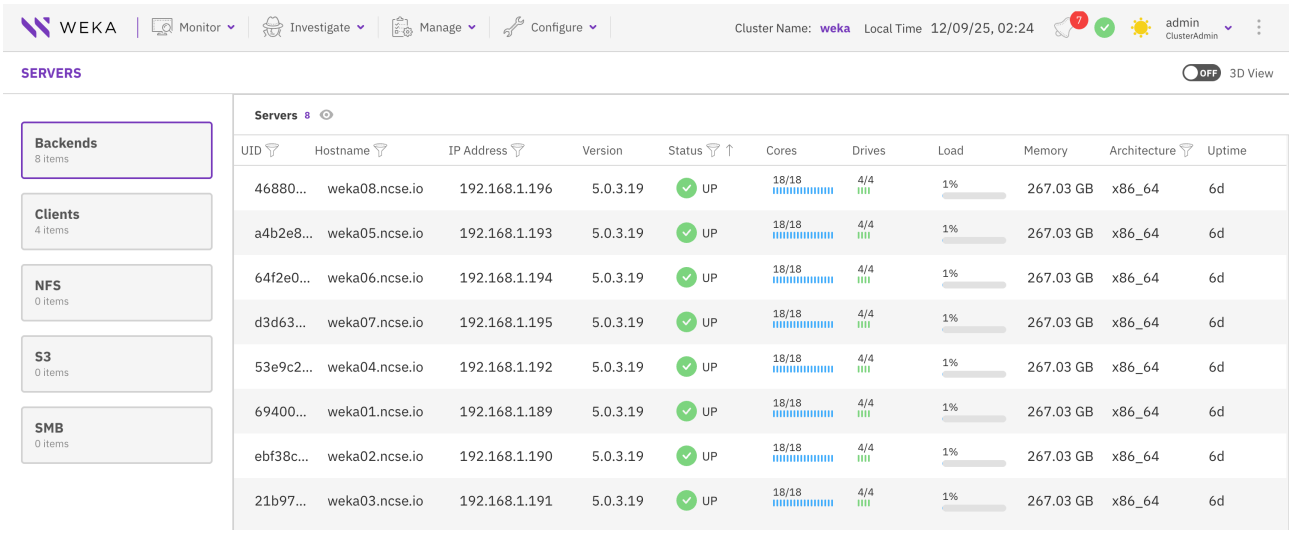
The **`weka status`** command provides a high-level health and configuration overview of a WEKA cluster.

```
[root@weka01 ~]# weka status
WekaIO v5.0.3.19 (CLI build 5.0.3.19)

       cluster: weka (5f6172d1-0ecf-4326-9d65-f02280b46b3b)
        status: OK (24 backend containers UP, 32 drives UP)
    protection: 5+2 (Fully protected)
     hot spare: 1 failure domain(s) (35.92 TiB)
 drive storage: 251.46 TiB total, 4.68 GiB unprovisioned
         cloud: disabled
       license: OK, valid thru 2028-04-28T11:27:59Z

     io status: STARTED 68 days ago (144 io-nodes UP, 2592 Buckets UP)
    link layer: Ethernet
       clients: 1 connected
         reads: 0 B/s (0 IO/s)
        writes: 0 B/s (0 IO/s)
    operations: 0 ops/s
        alerts: 2 active alerts, use `weka alerts` to list them
```

11.3.2 WEKA cluster status

```
[root@weka01 ~]# weka cluster servers list
HOST NAME       UID                                  PRIMARY IP     STATUS  UP SINCE                     CORES  RAM ALLOCATED  DRIVES  LOAD
ARCHITECTURE
slate4          00000000-0000-0000-0000-7cc255535831 192.168.1.175  UP  2025-12-02T21:11:49.581173Z      8    11794382848       0    1%  x86_64
weka01.ncse.io  69400a00-bd07-11ef-8000-905a0873cb7f 192.168.1.189  UP  2025-11-26T00:04:44.693813Z     18    267026169856      4    1%  x86_64
weka02.ncse.io  ebf38c6e-b6c0-4c9a-9d7f-d6277c0b4e3e 192.168.1.190  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
weka03.ncse.io  21b97e00-bd07-11ef-8000-905a0873ccc4 192.168.1.191  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
weka04.ncse.io  53e9c200-bcac-11ef-8000-905a0873cba8 192.168.1.192  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
weka05.ncse.io  a4b2e800-bcb7-11ef-8000-905a0873cbda 192.168.1.193  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
weka06.ncse.io  64f2e000-bb93-11ef-8000-905a0873cac7 192.168.1.194  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
weka07.ncse.io  d3d63600-bb9f-11ef-8000-905a0873cb7c 192.168.1.195  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
weka08.ncse.io  46880400-bcbb-11ef-8000-905a0873cc94 192.168.1.196  UP  2025-11-26T00:05:26.724514Z     18    267026169856      4    1%  x86_64
```

The WEKA UI can also be used to show each of storage nodes with their detailed usage. To view this information, navigate to: WEKA Home UI > Configure > Cluster Servers.



11.3.3 WEKA filesystem validation

The **`weka fs`** command shows the configured filesystems and their capacity usage. It lets you verify the filesystem name/ID, used and available SSD and total space, and whether thin provisioning is enabled.

```
[root@weka01 ~]# weka fs
FILESYSTEM ID  FILESYSTEM NAME  USED SSD  AVAILABLE SSD  USED TOTAL  AVAILABLE TOTAL  THIN PROVISIONED  THIN PROVISIONED MINIMUM SSD  THIN PROVISIONED
MAXIMUM SSD
0              fs1               6.38 TB      276.47 TB     6.38 TB        276.47 TB  False
```

11.3.4 WEKA cluster processes

This output displays the status of WEKA cluster processes running on each storage node, including container roles, IP addresses, and health state. It helps validate node reachability, process availability, and resource usage across the WEKA storage cluster.

```
[root@weka01 ~]# weka cluster processes
PROCESS ID  CONTAINER ID  SLOT IN HOST  HOSTNAME        CONTAINER  IPS            STATUS  RELEASE   ROLES       NETWORK     CPU  MEMORY    UPTIME
LAST FAILURE
0           0                         0  weka01.ncse.io  drives0    192.168.1.189  UP  5.0.3.19  MANAGEMENT  UDP             N/A       8:16:02h
Removed from cluster: Not reachable by the cluster (8 hours ago)
```

```
1          0                      1  weka01.ncse.io  drives0   192.168.1.189  UP  5.0.3.19  DRIVES      DPDK / GDS   12  1.56 GB   8:16:01h
Removed from cluster: Not reachable by the cluster (8 hours ago)
2          0                      2  weka01.ncse.io  drives0   192.168.1.189  UP  5.0.3.19  DRIVES      DPDK / GDS   29  1.56 GB   8:16:01h
Network port inactivity triggered node termination (8 hours ago)
3          0                      3  weka01.ncse.io  drives0   192.168.1.189  UP  5.0.3.19  DRIVES      DPDK / GDS   21  1.56 GB   8:15:56h
Network port inactivity triggered node termination (8 hours ago)
4          0                      4  weka01.ncse.io  drives0   192.168.1.189  UP  5.0.3.19  DRIVES      DPDK / GDS    2  1.56 GB   8:15:34h
Network port inactivity triggered node termination (8 hours ago)
20         1                      0  weka02.ncse.io  drives0   192.168.1.190  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:16:02h
Removed from cluster: Not reachable by the cluster (8 hours ago)
21         1                      1  weka02.ncse.io  drives0   192.168.1.190  UP  5.0.3.19  DRIVES      DPDK / GDS    1  1.56 GB   8:16:02h
Network port inactivity triggered node termination (8 hours ago)
22         1                      2  weka02.ncse.io  drives0   192.168.1.190  UP  5.0.3.19  DRIVES      DPDK / GDS   46  1.56 GB   8:15:28h
Network port inactivity triggered node termination (8 hours ago)
23         1                      3  weka02.ncse.io  drives0   192.168.1.190  UP  5.0.3.19  DRIVES      DPDK / GDS   37  1.56 GB   8:16:02h
Network port inactivity triggered node termination (8 hours ago)
24         1                      4  weka02.ncse.io  drives0   192.168.1.190  UP  5.0.3.19  DRIVES      DPDK / GDS   25  1.56 GB   8:15:34h
Network port inactivity triggered node termination (8 hours ago)
40         2                      0  weka03.ncse.io  drives0   192.168.1.191  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:16:02h
Network port inactivity triggered node termination (8 hours ago)
41         2                      1  weka03.ncse.io  drives0   192.168.1.191  UP  5.0.3.19  DRIVES      DPDK / GDS   16  1.56 GB   8:15:49h
Network port inactivity triggered node termination (8 hours ago)
42         2                      2  weka03.ncse.io  drives0   192.168.1.191  UP  5.0.3.19  DRIVES      DPDK / GDS   46  1.56 GB   8:15:45h
Network port inactivity triggered node termination (8 hours ago)
43         2                      3  weka03.ncse.io  drives0   192.168.1.191  UP  5.0.3.19  DRIVES      DPDK / GDS    5  1.56 GB   8:16:01h
Removed from cluster: Not reachable by the cluster (8 hours ago)
44         2                      4  weka03.ncse.io  drives0   192.168.1.191  UP  5.0.3.19  DRIVES      DPDK / GDS   11  1.56 GB   8:16:01h
Network port inactivity triggered node termination (8 hours ago)
60         3                      0  weka04.ncse.io  drives0   192.168.1.192  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:16:02h
Removed from cluster: Not reachable by the cluster (8 hours ago)
61         3                      1  weka04.ncse.io  drives0   192.168.1.192  UP  5.0.3.19  DRIVES      DPDK / GDS   46  1.56 GB   8:15:50h
Network port inactivity triggered node termination (8 hours ago)
62         3                      2  weka04.ncse.io  drives0   192.168.1.192  UP  5.0.3.19  DRIVES      DPDK / GDS   19  1.56 GB   8:15:56h
Network port inactivity triggered node termination (8 hours ago)
63         3                      3  weka04.ncse.io  drives0   192.168.1.192  UP  5.0.3.19  DRIVES      DPDK / GDS   40  1.56 GB   8:15:39h
Network port inactivity triggered node termination (8 hours ago)
64         3                      4  weka04.ncse.io  drives0   192.168.1.192  UP  5.0.3.19  DRIVES      DPDK / GDS    4  1.56 GB   8:15:49h
Network port inactivity triggered node termination (8 hours ago)
80         4                      0  weka05.ncse.io  drives0   192.168.1.193  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:15:56h
Removed from cluster: Not reachable by the cluster (8 hours ago)
81         4                      1  weka05.ncse.io  drives0   192.168.1.193  UP  5.0.3.19  DRIVES      DPDK / GDS    1  1.56 GB   8:16:02h
Removed from cluster: Not reachable by the cluster (8 hours ago)
82         4                      2  weka05.ncse.io  drives0   192.168.1.193  UP  5.0.3.19  DRIVES      DPDK / GDS   22  1.56 GB   8:16:01h
Removed from cluster: Not reachable by the cluster (8 hours ago)
83         4                      3  weka05.ncse.io  drives0   192.168.1.193  UP  5.0.3.19  DRIVES      DPDK / GDS   45  1.56 GB   8:15:18h
Network port inactivity triggered node termination (8 hours ago)
84         4                      4  weka05.ncse.io  drives0   192.168.1.193  UP  5.0.3.19  DRIVES      DPDK / GDS    9  1.56 GB   8:15:56h
Removed from cluster: Not reachable by the cluster (8 hours ago)
100        5                      0  weka06.ncse.io  drives0   192.168.1.194  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:15:56h
Removed from cluster: Not reachable by the cluster (8 hours ago)
101        5                      1  weka06.ncse.io  drives0   192.168.1.194  UP  5.0.3.19  DRIVES      DPDK / GDS   15  1.56 GB   8:15:50h
Network port inactivity triggered node termination (8 hours ago)
102        5                      2  weka06.ncse.io  drives0   192.168.1.194  UP  5.0.3.19  DRIVES      DPDK / GDS   36  1.56 GB   8:15:55h
Network port inactivity triggered node termination (8 hours ago)
103        5                      3  weka06.ncse.io  drives0   192.168.1.194  UP  5.0.3.19  DRIVES      DPDK / GDS   17  1.56 GB   8:15:56h
Removed from cluster: Not reachable by the cluster (8 hours ago)
104        5                      4  weka06.ncse.io  drives0   192.168.1.194  UP  5.0.3.19  DRIVES      DPDK / GDS    9  1.56 GB   8:15:56h
Removed from cluster: Not reachable by the cluster (8 hours ago)
120        6                      0  weka07.ncse.io  drives0   192.168.1.195  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:16:02h
Removed from cluster: Not reachable by the cluster (8 hours ago)
121        6                      1  weka07.ncse.io  drives0   192.168.1.195  UP  5.0.3.19  DRIVES      DPDK / GDS    2  1.56 GB   8:15:56h
Network port inactivity triggered node termination (8 hours ago)
122        6                      2  weka07.ncse.io  drives0   192.168.1.195  UP  5.0.3.19  DRIVES      DPDK / GDS   45  1.56 GB   8:15:50h
Network port inactivity triggered node termination (8 hours ago)
123        6                      3  weka07.ncse.io  drives0   192.168.1.195  UP  5.0.3.19  DRIVES      DPDK / GDS   26  1.56 GB   8:15:49h
Network port inactivity triggered node termination (8 hours ago)
124        6                      4  weka07.ncse.io  drives0   192.168.1.195  UP  5.0.3.19  DRIVES      DPDK / GDS    6  1.56 GB   8:16:01h
Network port inactivity triggered node termination (8 hours ago)
140        7                      0  weka08.ncse.io  drives0   192.168.1.196  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:16:54h
Removed from cluster: Not reachable by the cluster (7 days ago)
141        7                      1  weka08.ncse.io  drives0   192.168.1.196  UP  5.0.3.19  DRIVES      DPDK / GDS   45  1.56 GB   8:15:50h
Network port inactivity triggered node termination (8 hours ago)
142        7                      2  weka08.ncse.io  drives0   192.168.1.196  UP  5.0.3.19  DRIVES      DPDK / GDS   47  1.56 GB   8:16:02h
Removed from cluster: Not reachable by the cluster (8 hours ago)
143        7                      3  weka08.ncse.io  drives0   192.168.1.196  UP  5.0.3.19  DRIVES      DPDK / GDS   40  1.56 GB   8:15:29h
Network port inactivity triggered node termination (8 hours ago)
144        7                      4  weka08.ncse.io  drives0   192.168.1.196  UP  5.0.3.19  DRIVES      DPDK / GDS   27  1.56 GB   8:15:27h
Network port inactivity triggered node termination (8 hours ago)
160        8                      0  weka06.ncse.io  compute0  192.168.1.194  UP  5.0.3.19  MANAGEMENT  UDP              N/A       8:16:01h
Removed from cluster: Not reachable by the cluster (8 hours ago)
161        8                      1  weka06.ncse.io  compute0  192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   18  21.48 GB  8:15:44h
Network port inactivity triggered node termination (8 hours ago)
162        8                      2  weka06.ncse.io  compute0  192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   21  21.48 GB  8:15:43h
Network port inactivity triggered node termination (8 hours ago)
163        8                      3  weka06.ncse.io  compute0  192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   44  21.48 GB  8:15:39h
Network port inactivity triggered node termination (8 hours ago)
164        8                      4  weka06.ncse.io  compute0  192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   35  21.48 GB  8:16:01h
Removed from cluster: Not reachable by the cluster (8 hours ago)
165        8                      5  weka06.ncse.io  compute0  192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS    8  21.48 GB  8:15:38h
Network port inactivity triggered node termination (8 hours ago)
```

```
166         8                       6  weka06.ncse.io  compute0   192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   25  21.48 GB  8:15:23h
Network port inactivity triggered node termination (8 hours ago)
167         8                       7  weka06.ncse.io  compute0   192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   23  21.48 GB  8:15:45h
Network port inactivity triggered node termination (8 hours ago)
168         8                       8  weka06.ncse.io  compute0   192.168.1.194  UP  5.0.3.19  COMPUTE     DPDK / GDS   39  21.48 GB  8:15:44h
Network port inactivity triggered node termination (8 hours ago)

*snip*
```

11.4 GPU server validation

To validate the GPU server, perform the following steps.

1. Verify the CUDA version.

```
root@slate8:/home/nvidia# nvidia-smi
Wed Feb 25 20:28:57 2026
+-----------------------------------------------------------------------------------------
+
| NVIDIA-SMI 590.48.01              Driver Version: 590.48.01      CUDA Version: 13.1
|
+-----------------------------------------+------------------------+----------------------
+
```

2. Verify the topology of GPU topology with NVLink and interconnects.

```
nvidia@slate8:~$ nvidia-smi topo -m
      GPU0  GPU1  GPU2  GPU3  GPU4  GPU5  GPU6  GPU7  NIC0  NIC1  NIC2  NIC3  NIC4  NIC5  NIC6  NIC7  NIC8  CPU Affinity  NUMA Affinity  GPU NUMA ID
GPU0   X    NV18  NV18  NV18  NV18  NV18  NV18  NV18  PIX   NODE  NODE  NODE  SYS   SYS   SYS   SYS   SYS   0-95,192-287  0              N/A
GPU1  NV18   X    NV18  NV18  NV18  NV18  NV18  NV18  NODE  PIX   NODE  NODE  SYS   SYS   SYS   SYS   SYS   0-95,192-287  0              N/A
GPU2  NV18  NV18   X    NV18  NV18  NV18  NV18  NV18  NODE  NODE  PIX   NODE  SYS   SYS   SYS   SYS   SYS   0-95,192-287  0              N/A
GPU3  NV18  NV18  NV18   X    NV18  NV18  NV18  NV18  NODE  NODE  NODE  PIX   SYS   SYS   SYS   SYS   SYS   0-95,192-287  0              N/A
GPU4  NV18  NV18  NV18  NV18   X    NV18  NV18  NV18  SYS   SYS   SYS   SYS   PIX   NODE  NODE  NODE  NODE  96-191,288-383  1              N/A
GPU5  NV18  NV18  NV18  NV18  NV18   X    NV18  NV18  SYS   SYS   SYS   SYS   NODE  PIX   NODE  NODE  NODE  96-191,288-383  1              N/A
GPU6  NV18  NV18  NV18  NV18  NV18  NV18   X    NV18  SYS   SYS   SYS   SYS   NODE  NODE  PIX   NODE  PHB   96-191,288-383  1              N/A
GPU7  NV18  NV18  NV18  NV18  NV18  NV18  NV18   X    SYS   SYS   SYS   SYS   NODE  NODE  NODE  PIX   NODE  96-191,288-383  1              N/A
NIC0  PIX   NODE  NODE  NODE  SYS   SYS   SYS   SYS   X     NODE  NODE  NODE  SYS   SYS   SYS   SYS   SYS
NIC1  NODE  PIX   NODE  NODE  SYS   SYS   SYS   SYS   NODE  X     NODE  NODE  SYS   SYS   SYS   SYS   SYS
NIC2  NODE  NODE  PIX   NODE  SYS   SYS   SYS   SYS   NODE  NODE  X     NODE  SYS   SYS   SYS   SYS   SYS
NIC3  NODE  NODE  NODE  PIX   SYS   SYS   SYS   SYS   NODE  NODE  NODE  X     SYS   SYS   SYS   SYS   SYS
NIC4  SYS   SYS   SYS   SYS   PIX   NODE  NODE  NODE  SYS   SYS   SYS   SYS   X     NODE  NODE  NODE  NODE
NIC5  SYS   SYS   SYS   SYS   NODE  PIX   NODE  NODE  SYS   SYS   SYS   SYS   NODE  X     NODE  NODE  NODE
NIC6  SYS   SYS   SYS   SYS   NODE  NODE  PIX   NODE  SYS   SYS   SYS   SYS   NODE  NODE  X     NODE  PHB
NIC7  SYS   SYS   SYS   SYS   NODE  NODE  NODE  PIX   SYS   SYS   SYS   SYS   NODE  NODE  NODE  X     NODE
NIC8  SYS   SYS   SYS   SYS   NODE  NODE  PHB   NODE  SYS   SYS   SYS   SYS   NODE  NODE  PHB   NODE  X
```

3. Verify that the GPUs are being detected by CUDA.

```
root@slate8:/home/nvidia# nvidia-smi -L
GPU 0: NVIDIA H200 (UUID: GPU-80887775-37ce-a5f5-890b-d04c14b61a27)
GPU 1: NVIDIA H200 (UUID: GPU-cfd3ae55-0613-557a-581c-b4883e731ace)
GPU 2: NVIDIA H200 (UUID: GPU-bc41b234-9883-e061-dadb-745b1d4f93f0)
GPU 3: NVIDIA H200 (UUID: GPU-73693cb1-729d-1338-2362-98a6d5665bf4)
GPU 4: NVIDIA H200 (UUID: GPU-49c2104b-a4fd-5ba8-4d02-90d2e14651c2)
GPU 5: NVIDIA H200 (UUID: GPU-4f8547d7-d01c-1b91-1ed7-be46d42140cf)
GPU 6: NVIDIA H200 (UUID: GPU-171ce2eb-451b-3e0e-50fa-6c6db08ebc91)
GPU 7: NVIDIA H200 (UUID: GPU-562f23cb-9328-35ec-1f5f-a66da6a937e2)
```

4. Verify that all GPUs are operating within the expected range when the system is idle. When the system is idle, GPU and memory utilization should be 0%, clocks should be low, and temperature should be well under 85°C.

```
root@slate8:/home/nvidia# nvidia-smi dmon
# gpu    pwr  gtemp  mtemp     sm    mem    enc    dec    jpg    ofa   mclk   pclk
# Idx      W      C      C      %      %      %      %      %      %    MHz    MHz
    0     73     28     33      0      0      0      0      0      0   3201    345
    1     73     25     29      0      0      0      0      0      0   3201    345
    2     72     28     32      0      0      0      0      0      0   3201    345
    3     73     25     28      0      0      0      0      0      0   3201    345
    4     73     29     33      0      0      0      0      0      0   3201    345
    5     72     27     28      0      0      0      0      0      0   3201    345
    6     73     28     32      0      0      0      0      0      0   3201    345
    7     72     28     32      0      0      0      0      0      0   3201    345
```

11.4.1 GPU NICs

You can verify the GPU NIC's QoS settings using the **mlxfwmanager** tool. If no QoS settings appear, it indicates that the RDMA drivers were not installed correctly.

1. Verify that the Mellanox NICs are listed.

```
root@slate8:/home/nvidia# mlxfwmanager
Querying Mellanox devices firmware ...

Device #1:
----------

  Device Type:      BlueField3
  Part Number:      SN37B82788_Ax
  Description:      ThinkSystem NVIDIA BlueField-3 VPI QSFP112 1P 400G PCIe Gen5 x16
B3140H
  PSID:             LNV0000000071
  PCI Device Name:  0000:6a:00.0
  Base MAC:         b8e924b3ba2a
  Versions:         Current        Available
     FW             32.43.2026     N/A
     PXE            3.7.0500       N/A
     UEFI           14.36.0021     N/A
     UEFI Virtio blk   22.4.0014      N/A
     UEFI Virtio net   21.4.0013      N/A

  Status:           No matching image found

Device #2:
----------

  Device Type:      BlueField3
  Part Number:      SN37B82788_Ax
  Description:      ThinkSystem NVIDIA BlueField-3 VPI QSFP112 1P 400G PCIe Gen5 x16
B3140H
```

```
  PSID:              LNV0000000071
  PCI Device Name:   0000:84:00.0
  Base MAC:          b8e924b3b89e
  Versions:          Current        Available
     FW              32.43.2026     N/A
     PXE             3.7.0500       N/A
     UEFI            14.36.0021     N/A
     UEFI Virtio blk   22.4.0014      N/A
     UEFI Virtio net   21.4.0013      N/A

  Status:            No matching image found

~~ snip~~
```

2. Verify the NIC QoS settings.

```
root@slate8:/home/nvidia# mlnx_qos -i ens6f0np0
DCBX mode: OS controlled
Priority trust state: dscp
dscp2prio mapping:
   prio:0 dscp:07,06,05,04,03,02,01,00,
   prio:1 dscp:15,14,13,12,11,10,09,08,
   prio:2 dscp:23,22,21,20,19,18,17,16,
   prio:3 dscp:26,31,30,29,28,27,25,24,
   prio:4 dscp:39,38,37,36,35,34,33,32,
   prio:5 dscp:47,46,45,44,43,42,41,40,
   prio:6 dscp:48,55,54,53,52,51,50,49,
   prio:7 dscp:63,62,61,60,59,58,57,56,
Receive buffer size (bytes): 19872,588672,0,0,0,0,0,0,max_buffer_size=4151520
Cable len: 5
PFC configuration:
   priority    0   1   2   3   4   5   6   7
   enabled     0   0   0   1   0   0   0   0
   buffer      0   0   0   1   0   0   0   0
tc: 0 ratelimit: unlimited, tsa: ets, bw: 10%
    priority:  0
    priority:  1
    priority:  2
    priority:  4
    priority:  5
    priority:  7
tc: 3 ratelimit: unlimited, tsa: ets, bw: 90%
    priority:  3
tc: 6 ratelimit: unlimited, tsa: strict
    priority:  6
```

11.4.2 RoCEv2 statistics

Application layer statistics, particularly CNP and ECN packets, can be viewed as follows:

```
rdma statistic show link mlx5_0/1

COUNTER                         VALUE
------------------------------ ---------------
rx_write_requests               918214232
rx_read_requests                0
rx_atomic_requests              0
rx_dct_connect                  0
out_of_buffer                   0
out_of_sequence                 1
duplicate_request               0
rnr_nak_retry_err               0
packet_seq_err                  1679
implied_nak_seq_err             0
local_ack_timeout_err           0
resp_local_length_error         0
resp_cqe_error                  0
req_cqe_error                   0
req_remote_invalid_request      0
req_remote_access_errors        0
resp_remote_access_errors       0
resp_cqe_flush_error            0
req_cqe_flush_error             0
req_transport_retries_exceeded 0
req_rnr_retries_exceeded        0
roce_adp_retrans                198
roce_adp_retrans_to             0
roce_slow_restart               0
roce_slow_restart_cnps          198
roce_slow_restart_trans         0
rp_cnp_ignored                  0
rp_cnp_handled                  1570
np_ecn_marked_roce_packets      0
np_cnp_sent                     1
rx_icrc_encapsulated            0
```

11.4.3 Physical layer statistics

Physical layer statistics per NIC can be viewed using **`ethtool`**. For example, these statistics can be used to determine any PFCs received or generated by the NICs.

```
nvidia@slate5:~$ ethtool -S ens1f0np0 | grep prio3
     rx_prio3_bytes: 100497515161036
     rx_prio3_packets: 27358164976
     rx_prio3_discards: 0
     tx_prio3_bytes: 100078387433038
```

```
     tx_prio3_packets: 26385763603
     rx_prio3_pause: 354
     rx_prio3_pause_duration: 774
     tx_prio3_pause: 165438
     tx_prio3_pause_duration: 182143
     rx_prio3_pause_transition: 177
     rx_prio3_buf_discard: 0
     rx_prio3_cong_discard: 0
     rx_prio3_marked: 0
```

11.4.4 Server networking verification

Verify that the Netplan configuration has been applied and that all GPU interfaces have received their IP addresses with an MTU of 9000. Each interface's source address should correspond to its dedicated routing table (101–108).

```
nvidia@slate5:~$ ip addr show ens1f0np0
14: ens1f0np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP group default
qlen 1000
    link/ether b8:e9:24:b3:b9:d2 brd ff:ff:ff:ff:ff:ff
    altname enp164s0f0np0
    inet6 fe80::bae9:24ff:feb3:b9d2/64 scope link
       valid_lft forever preferred_lft forever
nvidia@slate5:~$ ip addr show ens1f0np0.1000
20: ens1f0np0.1000@ens1f0np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue
state UP group default qlen 1000
    link/ether b8:e9:24:b3:b9:d2 brd ff:ff:ff:ff:ff:ff
    inet6 fd00:2:9:1:2:1:0:2/96 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::bae9:24ff:feb3:b9d2/64 scope link
       valid_lft forever preferred_lft forever
nvidia@slate5:~$
nvidia@slate5:~$
nvidia@slate5:~$ ip -6 rule show
0:      from all lookup local
32758:  from fd00:2:13:1:2:1:0:2/96 lookup 105 proto static
32759:  from fd00:2:12:1:2:1:0:2/96 lookup 104 proto static
32760:  from fd00:2:16:1:2:1:0:2/96 lookup 108 proto static
32761:  from fd00:2:15:1:2:1:0:2/96 lookup 107 proto static
32762:  from fd00:2:10:1:2:1:0:2/96 lookup 102 proto static
32763:  from fd00:2:14:1:2:1:0:2/96 lookup 106 proto static
32764:  from fd00:2:9:1:2:1:0:2/96 lookup 101 proto static
32765:  from fd00:2:11:1:2:1:0:2/96 lookup 103 proto static
32766:  from all lookup main
nvidia@slate5:~$
nvidia@slate5:~$ ip -6 rule show table 101
32764:  from fd00:2:9:1:2:1:0:2/96 lookup 101 proto static
```

11.4.5 SLURM validation

Simple Linux Utility Resource Management (SLURM) is an open-source job scheduling tool/system serving as AI workload manager. It orchestrates resource management for AI/ML training and inference. Key functions of SLURM are: allocating GPU, CPU, and storage resources to users for a period to conduct their job, provisioning a means to start, schedule and monitor a job on a set of nodes, and scheduling resource access by queuing incoming requests.

Similar to any orchestrator, SLURM has a controller daemon called slurmctld which has a global view of the cluster, including nodes, storage partitions, job queues, and reservations. The controller makes the scheduling decisions. Each compute/GPU node runs a lightweight agent called slurmd that reports node health state information to the controller node. Users interact through client commands such as **`sbatch`** (to submit jobs) and **`srun`** (to run parallel jobs).

11.4.5.1 SLURM Cluster verification

The following example shows the overall cluster status.

```
nvidia@headnode:~$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute1*    up   infinite      4   idle slate[5-8]


nvidia@headnode:~$ sinfo -N -l
Wed Feb 25 22:50:51 2026
NODELIST   NODES PARTITION       STATE CPUS    S:C:T MEMORY TMP_DISK WEIGHT AVAIL_FE
REASON
slate5         1 compute1*        idle 384   384:1:1 232181        0      1   (null) none
slate6         1 compute1*        idle 384   384:1:1 232181        0      1   (null) none
slate7         1 compute1*        idle 384   384:1:1 232181        0      1   (null) none
slate8         1 compute1*        idle 384   384:1:1 232181        0      1   (null) none
```

The following example shows the SLURM node health check.

```
nvidia@headnode:~$ scontrol show nodes
NodeName=slate5 Arch=x86_64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=384 CPUTot=384 CPULoad=8.34
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:8
   NodeAddr=slate5 NodeHostName=slate5 Version=24.05.8
   OS=Linux 6.8.0-100-generic #100-Ubuntu SMP PREEMPT_DYNAMIC Tue Jan 13 16:40:06 UTC 2026
   RealMemory=2321819 AllocMem=0 FreeMem=2067612 Sockets=384 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=compute1
   BootTime=2026-02-12T13:56:58 SlurmdStartTime=2026-02-12T22:07:17
   LastBusyTime=2026-02-25T16:36:09 ResumeAfterTime=None
```

```
   CfgTRES=cpu=384,mem=2321819M,billing=384,gres/gpu=8
   AllocTRES=
   CurrentWatts=0 AveWatts=0


NodeName=slate6 Arch=x86_64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=384 CPUTot=384 CPULoad=10.20
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:8
   NodeAddr=slate6 NodeHostName=slate6 Version=24.05.8
   OS=Linux 6.8.0-100-generic #100-Ubuntu SMP PREEMPT_DYNAMIC Tue Jan 13 16:40:06 UTC 2026
   RealMemory=2321819 AllocMem=0 FreeMem=2069276 Sockets=384 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=compute1
   BootTime=2026-02-23T10:18:35 SlurmdStartTime=2026-02-23T10:19:39
   LastBusyTime=2026-02-25T16:36:09 ResumeAfterTime=None
   CfgTRES=cpu=384,mem=2321819M,billing=384,gres/gpu=8
   AllocTRES=
   CurrentWatts=0 AveWatts=0


NodeName=slate7 Arch=x86_64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=384 CPUTot=384 CPULoad=9.20
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:8
   NodeAddr=slate7 NodeHostName=slate7 Version=24.05.8
   OS=Linux 6.8.0-100-generic #100-Ubuntu SMP PREEMPT_DYNAMIC Tue Jan 13 16:40:06 UTC 2026
   RealMemory=2321819 AllocMem=0 FreeMem=2061554 Sockets=384 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=compute1
   BootTime=2026-02-12T14:08:12 SlurmdStartTime=2026-02-13T13:18:05
   LastBusyTime=2026-02-25T14:23:11 ResumeAfterTime=None
   CfgTRES=cpu=384,mem=2321819M,billing=384,gres/gpu=8
   AllocTRES=
   CurrentWatts=0 AveWatts=0


NodeName=slate8 Arch=x86_64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=384 CPUTot=384 CPULoad=10.35
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=gpu:8
   NodeAddr=slate8 NodeHostName=slate8 Version=24.05.8
   OS=Linux 6.8.0-100-generic #100-Ubuntu SMP PREEMPT_DYNAMIC Tue Jan 13 16:40:06 UTC 2026
   RealMemory=2321819 AllocMem=0 FreeMem=2044640 Sockets=384 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=compute1
   BootTime=2026-02-12T14:09:01 SlurmdStartTime=2026-02-12T22:09:03
   LastBusyTime=2026-02-25T08:37:05 ResumeAfterTime=None
   CfgTRES=cpu=384,mem=2321819M,billing=384,gres/gpu=8
   AllocTRES=
   CurrentWatts=0 AveWatts=0
```

12 Performance tests

12.1 NCCL operations test

NCCL tests, using mpirun, are used to measure collective communication performance within a single node (a single NVIDIA H200 server) and across multiple nodes (up to four NVIDIA H200 servers, in this design). These tests include different collective algorithms such as AllReduce, ReduceScatter, Broadcast, and All-to-All.

Note: The All-to-All collective is the most stressful collective algorithm because it includes sending traffic from every rank to every other rank in your GPU cluster (with *N* nodes, you have an *N* x (*N*-1) send/receive).

The All-to-All collective test ensures that it completes within reasonable times, indicating that there are no PFC storms or deadlocks and that ECN, PFC, and buffer thresholds are all configured effectively.

NCCL test				
Collective	**Accelerators (GPU)**	**Peak algbw (GB/s)**	**Peak busbw (GB/s)**	**Average busbw (GB/s)**
All-to-All	8	395.24	345.83	121.102
	16	97.15	91.07	33.0501
	24	67.89	64.47	24.4943
	32	59.67	57.81	21.0871
Broadcast	8	363.78	363.78	133.58
	16	329.91	329.91	91.911
	24	327.27	327.27	84.6267
	32	325.11	325.11	79.8153
ReduceScatter	8	416.81	364.71	125.602
	16	388.79	364.49	110.475
	24	374.89	359.27	101.206
	32	374.10	362.41	101.626
AllReduce	8	275.55	482.22	160.746
	16	259.98	487.46	141.652

	24	184.29	353.23	108.369
	32	184.35	357.18	102.32

12.1.1 AllReduce collective

```
slate5:~/nvidia-benchmarking-tests/nccl-tests/testing-scripts$ ./nccl-allreduce.sh
+ export OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ export UCX_DIR=/usr/bin
+ UCX_DIR=/usr/bin
+ export PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ echo /usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ /usr/mpi/gcc/openmpi-4.1.9a1/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x NCCL_DEBUG=VERSION -x PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/ -x LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib: -x NCCL_SOCKET_IFNAME=ens11f3 -x
NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3,mlx5_4,mlx5_5,mlx5_6,mlx5_9 -x
UCX_NET_DEVICES=ens11f3 -x HSA_DISABLE_CACHE=1 -x NCCL_PXN_DISABLE=0 -x NCCL_IB_SL=3 -x NCCL_IB_TC=104 --bind-to numa --mca pml
ucx --mca osc ucx --mca spml ucx --mca btl '^vader,openib' --mca btl_tcp_if_include ens11f3 /home/nvidia/nvidia-benchmarking-
tests/nccl-tests/build/all_reduce_perf -b 8 -e 16G -f 2 -i 0 -g 1
# nccl-tests version 2.17.8 nccl-headers=22902 nccl-library=22902
# Collective test starting: all_reduce_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 1 iters: 20 agg iters: 1 validation: 1 graph: 0
#
# Using devices
#  Rank  0 Group  0 Pid 1624470 on     slate5 device  0 [0000:03:00] NVIDIA H200
#  Rank  1 Group  0 Pid 1624471 on     slate5 device  1 [0000:23:00] NVIDIA H200
#  Rank  2 Group  0 Pid 1624472 on     slate5 device  2 [0000:43:00] NVIDIA H200
#  Rank  3 Group  0 Pid 1624473 on     slate5 device  3 [0000:63:00] NVIDIA H200
#  Rank  4 Group  0 Pid 1624474 on     slate5 device  4 [0000:83:00] NVIDIA H200
#  Rank  5 Group  0 Pid 1624475 on     slate5 device  5 [0000:a3:00] NVIDIA H200
#  Rank  6 Group  0 Pid 1624476 on     slate5 device  6 [0000:c3:00] NVIDIA H200
#  Rank  7 Group  0 Pid 1624480 on     slate5 device  7 [0000:e3:00] NVIDIA H200
#  Rank  8 Group  0 Pid 1713455 on     slate6 device  0 [0000:03:00] NVIDIA H200
#  Rank  9 Group  0 Pid 1713456 on     slate6 device  1 [0000:23:00] NVIDIA H200
#  Rank 10 Group  0 Pid 1713457 on     slate6 device  2 [0000:43:00] NVIDIA H200
#  Rank 11 Group  0 Pid 1713458 on     slate6 device  3 [0000:63:00] NVIDIA H200
#  Rank 12 Group  0 Pid 1713459 on     slate6 device  4 [0000:83:00] NVIDIA H200
#  Rank 13 Group  0 Pid 1713460 on     slate6 device  5 [0000:a3:00] NVIDIA H200
#  Rank 14 Group  0 Pid 1713462 on     slate6 device  6 [0000:c3:00] NVIDIA H200
#  Rank 15 Group  0 Pid 1713465 on     slate6 device  7 [0000:e3:00] NVIDIA H200
#  Rank 16 Group  0 Pid 1078570 on     slate7 device  0 [0000:03:00] NVIDIA H200
#  Rank 17 Group  0 Pid 1078571 on     slate7 device  1 [0000:23:00] NVIDIA H200
#  Rank 18 Group  0 Pid 1078572 on     slate7 device  2 [0000:43:00] NVIDIA H200
#  Rank 19 Group  0 Pid 1078573 on     slate7 device  3 [0000:63:00] NVIDIA H200
#  Rank 20 Group  0 Pid 1078574 on     slate7 device  4 [0000:83:00] NVIDIA H200
#  Rank 21 Group  0 Pid 1078575 on     slate7 device  5 [0000:a3:00] NVIDIA H200
#  Rank 22 Group  0 Pid 1078576 on     slate7 device  6 [0000:c3:00] NVIDIA H200
#  Rank 23 Group  0 Pid 1078578 on     slate7 device  7 [0000:e3:00] NVIDIA H200
#  Rank 24 Group  0 Pid 1087315 on     slate8 device  0 [0000:03:00] NVIDIA H200
#  Rank 25 Group  0 Pid 1087316 on     slate8 device  1 [0000:23:00] NVIDIA H200
#  Rank 26 Group  0 Pid 1087317 on     slate8 device  2 [0000:43:00] NVIDIA H200
#  Rank 27 Group  0 Pid 1087318 on     slate8 device  3 [0000:63:00] NVIDIA H200
#  Rank 28 Group  0 Pid 1087319 on     slate8 device  4 [0000:83:00] NVIDIA H200
#  Rank 29 Group  0 Pid 1087320 on     slate8 device  5 [0000:a3:00] NVIDIA H200
#  Rank 30 Group  0 Pid 1087322 on     slate8 device  6 [0000:c3:00] NVIDIA H200
#  Rank 31 Group  0 Pid 1087325 on     slate8 device  7 [0000:e3:00] NVIDIA H200
NCCL version 2.29.2+cuda13.1
#
#                                                              out-of-place                       in-place
#       size         count      type   redop    root     time   algbw   busbw  #wrong     time   algbw   busbw  #wrong
#        (B)    (elements)                               (us)  (GB/s)  (GB/s)            (us)  (GB/s)  (GB/s)
```

```
           8             2     float     sum      -1    48.77    0.00    0.00       0    46.99    0.00    0.00       0
          16             4     float     sum      -1    46.88    0.00    0.00       0    47.86    0.00    0.00       0
          32             8     float     sum      -1    47.85    0.00    0.00       0    47.33    0.00    0.00       0
          64            16     float     sum      -1    48.05    0.00    0.00       0    47.51    0.00    0.00       0
         128            32     float     sum      -1    48.52    0.00    0.01       0    49.64    0.00    0.00       0
         256            64     float     sum      -1    48.39    0.01    0.01       0    49.41    0.01    0.01       0
         512           128     float     sum      -1    49.05    0.01    0.02       0    49.65    0.01    0.02       0
        1024           256     float     sum      -1    52.55    0.02    0.04       0    52.19    0.02    0.04       0
        2048           512     float     sum      -1    55.75    0.04    0.07       0    54.58    0.04    0.07       0
        4096          1024     float     sum      -1    57.53    0.07    0.14       0    57.33    0.07    0.14       0
        8192          2048     float     sum      -1    59.95    0.14    0.26       0    57.70    0.14    0.28       0
       16384          4096     float     sum      -1    60.15    0.27    0.53       0    59.29    0.28    0.54       0
       32768          8192     float     sum      -1    61.51    0.53    1.03       0    59.84    0.55    1.06       0
       65536         16384     float     sum      -1    61.52    1.07    2.06       0    62.21    1.05    2.04       0
      131072         32768     float     sum      -1    68.91    1.90    3.69       0    66.63    1.97    3.81       0
      262144         65536     float     sum      -1    78.24    3.35    6.49       0    78.03    3.36    6.51       0
      524288        131072     float     sum      -1    88.46    5.93   11.48       0    88.97    5.89   11.42       0
     1048576        262144     float     sum      -1    96.93   10.82   20.96       0    96.99   10.81   20.95       0
     2097152        524288     float     sum      -1   119.17   17.60   34.10       0   121.06   17.32   33.56       0
     4194304       1048576     float     sum      -1   171.72   24.42   47.32       0   167.49   25.04   48.52       0
     8388608       2097152     float     sum      -1   236.59   35.46   68.70       0   220.87   37.98   73.59       0
    16777216       4194304     float     sum      -1   272.42   61.58  119.32       0   274.25   61.17  118.53       0
    33554432       8388608     float     sum      -1   386.49   86.82  168.21       0   408.68   82.10  159.08       0
    67108864      16777216     float     sum      -1   599.62  111.92  216.84       0   603.26  111.24  215.53       0
   134217728      33554432     float     sum      -1  1038.27  129.27  250.46       0  1033.56  129.86  251.60       0
   268435456      67108864     float     sum      -1  1835.10  146.28  283.41       0  1834.84  146.30  283.45       0
   536870912     134217728     float     sum      -1  3416.05  157.16  304.50       0  3415.42  157.19  304.56       0
  1073741824     268435456     float     sum      -1  6538.16  164.23  318.19       0  6673.41  160.90  311.74       0
  2147483648     536870912     float     sum      -1  11764.4  182.54  353.67       0  11785.1  182.22  353.05       0
  4294967296    1073741824     float     sum      -1  23406.4  183.50  355.52       0  23407.6  183.49  355.50       0
  8589934592    2147483648     float     sum      -1  46707.0  183.91  356.33       0  46706.8  183.91  356.33       0
 17179869184    4294967296     float     sum      -1  93190.8  184.35  357.18       0  93495.9  183.75  356.02       0
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 102.32
#
# Collective test concluded: all_reduce_perf
```

12.1.2 ReduceScatter collective

```
slate5:~/nvidia-benchmarking-tests/nccl-tests/testing-scripts$ ./nccl-reducescatter.sh
+ export OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ export UCX_DIR=/usr/bin
+ UCX_DIR=/usr/bin
+ export PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ echo /usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ /usr/mpi/gcc/openmpi-4.1.9a1/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x NCCL_DEBUG=VERSION -x PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/ -x LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib: -x NCCL_SOCKET_IFNAME=ens11f3 -x
NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3,mlx5_4,mlx5_5,mlx5_6,mlx5_9 -x
UCX_NET_DEVICES=ens11f3 -x HSA_DISABLE_CACHE=1 -x NCCL_PXN_DISABLE=0 -x NCCL_IB_SL=3 -x NCCL_IB_TC=104 --bind-to numa --mca pml
ucx --mca osc ucx --mca spml ucx --mca btl '^vader,openib' --mca btl_tcp_if_include ens11f3 /home/nvidia/nvidia-benchmarking-
tests/nccl-tests/build/reduce_scatter_perf -b 8 -e 16G -f 2 -i 0 -g 1
# nccl-tests version 2.17.8 nccl-headers=22902 nccl-library=22902
# Collective test starting: reduce_scatter_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 1 iters: 20 agg iters: 1 validation: 1 graph: 0
#
# Using devices
#  Rank  0 Group  0 Pid 1620228 on     slate5 device  0 [0000:03:00] NVIDIA H200
#  Rank  1 Group  0 Pid 1620229 on     slate5 device  1 [0000:23:00] NVIDIA H200
#  Rank  2 Group  0 Pid 1620230 on     slate5 device  2 [0000:43:00] NVIDIA H200
#  Rank  3 Group  0 Pid 1620231 on     slate5 device  3 [0000:63:00] NVIDIA H200
#  Rank  4 Group  0 Pid 1620232 on     slate5 device  4 [0000:83:00] NVIDIA H200
#  Rank  5 Group  0 Pid 1620233 on     slate5 device  5 [0000:a3:00] NVIDIA H200
#  Rank  6 Group  0 Pid 1620234 on     slate5 device  6 [0000:c3:00] NVIDIA H200
#  Rank  7 Group  0 Pid 1620237 on     slate5 device  7 [0000:e3:00] NVIDIA H200
```

```
#  Rank  8 Group  0 Pid 1707952 on     slate6 device  0 [0000:03:00] NVIDIA H200
#  Rank  9 Group  0 Pid 1707953 on     slate6 device  1 [0000:23:00] NVIDIA H200
#  Rank 10 Group  0 Pid 1707954 on     slate6 device  2 [0000:43:00] NVIDIA H200
#  Rank 11 Group  0 Pid 1707955 on     slate6 device  3 [0000:63:00] NVIDIA H200
#  Rank 12 Group  0 Pid 1707956 on     slate6 device  4 [0000:83:00] NVIDIA H200
#  Rank 13 Group  0 Pid 1707957 on     slate6 device  5 [0000:a3:00] NVIDIA H200
#  Rank 14 Group  0 Pid 1707959 on     slate6 device  6 [0000:c3:00] NVIDIA H200
#  Rank 15 Group  0 Pid 1707960 on     slate6 device  7 [0000:e3:00] NVIDIA H200
#  Rank 16 Group  0 Pid 1075417 on     slate7 device  0 [0000:03:00] NVIDIA H200
#  Rank 17 Group  0 Pid 1075418 on     slate7 device  1 [0000:23:00] NVIDIA H200
#  Rank 18 Group  0 Pid 1075419 on     slate7 device  2 [0000:43:00] NVIDIA H200
#  Rank 19 Group  0 Pid 1075420 on     slate7 device  3 [0000:63:00] NVIDIA H200
#  Rank 20 Group  0 Pid 1075421 on     slate7 device  4 [0000:83:00] NVIDIA H200
#  Rank 21 Group  0 Pid 1075422 on     slate7 device  5 [0000:a3:00] NVIDIA H200
#  Rank 22 Group  0 Pid 1075423 on     slate7 device  6 [0000:c3:00] NVIDIA H200
#  Rank 23 Group  0 Pid 1075425 on     slate7 device  7 [0000:e3:00] NVIDIA H200
#  Rank 24 Group  0 Pid 1084980 on     slate8 device  0 [0000:03:00] NVIDIA H200
#  Rank 25 Group  0 Pid 1084981 on     slate8 device  1 [0000:23:00] NVIDIA H200
#  Rank 26 Group  0 Pid 1084982 on     slate8 device  2 [0000:43:00] NVIDIA H200
#  Rank 27 Group  0 Pid 1084983 on     slate8 device  3 [0000:63:00] NVIDIA H200
#  Rank 28 Group  0 Pid 1084984 on     slate8 device  4 [0000:83:00] NVIDIA H200
#  Rank 29 Group  0 Pid 1084985 on     slate8 device  5 [0000:a3:00] NVIDIA H200
#  Rank 30 Group  0 Pid 1084986 on     slate8 device  6 [0000:c3:00] NVIDIA H200
#  Rank 31 Group  0 Pid 1084989 on     slate8 device  7 [0000:e3:00] NVIDIA H200
NCCL version 2.29.2+cuda13.1
#
#                                                              out-of-place                       in-place
#       size         count      type   redop    root     time   algbw   busbw  #wrong     time   algbw   busbw  #wrong
#        (B)    (elements)                               (us)  (GB/s)  (GB/s)             (us)  (GB/s)  (GB/s)
           0             0     float     sum      -1     0.28    0.00    0.00       0     0.21    0.00    0.00       0
           0             0     float     sum      -1     0.21    0.00    0.00       0     0.19    0.00    0.00       0
           0             0     float     sum      -1     0.29    0.00    0.00       0     0.46    0.00    0.00       0
           0             0     float     sum      -1     0.33    0.00    0.00       0     0.20    0.00    0.00       0
           0             0     float     sum      -1     0.20    0.00    0.00       0     0.19    0.00    0.00       0
           0             0     float     sum      -1     0.19    0.00    0.00       0     0.20    0.00    0.00       0
         512             4     float     sum      -1    78.79    0.01    0.01       0    76.10    0.01    0.01       0
        1024             8     float     sum      -1    76.42    0.01    0.01       0    76.66    0.01    0.01       0
        2048            16     float     sum      -1    77.49    0.03    0.03       0    77.32    0.03    0.03       0
        4096            32     float     sum      -1    77.46    0.05    0.05       0    78.41    0.05    0.05       0
        8192            64     float     sum      -1    78.62    0.10    0.10       0    79.09    0.10    0.10       0
       16384           128     float     sum      -1    79.56    0.21    0.20       0    79.36    0.21    0.20       0
       32768           256     float     sum      -1    80.62    0.41    0.39       0    79.65    0.41    0.40       0
       65536           512     float     sum      -1    84.55    0.78    0.75       0    84.04    0.78    0.76       0
      131072          1024     float     sum      -1    89.93    1.46    1.41       0    89.47    1.46    1.42       0
      262144          2048     float     sum      -1   110.23    2.38    2.30       0   108.30    2.42    2.34       0
      524288          4096     float     sum      -1   128.29    4.09    3.96       0   127.37    4.12    3.99       0
     1048576          8192     float     sum      -1   127.59    8.22    7.96       0   128.35    8.17    7.91       0
     2097152         16384     float     sum      -1   127.70   16.42   15.91       0   128.44   16.33   15.82       0
     4194304         32768     float     sum      -1   129.28   32.44   31.43       0   134.31   31.23   30.25       0
     8388608         65536     float     sum      -1   135.27   62.02   60.08       0   137.98   60.80   58.90       0
    16777216        131072     float     sum      -1   159.89  104.93  101.65       0   158.53  105.83  102.52       0
    33554432        262144     float     sum      -1   208.80  160.70  155.68       0   207.79  161.49  156.44       0
    67108864        524288     float     sum      -1   300.34  223.44  216.46       0   309.41  216.89  210.11       0
   134217728       1048576     float     sum      -1   461.08  291.09  282.00       0   461.56  290.79  281.70       0
   268435456       2097152     float     sum      -1   967.21  277.53  268.86       0   964.97  278.18  269.49       0
   536870912       4194304     float     sum      -1  1673.56  320.80  310.77       0  1667.65  321.93  311.87       0
  1073741824       8388608     float     sum      -1  2947.31  364.31  352.93       0  2945.13  364.58  353.19       0
  2147483648      16777216     float     sum      -1  5815.93  369.24  357.70       0  5812.15  369.48  357.94       0
  4294967296      33554432     float     sum      -1  11547.9  371.93  360.30       0  11554.0  371.73  360.11       0
  8589934592      67108864     float     sum      -1  23053.8  372.60  360.96       0  23034.5  372.92  361.26       0
 17179869184     134217728     float     sum      -1  45923.6  374.10  362.41       0  45853.8  374.67  362.96       0
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 101.626
#
# Collective test concluded: reduce_scatter_perf
#
```

12.1.3 Broadcast collective

```
slate5:~/nvidia-benchmarking-tests/nccl-tests/testing-scripts$ ./nccl-broadcast.sh
+ export OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ export UCX_DIR=/usr/bin
+ UCX_DIR=/usr/bin
```

```
+ export PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ echo /usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ /usr/mpi/gcc/openmpi-4.1.9a1/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x NCCL_DEBUG=VERSION -x PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/ -x LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib: -x NCCL_SOCKET_IFNAME=ens11f3 -x
NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3,mlx5_4,mlx5_5,mlx5_6,mlx5_9 -x
UCX_NET_DEVICES=ens11f3 -x HSA_DISABLE_CACHE=1 -x NCCL_PXN_DISABLE=0 -x NCCL_IB_SL=3 -x NCCL_IB_TC=104 --bind-to numa --mca pml
ucx --mca osc ucx --mca spml ucx --mca btl '^vader,openib' --mca btl_tcp_if_include ens11f3 /home/nvidia/nvidia-benchmarking-
tests/nccl-tests/build/broadcast_perf -b 8 -e 16G -f 2 -i 0 -g 1
# nccl-tests version 2.17.8 nccl-headers=22902 nccl-library=22902
# Collective test starting: broadcast_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 1 iters: 20 agg iters: 1 validation: 1 graph: 0
#
# Using devices
#  Rank  0 Group  0 Pid 1616207 on     slate5 device  0 [0000:03:00] NVIDIA H200
#  Rank  1 Group  0 Pid 1616208 on     slate5 device  1 [0000:23:00] NVIDIA H200
#  Rank  2 Group  0 Pid 1616209 on     slate5 device  2 [0000:43:00] NVIDIA H200
#  Rank  3 Group  0 Pid 1616210 on     slate5 device  3 [0000:63:00] NVIDIA H200
#  Rank  4 Group  0 Pid 1616211 on     slate5 device  4 [0000:83:00] NVIDIA H200
#  Rank  5 Group  0 Pid 1616212 on     slate5 device  5 [0000:a3:00] NVIDIA H200
#  Rank  6 Group  0 Pid 1616213 on     slate5 device  6 [0000:c3:00] NVIDIA H200
#  Rank  7 Group  0 Pid 1616216 on     slate5 device  7 [0000:e3:00] NVIDIA H200
#  Rank  8 Group  0 Pid 1700695 on     slate6 device  0 [0000:03:00] NVIDIA H200
#  Rank  9 Group  0 Pid 1700696 on     slate6 device  1 [0000:23:00] NVIDIA H200
#  Rank 10 Group  0 Pid 1700697 on     slate6 device  2 [0000:43:00] NVIDIA H200
#  Rank 11 Group  0 Pid 1700698 on     slate6 device  3 [0000:63:00] NVIDIA H200
#  Rank 12 Group  0 Pid 1700699 on     slate6 device  4 [0000:83:00] NVIDIA H200
#  Rank 13 Group  0 Pid 1700700 on     slate6 device  5 [0000:a3:00] NVIDIA H200
#  Rank 14 Group  0 Pid 1700701 on     slate6 device  6 [0000:c3:00] NVIDIA H200
#  Rank 15 Group  0 Pid 1700704 on     slate6 device  7 [0000:e3:00] NVIDIA H200
#  Rank 16 Group  0 Pid 1071303 on     slate7 device  0 [0000:03:00] NVIDIA H200
#  Rank 17 Group  0 Pid 1071304 on     slate7 device  1 [0000:23:00] NVIDIA H200
#  Rank 18 Group  0 Pid 1071305 on     slate7 device  2 [0000:43:00] NVIDIA H200
#  Rank 19 Group  0 Pid 1071306 on     slate7 device  3 [0000:63:00] NVIDIA H200
#  Rank 20 Group  0 Pid 1071307 on     slate7 device  4 [0000:83:00] NVIDIA H200
#  Rank 21 Group  0 Pid 1071308 on     slate7 device  5 [0000:a3:00] NVIDIA H200
#  Rank 22 Group  0 Pid 1071310 on     slate7 device  6 [0000:c3:00] NVIDIA H200
#  Rank 23 Group  0 Pid 1071313 on     slate7 device  7 [0000:e3:00] NVIDIA H200
#  Rank 24 Group  0 Pid 1080889 on     slate8 device  0 [0000:03:00] NVIDIA H200
#  Rank 25 Group  0 Pid 1080890 on     slate8 device  1 [0000:23:00] NVIDIA H200
#  Rank 26 Group  0 Pid 1080891 on     slate8 device  2 [0000:43:00] NVIDIA H200
#  Rank 27 Group  0 Pid 1080892 on     slate8 device  3 [0000:63:00] NVIDIA H200
#  Rank 28 Group  0 Pid 1080893 on     slate8 device  4 [0000:83:00] NVIDIA H200
#  Rank 29 Group  0 Pid 1080894 on     slate8 device  5 [0000:a3:00] NVIDIA H200
#  Rank 30 Group  0 Pid 1080895 on     slate8 device  6 [0000:c3:00] NVIDIA H200
#  Rank 31 Group  0 Pid 1080897 on     slate8 device  7 [0000:e3:00] NVIDIA H200
NCCL version 2.29.2+cuda13.1
#
#                                                              out-of-place                       in-place
#       size         count      type   redop    root     time   algbw   busbw  #wrong     time   algbw   busbw  #wrong
#        (B)    (elements)                               (us)  (GB/s)  (GB/s)             (us)  (GB/s)  (GB/s)
           8             2     float    none       0    16.42    0.00    0.00       0    11.76    0.00    0.00       0
          16             4     float    none       0    11.86    0.00    0.00       0    11.72    0.00    0.00       0
          32             8     float    none       0    11.74    0.00    0.00       0    12.05    0.00    0.00       0
          64            16     float    none       0    12.34    0.01    0.01       0    11.91    0.01    0.01       0
         128            32     float    none       0    12.07    0.01    0.01       0    12.01    0.01    0.01       0
         256            64     float    none       0    11.86    0.02    0.02       0    11.88    0.02    0.02       0
         512           128     float    none       0    12.29    0.04    0.04       0    12.06    0.04    0.04       0
        1024           256     float    none       0    12.13    0.08    0.08       0    12.40    0.08    0.08       0
        2048           512     float    none       0    12.88    0.16    0.16       0    13.16    0.16    0.16       0
        4096          1024     float    none       0    13.53    0.30    0.30       0    13.79    0.30    0.30       0
        8192          2048     float    none       0    15.63    0.52    0.52       0    17.12    0.48    0.48       0
       16384          4096     float    none       0    21.21    0.77    0.77       0    22.07    0.74    0.74       0
       32768          8192     float    none       0    27.45    1.19    1.19       0    26.85    1.22    1.22       0
       65536         16384     float    none       0    30.73    2.13    2.13       0    30.80    2.13    2.13       0
      131072         32768     float    none       0    38.15    3.44    3.44       0    36.58    3.58    3.58       0
      262144         65536     float    none       0    80.07    3.27    3.27       0    80.20    3.27    3.27       0
      524288        131072     float    none       0   115.67    4.53    4.53       0   110.90    4.73    4.73       0
     1048576        262144     float    none       0   104.88   10.00   10.00       0   105.17    9.97    9.97       0
```

```
     2097152        524288     float    none       0   125.09   16.76   16.76       0   123.44   16.99   16.99       0
     4194304       1048576     float    none       0   158.11   26.53   26.53       0   158.15   26.52   26.52       0
     8388608       2097152     float    none       0   229.24   36.59   36.59       0   228.17   36.77   36.77       0
    16777216       4194304     float    none       0   273.62   61.32   61.32       0   274.06   61.22   61.22       0
    33554432       8388608     float    none       0   336.85   99.61   99.61       0   336.63   99.68   99.68       0
    67108864      16777216     float    none       0   456.55  146.99  146.99       0   456.83  146.90  146.90       0
   134217728      33554432     float    none       0   694.80  193.17  193.17       0   693.50  193.54  193.54       0
   268435456      67108864     float    none       0  1161.90  231.03  231.03       0  1161.78  231.05  231.05       0
   536870912     134217728     float    none       0  2530.80  212.14  212.14       0  2527.57  212.41  212.41       0
  1073741824     268435456     float    none       0  4159.99  258.11  258.11       0  4162.91  257.93  257.93       0
  2147483648     536870912     float    none       0  7410.68  289.78  289.78       0  7417.33  289.52  289.52       0
  4294967296    1073741824     float    none       0  13855.0  309.99  309.99       0  13881.5  309.40  309.40       0
  8589934592    2147483648     float    none       0  26822.9  320.25  320.25       0  26854.2  319.87  319.87       0
 17179869184    4294967296     float    none       0  52842.8  325.11  325.11       0  52738.2  325.76  325.76       0
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 79.8153
#
# Collective test concluded: broadcast_perf
#
```

12.1.4 All-to-All collective

```
slate5:~/nvidia-benchmarking-tests/nccl-tests/testing-scripts$ ./nccl-alltoall.sh
+ export OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ OMPI_DIR=/usr/mpi/gcc/openmpi-4.1.9a1
+ export UCX_DIR=/usr/bin
+ UCX_DIR=/usr/bin
+ export PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/
+ export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ echo /usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib:
+ /usr/mpi/gcc/openmpi-4.1.9a1/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x NCCL_DEBUG=VERSION -x PATH=/usr/mpi/gcc/openmpi-
4.1.9a1/bin:/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/mell
anox/doca/tools/ -x LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.1.9a1/lib:/lib:/usr/lib: -x NCCL_SOCKET_IFNAME=ens11f3 -x
NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3,mlx5_4,mlx5_5,mlx5_6,mlx5_9 -x
UCX_NET_DEVICES=ens11f3 -x HSA_DISABLE_CACHE=1 -x NCCL_PXN_DISABLE=0 -x NCCL_IB_SL=3 -x NCCL_IB_TC=104 --bind-to numa --mca pml
ucx --mca osc ucx --mca spml ucx --mca btl '^vader,openib' --mca btl_tcp_if_include ens11f3 /home/nvidia/nvidia-benchmarking-
tests/nccl-tests/build/alltoall_perf -b 8 -e 16G -f 2 -i 0 -g 1
# nccl-tests version 2.17.8 nccl-headers=22902 nccl-library=22902
# Collective test starting: alltoall_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 1 iters: 20 agg iters: 1 validation: 1 graph: 0
#
# Using devices
#  Rank  0 Group  0 Pid 1610013 on     slate5 device  0 [0000:03:00] NVIDIA H200
#  Rank  1 Group  0 Pid 1610014 on     slate5 device  1 [0000:23:00] NVIDIA H200
#  Rank  2 Group  0 Pid 1610015 on     slate5 device  2 [0000:43:00] NVIDIA H200
#  Rank  3 Group  0 Pid 1610016 on     slate5 device  3 [0000:63:00] NVIDIA H200
#  Rank  4 Group  0 Pid 1610017 on     slate5 device  4 [0000:83:00] NVIDIA H200
#  Rank  5 Group  0 Pid 1610018 on     slate5 device  5 [0000:a3:00] NVIDIA H200
#  Rank  6 Group  0 Pid 1610019 on     slate5 device  6 [0000:c3:00] NVIDIA H200
#  Rank  7 Group  0 Pid 1610020 on     slate5 device  7 [0000:e3:00] NVIDIA H200
#  Rank  8 Group  0 Pid 1691554 on     slate6 device  0 [0000:03:00] NVIDIA H200
#  Rank  9 Group  0 Pid 1691555 on     slate6 device  1 [0000:23:00] NVIDIA H200
#  Rank 10 Group  0 Pid 1691556 on     slate6 device  2 [0000:43:00] NVIDIA H200
#  Rank 11 Group  0 Pid 1691557 on     slate6 device  3 [0000:63:00] NVIDIA H200
#  Rank 12 Group  0 Pid 1691558 on     slate6 device  4 [0000:83:00] NVIDIA H200
#  Rank 13 Group  0 Pid 1691559 on     slate6 device  5 [0000:a3:00] NVIDIA H200
#  Rank 14 Group  0 Pid 1691560 on     slate6 device  6 [0000:c3:00] NVIDIA H200
#  Rank 15 Group  0 Pid 1691561 on     slate6 device  7 [0000:e3:00] NVIDIA H200
#  Rank 16 Group  0 Pid 1066154 on     slate7 device  0 [0000:03:00] NVIDIA H200
#  Rank 17 Group  0 Pid 1066155 on     slate7 device  1 [0000:23:00] NVIDIA H200
#  Rank 18 Group  0 Pid 1066156 on     slate7 device  2 [0000:43:00] NVIDIA H200
#  Rank 19 Group  0 Pid 1066157 on     slate7 device  3 [0000:63:00] NVIDIA H200
#  Rank 20 Group  0 Pid 1066158 on     slate7 device  4 [0000:83:00] NVIDIA H200
#  Rank 21 Group  0 Pid 1066159 on     slate7 device  5 [0000:a3:00] NVIDIA H200
#  Rank 22 Group  0 Pid 1066160 on     slate7 device  6 [0000:c3:00] NVIDIA H200
#  Rank 23 Group  0 Pid 1066161 on     slate7 device  7 [0000:e3:00] NVIDIA H200
#  Rank 24 Group  0 Pid 1076494 on     slate8 device  0 [0000:03:00] NVIDIA H200
```

```
#  Rank 25 Group  0 Pid 1076495 on     slate8 device  1 [0000:23:00] NVIDIA H200
#  Rank 26 Group  0 Pid 1076496 on     slate8 device  2 [0000:43:00] NVIDIA H200
#  Rank 27 Group  0 Pid 1076497 on     slate8 device  3 [0000:63:00] NVIDIA H200
#  Rank 28 Group  0 Pid 1076498 on     slate8 device  4 [0000:83:00] NVIDIA H200
#  Rank 29 Group  0 Pid 1076499 on     slate8 device  5 [0000:a3:00] NVIDIA H200
#  Rank 30 Group  0 Pid 1076500 on     slate8 device  6 [0000:c3:00] NVIDIA H200
#  Rank 31 Group  0 Pid 1076501 on     slate8 device  7 [0000:e3:00] NVIDIA H200
NCCL version 2.29.2+cuda13.1
#
#                                                              out-of-place                       in-place
#       size         count      type   redop    root     time   algbw   busbw  #wrong     time   algbw   busbw  #wrong
#        (B)    (elements)                               (us)  (GB/s)  (GB/s)             (us)  (GB/s)  (GB/s)
           0             0     float    none      -1     0.44    0.00    0.00       0     0.23    0.00    0.00    N/A
           0             0     float    none      -1     0.23    0.00    0.00       0     0.23    0.00    0.00    N/A
           0             0     float    none      -1     0.22    0.00    0.00       0     0.25    0.00    0.00    N/A
           0             0     float    none      -1     0.22    0.00    0.00       0     0.22    0.00    0.00    N/A
           0             0     float    none      -1     0.22    0.00    0.00       0     0.22    0.00    0.00    N/A
           0             0     float    none      -1     0.21    0.00    0.00       0     0.21    0.00    0.00    N/A
         512             4     float    none      -1    47.26    0.01    0.01       0    45.68    0.01    0.01    N/A
        1024             8     float    none      -1    47.11    0.02    0.02       0    45.44    0.02    0.02    N/A
        2048            16     float    none      -1    43.70    0.05    0.05       0    55.60    0.04    0.04    N/A
        4096            32     float    none      -1    43.56    0.09    0.09       0    42.83    0.10    0.09    N/A
        8192            64     float    none      -1    43.52    0.19    0.18       0    44.82    0.18    0.18    N/A
       16384           128     float    none      -1    46.16    0.35    0.34       0    42.06    0.39    0.38    N/A
       32768           256     float    none      -1    46.83    0.70    0.68       0    43.37    0.76    0.73    N/A
       65536           512     float    none      -1    44.77    1.46    1.42       0    44.60    1.47    1.42    N/A
      131072          1024     float    none      -1    45.79    2.86    2.77       0    45.65    2.87    2.78    N/A
      262144          2048     float    none      -1    50.81    5.16    5.00       0    49.24    5.32    5.16    N/A
      524288          4096     float    none      -1    61.72    8.49    8.23       0    58.55    8.96    8.68    N/A
     1048576          8192     float    none      -1    74.03   14.16   13.72       0    75.62   13.87   13.43    N/A
     2097152         16384     float    none      -1   102.23   20.52   19.87       0   101.65   20.63   19.99    N/A
     4194304         32768     float    none      -1   153.39   27.34   26.49       0   157.44   26.64   25.81    N/A
     8388608         65536     float    none      -1   230.28   36.43   35.29       0   264.31   31.74   30.75    N/A
    16777216        131072     float    none      -1   442.27   37.93   36.75       0   433.86   38.67   37.46    N/A
    33554432        262144     float    none      -1   852.82   39.35   38.12       0   862.34   38.91   37.69    N/A
    67108864        524288     float    none      -1  1381.81   48.57   47.05       0  1223.78   54.84   53.12    N/A
   134217728       1048576     float    none      -1  2598.25   51.66   50.04       0  2468.91   54.36   52.66    N/A
   268435456       2097152     float    none      -1  4822.49   55.66   53.92       0  4922.24   54.54   52.83    N/A
   536870912       4194304     float    none      -1  9599.41   55.93   54.18       0  9745.13   55.09   53.37    N/A
  1073741824       8388608     float    none      -1  18901.1   56.81   55.03       0  19281.1   55.69   53.95    N/A
  2147483648      16777216     float    none      -1  38234.8   56.17   54.41       0  38373.2   55.96   54.21    N/A
  4294967296      33554432     float    none      -1  74359.9   57.76   55.95       0  74323.1   57.79   55.98    N/A
  8589934592      67108864     float    none      -1   146099   58.80   56.96       0   146049   58.82   56.98    N/A
 17179869184     134217728     float    none      -1   287912   59.67   57.81       0   289633   59.32   57.46    N/A
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 21.0871
#
# Collective test concluded: alltoall_perf
```

12.2 Mellanox ConnectX-7/BlueField-3 NIC throughput

ROCEv2 perf test	
ib_send_bw for different packet sizes 256 to 4096 bytes	256 – 234.40 Gb/s 512 – 331.32 Gb/s 1024 – 359.18 Gb/s 2048 – 380.12 Gb/s 4096 – 391.08 Gb/s
ib_read_bw average for all interfaces – RoCEv2 4096-byte packets	389.5 Gb/s
Ib_write_bw average for all interfaces – RoCEv2 4096-byte packets	389.5 Gb/s

12.2.1 MTU sweep with ib_send_bw

An MTU sweep test is used to determine basic NIC and PCIe sanity while showcasing average bandwidth for a specific RoCEv2 NIC. Running this test across different RoCEv2 MTU sizes demonstrates how higher bandwidth is achieved with a higher size. While the logs displayed below (showing only server-side logs) show how this testing is performed using one NIC, the chart provides an overall view of all NICs.

This test was conducted between NIC1 on server-1 (server) and NIC1 on server-2 (client), with traffic passing through the switch. The tests were performed by first running the following command on the server side (for example, on slate5):

```
numactl --cpunodebind=netdev:ens1f0np0 --localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5
-s 1M -q 32 --sl 3 --tclass 104 --report_gbits --ipv6 --ipv6-addr -m 256
```

Afterward, in another terminal window, we ran the following command to start the client on server-2. The specified IPv6 address is the one assigned to the interface mapped to RDMA interface mlx5_5.

```
numactl --cpunodebind=netdev:ens1f0np0 --localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5
-s 1M -q 32 --sl 3 --tclass 104 --report_gbits --ipv6 --ipv6-addr -m 256
fd00:2:9:1:2:1:0:2
```

12.2.1.1 Test results for RoCE MTU size 256

```
nvidia@slate5:~/nvidia-benchmarking-tests/perftest$ numactl --cpunodebind=netdev:ens1f0np0
--localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5 -s 1M -q 32 --sl 3 --tclass 104 --
report_gbits --ipv6 --ipv6-addr -m 256
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_5
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
```

```
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 256[B]
 Link type       : Ethernet
 GID index       : 5
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x060d PSN 0x4935a7 RKey 0x1fff00 VAddr 0x0071d5fdff5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x060e PSN 0x1b4c05 RKey 0x1fff00 VAddr 0x0071d5fe0f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x060f PSN 0xf95d6b RKey 0x1fff00 VAddr 0x0071d5fe1f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0610 PSN 0xe587de RKey 0x1fff00 VAddr 0x0071d5fe2f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0611 PSN 0xe60399 RKey 0x1fff00 VAddr 0x0071d5fe3f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0612 PSN 0xcf5387 RKey 0x1fff00 VAddr 0x0071d5fe4f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0613 PSN 0xaf9bda RKey 0x1fff00 VAddr 0x0071d5fe5f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0614 PSN 0x8873cb RKey 0x1fff00 VAddr 0x0071d5fe6f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0615 PSN 0x148913 RKey 0x1fff00 VAddr 0x0071d5fe7f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0616 PSN 0x2fec99 RKey 0x1fff00 VAddr 0x0071d5fe8f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0617 PSN 0x1668c9 RKey 0x1fff00 VAddr 0x0071d5fe9f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0618 PSN 0x35f428 RKey 0x1fff00 VAddr 0x0071d5feaf5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0619 PSN 0x3dd2f4 RKey 0x1fff00 VAddr 0x0071d5febf5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x061a PSN 0x15f391 RKey 0x1fff00 VAddr 0x0071d5fecf5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x061b PSN 0x587f7b RKey 0x1fff00 VAddr 0x0071d5fedf5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x061c PSN 0xe49923 RKey 0x1fff00 VAddr 0x0071d5feef5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x061d PSN 0x5320fc RKey 0x1fff00 VAddr 0x0071d5feff5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x061e PSN 0xbb0b06 RKey 0x1fff00 VAddr 0x0071d5ff0f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x061f PSN 0xf65469 RKey 0x1fff00 VAddr 0x0071d5ff1f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0620 PSN 0xc6afa2 RKey 0x1fff00 VAddr 0x0071d5ff2f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0621 PSN 0x86ad5b RKey 0x1fff00 VAddr 0x0071d5ff3f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0622 PSN 0xe48d8e RKey 0x1fff00 VAddr 0x0071d5ff4f5000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
```

```
local address: LID 0000 QPN 0x0623 PSN 0x1b50da RKey 0x1fff00 VAddr 0x0071d5ff5f5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0624 PSN 0x18fdda RKey 0x1fff00 VAddr 0x0071d5ff6f5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0625 PSN 0x5522d3 RKey 0x1fff00 VAddr 0x0071d5ff7f5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0626 PSN 0xcc99a2 RKey 0x1fff00 VAddr 0x0071d5ff8f5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0627 PSN 0xdcee1b RKey 0x1fff00 VAddr 0x0071d5ff9f5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0628 PSN 0xef7701 RKey 0x1fff00 VAddr 0x0071d5ffaf5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0629 PSN 0x380b8c RKey 0x1fff00 VAddr 0x0071d5ffbf5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x062a PSN 0xdcb99e RKey 0x1fff00 VAddr 0x0071d5ffcf5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x062b PSN 0x7e0a5c RKey 0x1fff00 VAddr 0x0071d5ffdf5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x062c PSN 0x14381b RKey 0x1fff00 VAddr 0x0071d5ffef5000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
remote address: LID 0000 QPN 0x069b PSN 0x3a7f16 RKey 0x1fff00 VAddr 0x0076761b78d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x069c PSN 0x254a48 RKey 0x1fff00 VAddr 0x0076761b88d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x069d PSN 0x6ed1f2 RKey 0x1fff00 VAddr 0x0076761b98d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x069e PSN 0xd6f59 RKey 0x1fff00 VAddr 0x0076761ba8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x069f PSN 0x4014f8 RKey 0x1fff00 VAddr 0x0076761bb8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a0 PSN 0x5812fa RKey 0x1fff00 VAddr 0x0076761bc8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a1 PSN 0xee71d1 RKey 0x1fff00 VAddr 0x0076761bd8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a2 PSN 0x3977f6 RKey 0x1fff00 VAddr 0x0076761be8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a3 PSN 0xd5f162 RKey 0x1fff00 VAddr 0x0076761bf8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a4 PSN 0x14503c RKey 0x1fff00 VAddr 0x0076761c08d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a5 PSN 0x1d4730 RKey 0x1fff00 VAddr 0x0076761c18d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a6 PSN 0xe73003 RKey 0x1fff00 VAddr 0x0076761c28d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a7 PSN 0x5f1133 RKey 0x1fff00 VAddr 0x0076761c38d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a8 PSN 0x3e0e64 RKey 0x1fff00 VAddr 0x0076761c48d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06a9 PSN 0x97d52 RKey 0x1fff00 VAddr 0x0076761c58d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06aa PSN 0x83d7ae RKey 0x1fff00 VAddr 0x0076761c68d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ab PSN 0x75a42b RKey 0x1fff00 VAddr 0x0076761c78d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
```

```
 remote address: LID 0000 QPN 0x06ac PSN 0x9d2009 RKey 0x1fff00 VAddr 0x0076761c88d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06ad PSN 0xddf8b0 RKey 0x1fff00 VAddr 0x0076761c98d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06ae PSN 0x736bdd RKey 0x1fff00 VAddr 0x0076761ca8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06af PSN 0x93d47a RKey 0x1fff00 VAddr 0x0076761cb8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b0 PSN 0x380fc1 RKey 0x1fff00 VAddr 0x0076761cc8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b1 PSN 0xa79291 RKey 0x1fff00 VAddr 0x0076761cd8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b2 PSN 0x8f62c5 RKey 0x1fff00 VAddr 0x0076761ce8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b3 PSN 0xd3ce2 RKey 0x1fff00 VAddr 0x0076761cf8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b4 PSN 0x6d2c05 RKey 0x1fff00 VAddr 0x0076761d08d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b5 PSN 0x43442 RKey 0x1fff00 VAddr 0x0076761d18d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b6 PSN 0xa35f9c RKey 0x1fff00 VAddr 0x0076761d28d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b7 PSN 0xb1578b RKey 0x1fff00 VAddr 0x0076761d38d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b8 PSN 0xdd2f31 RKey 0x1fff00 VAddr 0x0076761d48d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06b9 PSN 0xd62bf3 RKey 0x1fff00 VAddr 0x0076761d58d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06ba PSN 0xfc2f66 RKey 0x1fff00 VAddr 0x0076761d68d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           258.20             258.17               0.030776
---------------------------------------------------------------------------------------
```

12.2.1.2 Test results for RoCE MTU size 512

```
numactl --cpunodebind=netdev:ens1f0np0 --localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5
-s 1M -q 32 --sl 3 --tclass 104 --report_gbits --ipv6 --ipv6-addr -m 512
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_5
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 512[B]
 Link type       : Ethernet
 GID index       : 5
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
```

```
 Data ex. method : Ethernet
---------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x7470 PSN 0x2f7063 RKey 0x1a1f00 VAddr 0x00740cc5ec6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7471 PSN 0x373611 RKey 0x1a1f00 VAddr 0x00740cc5fc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7472 PSN 0x32cc87 RKey 0x1a1f00 VAddr 0x00740cc60c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7473 PSN 0xfcfeca RKey 0x1a1f00 VAddr 0x00740cc61c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7474 PSN 0x4bae15 RKey 0x1a1f00 VAddr 0x00740cc62c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7475 PSN 0x945253 RKey 0x1a1f00 VAddr 0x00740cc63c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7476 PSN 0x8b60b6 RKey 0x1a1f00 VAddr 0x00740cc64c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7477 PSN 0xc8ed77 RKey 0x1a1f00 VAddr 0x00740cc65c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7478 PSN 0x68df4f RKey 0x1a1f00 VAddr 0x00740cc66c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7479 PSN 0xe28c25 RKey 0x1a1f00 VAddr 0x00740cc67c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x747a PSN 0x8c1f65 RKey 0x1a1f00 VAddr 0x00740cc68c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x747b PSN 0x34dc94 RKey 0x1a1f00 VAddr 0x00740cc69c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x747c PSN 0x27d0f0 RKey 0x1a1f00 VAddr 0x00740cc6ac6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x747d PSN 0x377fdd RKey 0x1a1f00 VAddr 0x00740cc6bc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x747e PSN 0xbd83d7 RKey 0x1a1f00 VAddr 0x00740cc6cc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x747f PSN 0xa21c4f RKey 0x1a1f00 VAddr 0x00740cc6dc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7480 PSN 0x6d82b8 RKey 0x1a1f00 VAddr 0x00740cc6ec6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7481 PSN 0x159012 RKey 0x1a1f00 VAddr 0x00740cc6fc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7482 PSN 0x19c285 RKey 0x1a1f00 VAddr 0x00740cc70c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7483 PSN 0x99b98e RKey 0x1a1f00 VAddr 0x00740cc71c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7484 PSN 0xdbeed7 RKey 0x1a1f00 VAddr 0x00740cc72c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7485 PSN 0x56d75a RKey 0x1a1f00 VAddr 0x00740cc73c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7486 PSN 0x2204b6 RKey 0x1a1f00 VAddr 0x00740cc74c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7487 PSN 0x5b3a86 RKey 0x1a1f00 VAddr 0x00740cc75c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7488 PSN 0x9b800f RKey 0x1a1f00 VAddr 0x00740cc76c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x7489 PSN 0xd6342e RKey 0x1a1f00 VAddr 0x00740cc77c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
```

```
local address: LID 0000 QPN 0x748a PSN 0xdd83b7 RKey 0x1a1f00 VAddr 0x00740cc78c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x748b PSN 0x69526d RKey 0x1a1f00 VAddr 0x00740cc79c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x748c PSN 0xcd8088 RKey 0x1a1f00 VAddr 0x00740cc7ac6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x748d PSN 0xe9f0ea RKey 0x1a1f00 VAddr 0x00740cc7bc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x748e PSN 0xdcddb8 RKey 0x1a1f00 VAddr 0x00740cc7cc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x748f PSN 0x28de47 RKey 0x1a1f00 VAddr 0x00740cc7dc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
remote address: LID 0000 QPN 0x057c PSN 0x459d4a RKey 0x200033 VAddr 0x007b836df8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x057d PSN 0xf3de6c RKey 0x200033 VAddr 0x007b836e08d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x057e PSN 0xac3d46 RKey 0x200033 VAddr 0x007b836e18d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x057f PSN 0x119a1d RKey 0x200033 VAddr 0x007b836e28d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0580 PSN 0x22626c RKey 0x200033 VAddr 0x007b836e38d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0581 PSN 0x39455e RKey 0x200033 VAddr 0x007b836e48d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0582 PSN 0x46be65 RKey 0x200033 VAddr 0x007b836e58d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0583 PSN 0xf0afa RKey 0x200033 VAddr 0x007b836e68d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0584 PSN 0x8b2216 RKey 0x200033 VAddr 0x007b836e78d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0585 PSN 0xebc4e0 RKey 0x200033 VAddr 0x007b836e88d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0586 PSN 0xd54904 RKey 0x200033 VAddr 0x007b836e98d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0587 PSN 0x546f47 RKey 0x200033 VAddr 0x007b836ea8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0588 PSN 0x6a1927 RKey 0x200033 VAddr 0x007b836eb8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0589 PSN 0x5a948 RKey 0x200033 VAddr 0x007b836ec8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x058a PSN 0x5f4866 RKey 0x200033 VAddr 0x007b836ed8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x058b PSN 0xef4732 RKey 0x200033 VAddr 0x007b836ee8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x058c PSN 0x14b75f RKey 0x200033 VAddr 0x007b836ef8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x058d PSN 0x89052d RKey 0x200033 VAddr 0x007b836f08d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x058e PSN 0x5ce104 RKey 0x200033 VAddr 0x007b836f18d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x058f PSN 0xa7cfa1 RKey 0x200033 VAddr 0x007b836f28d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0590 PSN 0x3466ee RKey 0x200033 VAddr 0x007b836f38d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
```

```
 remote address: LID 0000 QPN 0x0591 PSN 0x1aa325 RKey 0x200033 VAddr 0x007b836f48d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0592 PSN 0xbc2c25 RKey 0x200033 VAddr 0x007b836f58d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0593 PSN 0xbfbec9 RKey 0x200033 VAddr 0x007b836f68d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0594 PSN 0x0296 RKey 0x200033 VAddr 0x007b836f78d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0595 PSN 0x5b11a9 RKey 0x200033 VAddr 0x007b836f88d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0596 PSN 0xd05316 RKey 0x200033 VAddr 0x007b836f98d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0597 PSN 0xaef7e0 RKey 0x200033 VAddr 0x007b836fa8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0598 PSN 0x1e447f RKey 0x200033 VAddr 0x007b836fb8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0599 PSN 0x114b15 RKey 0x200033 VAddr 0x007b836fc8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x059a PSN 0x44e407 RKey 0x200033 VAddr 0x007b836fd8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x059b PSN 0xf287ea RKey 0x200033 VAddr 0x007b836fe8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           331.32             331.32               0.039496
```

12.2.1.3 Test results for RoCE MTU size 1024

```
nvidia@slate5:~/nvidia-benchmarking-tests/perftest$ numactl --cpunodebind=netdev:ens1f0np0
--localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5 -s 1M -q 32 --sl 3 --tclass 104 --
report_gbits --ipv6 --ipv6-addr -m 1024
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_5
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 1024[B]
 Link type       : Ethernet
 GID index       : 5
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x064d PSN 0xf9e2b2 RKey 0x1fff00 VAddr 0x007c545c2c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x064e PSN 0xf0dbb4 RKey 0x1fff00 VAddr 0x007c545c3c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x064f PSN 0x52c4ee RKey 0x1fff00 VAddr 0x007c545c4c6000
```

```
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0650 PSN 0xba4ca5 RKey 0x1fff00 VAddr 0x007c545c5c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0651 PSN 0xbc9654 RKey 0x1fff00 VAddr 0x007c545c6c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0652 PSN 0x63cf26 RKey 0x1fff00 VAddr 0x007c545c7c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0653 PSN 0x8d188d RKey 0x1fff00 VAddr 0x007c545c8c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0654 PSN 0xe35e02 RKey 0x1fff00 VAddr 0x007c545c9c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0655 PSN 0xebac7e RKey 0x1fff00 VAddr 0x007c545cac6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0656 PSN 0x402328 RKey 0x1fff00 VAddr 0x007c545cbc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0657 PSN 0xd61dac RKey 0x1fff00 VAddr 0x007c545ccc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0658 PSN 0x9deacf RKey 0x1fff00 VAddr 0x007c545cdc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0659 PSN 0xfe20f RKey 0x1fff00 VAddr 0x007c545cec6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x065a PSN 0xfca410 RKey 0x1fff00 VAddr 0x007c545cfc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x065b PSN 0x37bf8e RKey 0x1fff00 VAddr 0x007c545d0c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x065c PSN 0x7f33a RKey 0x1fff00 VAddr 0x007c545d1c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x065d PSN 0x8526c7 RKey 0x1fff00 VAddr 0x007c545d2c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x065e PSN 0x5fe475 RKey 0x1fff00 VAddr 0x007c545d3c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x065f PSN 0xd4a2ac RKey 0x1fff00 VAddr 0x007c545d4c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0660 PSN 0x823429 RKey 0x1fff00 VAddr 0x007c545d5c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0661 PSN 0xe364d6 RKey 0x1fff00 VAddr 0x007c545d6c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0662 PSN 0x9b2eed RKey 0x1fff00 VAddr 0x007c545d7c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0663 PSN 0x6d604d RKey 0x1fff00 VAddr 0x007c545d8c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0664 PSN 0xeee3d1 RKey 0x1fff00 VAddr 0x007c545d9c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0665 PSN 0xb7f6fe RKey 0x1fff00 VAddr 0x007c545dac6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0666 PSN 0x2391f1 RKey 0x1fff00 VAddr 0x007c545dbc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0667 PSN 0xcfa1be RKey 0x1fff00 VAddr 0x007c545dcc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0668 PSN 0xee6568 RKey 0x1fff00 VAddr 0x007c545ddc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0669 PSN 0xe81767 RKey 0x1fff00 VAddr 0x007c545dec6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x066a PSN 0x5c87dd RKey 0x1fff00 VAddr 0x007c545dfc6000
```

```
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x066b PSN 0x49752f RKey 0x1fff00 VAddr 0x007c545e0c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x066c PSN 0x2e45f2 RKey 0x1fff00 VAddr 0x007c545e1c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
remote address: LID 0000 QPN 0x06db PSN 0x60c28c RKey 0x1fff00 VAddr 0x0076a1ebd8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06dc PSN 0x4a7abd RKey 0x1fff00 VAddr 0x0076a1ebe8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06dd PSN 0x331e8d RKey 0x1fff00 VAddr 0x0076a1ebf8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06de PSN 0xe2dc21 RKey 0x1fff00 VAddr 0x0076a1ec08d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06df PSN 0xb18957 RKey 0x1fff00 VAddr 0x0076a1ec18d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e0 PSN 0x518637 RKey 0x1fff00 VAddr 0x0076a1ec28d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e1 PSN 0x72dd94 RKey 0x1fff00 VAddr 0x0076a1ec38d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e2 PSN 0xba18c7 RKey 0x1fff00 VAddr 0x0076a1ec48d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e3 PSN 0x4bbd29 RKey 0x1fff00 VAddr 0x0076a1ec58d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e4 PSN 0x4e6ac0 RKey 0x1fff00 VAddr 0x0076a1ec68d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e5 PSN 0xd39c9a RKey 0x1fff00 VAddr 0x0076a1ec78d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e6 PSN 0x753d5c RKey 0x1fff00 VAddr 0x0076a1ec88d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e7 PSN 0x6e42e2 RKey 0x1fff00 VAddr 0x0076a1ec98d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e8 PSN 0x36fcb0 RKey 0x1fff00 VAddr 0x0076a1eca8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06e9 PSN 0xa5aee4 RKey 0x1fff00 VAddr 0x0076a1ecb8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ea PSN 0x9c920e RKey 0x1fff00 VAddr 0x0076a1ecc8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06eb PSN 0x65b242 RKey 0x1fff00 VAddr 0x0076a1ecd8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ec PSN 0x39569d RKey 0x1fff00 VAddr 0x0076a1ece8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ed PSN 0x50a0eb RKey 0x1fff00 VAddr 0x0076a1ecf8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ee PSN 0x579bc5 RKey 0x1fff00 VAddr 0x0076a1ed08d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ef PSN 0x1d1d78 RKey 0x1fff00 VAddr 0x0076a1ed18d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06f0 PSN 0xdecb1e RKey 0x1fff00 VAddr 0x0076a1ed28d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06f1 PSN 0x4073f4 RKey 0x1fff00 VAddr 0x0076a1ed38d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06f2 PSN 0xf3d8b6 RKey 0x1fff00 VAddr 0x0076a1ed48d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06f3 PSN 0x7f8749 RKey 0x1fff00 VAddr 0x0076a1ed58d000
```

```
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06f4 PSN 0x6cf0aa RKey 0x1fff00 VAddr 0x0076a1ed68d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06f5 PSN 0x3cb94d RKey 0x1fff00 VAddr 0x0076a1ed78d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06f6 PSN 0xe97415 RKey 0x1fff00 VAddr 0x0076a1ed88d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06f7 PSN 0x251d9 RKey 0x1fff00 VAddr 0x0076a1ed98d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06f8 PSN 0x98499e RKey 0x1fff00 VAddr 0x0076a1eda8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06f9 PSN 0x43e725 RKey 0x1fff00 VAddr 0x0076a1edb8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x06fa PSN 0x7242e7 RKey 0x1fff00 VAddr 0x0076a1edc8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           361.97             361.12               0.043049
---------------------------------------------------------------------------------------
```

12.2.1.4 Test results for RoCE MTU size 2048

```
nvidia@slate5:~/nvidia-benchmarking-tests/perftest$ numactl --cpunodebind=netdev:ens1f0np0
--localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5 -s 1M -q 32 --sl 3 --tclass 104 --
report_gbits --ipv6 --ipv6-addr -m 2048
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_5
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 2048[B]
 Link type       : Ethernet
 GID index       : 5
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x066d PSN 0x54d4ed RKey 0x1fff00 VAddr 0x007735b94c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x066e PSN 0x25740a RKey 0x1fff00 VAddr 0x007735b95c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x066f PSN 0x667256 RKey 0x1fff00 VAddr 0x007735b96c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0670 PSN 0x501eb6 RKey 0x1fff00 VAddr 0x007735b97c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0671 PSN 0x64c0c8 RKey 0x1fff00 VAddr 0x007735b98c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0672 PSN 0x9ba754 RKey 0x1fff00 VAddr 0x007735b99c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
```

```
local address: LID 0000 QPN 0x0673 PSN 0x7f04ed RKey 0x1fff00 VAddr 0x007735b9ac6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0674 PSN 0x4ccfac RKey 0x1fff00 VAddr 0x007735b9bc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0675 PSN 0x315aaa RKey 0x1fff00 VAddr 0x007735b9cc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0676 PSN 0x8060ad RKey 0x1fff00 VAddr 0x007735b9dc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0677 PSN 0x884083 RKey 0x1fff00 VAddr 0x007735b9ec6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0678 PSN 0x6e5d91 RKey 0x1fff00 VAddr 0x007735b9fc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0679 PSN 0x4c9773 RKey 0x1fff00 VAddr 0x007735ba0c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x067a PSN 0x87446d RKey 0x1fff00 VAddr 0x007735ba1c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x067b PSN 0x84085d RKey 0x1fff00 VAddr 0x007735ba2c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x067c PSN 0x376093 RKey 0x1fff00 VAddr 0x007735ba3c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x067d PSN 0x771ee3 RKey 0x1fff00 VAddr 0x007735ba4c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x067e PSN 0xa93d2a RKey 0x1fff00 VAddr 0x007735ba5c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x067f PSN 0x1478f4 RKey 0x1fff00 VAddr 0x007735ba6c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0680 PSN 0x1ead9a RKey 0x1fff00 VAddr 0x007735ba7c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0681 PSN 0x411329 RKey 0x1fff00 VAddr 0x007735ba8c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0682 PSN 0x176d7b RKey 0x1fff00 VAddr 0x007735ba9c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0683 PSN 0xf9238d RKey 0x1fff00 VAddr 0x007735baac6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0684 PSN 0x6612db RKey 0x1fff00 VAddr 0x007735babc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0685 PSN 0x7b870a RKey 0x1fff00 VAddr 0x007735bacc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0686 PSN 0x6c3bd7 RKey 0x1fff00 VAddr 0x007735badc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0687 PSN 0xf62976 RKey 0x1fff00 VAddr 0x007735baec6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0688 PSN 0x6f0b8a RKey 0x1fff00 VAddr 0x007735bafc6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0689 PSN 0xd2ecaa RKey 0x1fff00 VAddr 0x007735bb0c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x068a PSN 0x2dfa9b RKey 0x1fff00 VAddr 0x007735bb1c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x068b PSN 0x3f90de RKey 0x1fff00 VAddr 0x007735bb2c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x068c PSN 0xa1bcac RKey 0x1fff00 VAddr 0x007735bb3c6000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
remote address: LID 0000 QPN 0x06fb PSN 0x765153 RKey 0x1fff00 VAddr 0x00771462f8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
```

```
remote address: LID 0000 QPN 0x06fc PSN 0x330f78 RKey 0x1fff00 VAddr 0x0077146308d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06fd PSN 0x3e742c RKey 0x1fff00 VAddr 0x0077146318d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06fe PSN 0x372ad4 RKey 0x1fff00 VAddr 0x0077146328d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x06ff PSN 0x4fdd8e RKey 0x1fff00 VAddr 0x0077146338d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0700 PSN 0x18cba2 RKey 0x1fff00 VAddr 0x0077146348d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0701 PSN 0x4d8e23 RKey 0x1fff00 VAddr 0x0077146358d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0702 PSN 0x4bbfaa RKey 0x1fff00 VAddr 0x0077146368d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0703 PSN 0x83bdd0 RKey 0x1fff00 VAddr 0x0077146378d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0704 PSN 0xb0bbdb RKey 0x1fff00 VAddr 0x0077146388d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0705 PSN 0x726719 RKey 0x1fff00 VAddr 0x0077146398d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0706 PSN 0x138f6f RKey 0x1fff00 VAddr 0x00771463a8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0707 PSN 0x1846f9 RKey 0x1fff00 VAddr 0x00771463b8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0708 PSN 0x7b647b RKey 0x1fff00 VAddr 0x00771463c8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0709 PSN 0x2a4253 RKey 0x1fff00 VAddr 0x00771463d8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x070a PSN 0xd91251 RKey 0x1fff00 VAddr 0x00771463e8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x070b PSN 0x2880c9 RKey 0x1fff00 VAddr 0x00771463f8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x070c PSN 0x2c9018 RKey 0x1fff00 VAddr 0x0077146408d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x070d PSN 0xe89c4a RKey 0x1fff00 VAddr 0x0077146418d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x070e PSN 0x78fd38 RKey 0x1fff00 VAddr 0x0077146428d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x070f PSN 0xe4ed6f RKey 0x1fff00 VAddr 0x0077146438d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0710 PSN 0x7e4149 RKey 0x1fff00 VAddr 0x0077146448d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0711 PSN 0xf46643 RKey 0x1fff00 VAddr 0x0077146458d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0712 PSN 0x18fe59 RKey 0x1fff00 VAddr 0x0077146468d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0713 PSN 0x64ffb0 RKey 0x1fff00 VAddr 0x0077146478d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0714 PSN 0x41be85 RKey 0x1fff00 VAddr 0x0077146488d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0715 PSN 0xaf218c RKey 0x1fff00 VAddr 0x0077146498d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0716 PSN 0x5c70e8 RKey 0x1fff00 VAddr 0x00771464a8d000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
```

```
 remote address: LID 0000 QPN 0x0717 PSN 0xa189b0 RKey 0x1fff00 VAddr 0x00771464b8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0718 PSN 0xc23a29 RKey 0x1fff00 VAddr 0x00771464c8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x0719 PSN 0x603454 RKey 0x1fff00 VAddr 0x00771464d8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x071a PSN 0x2b59ea RKey 0x1fff00 VAddr 0x00771464e8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           379.67             378.00               0.045061
---------------------------------------------------------------------------------------
```

12.2.1.5 Test results for RoCE MTU size 4096

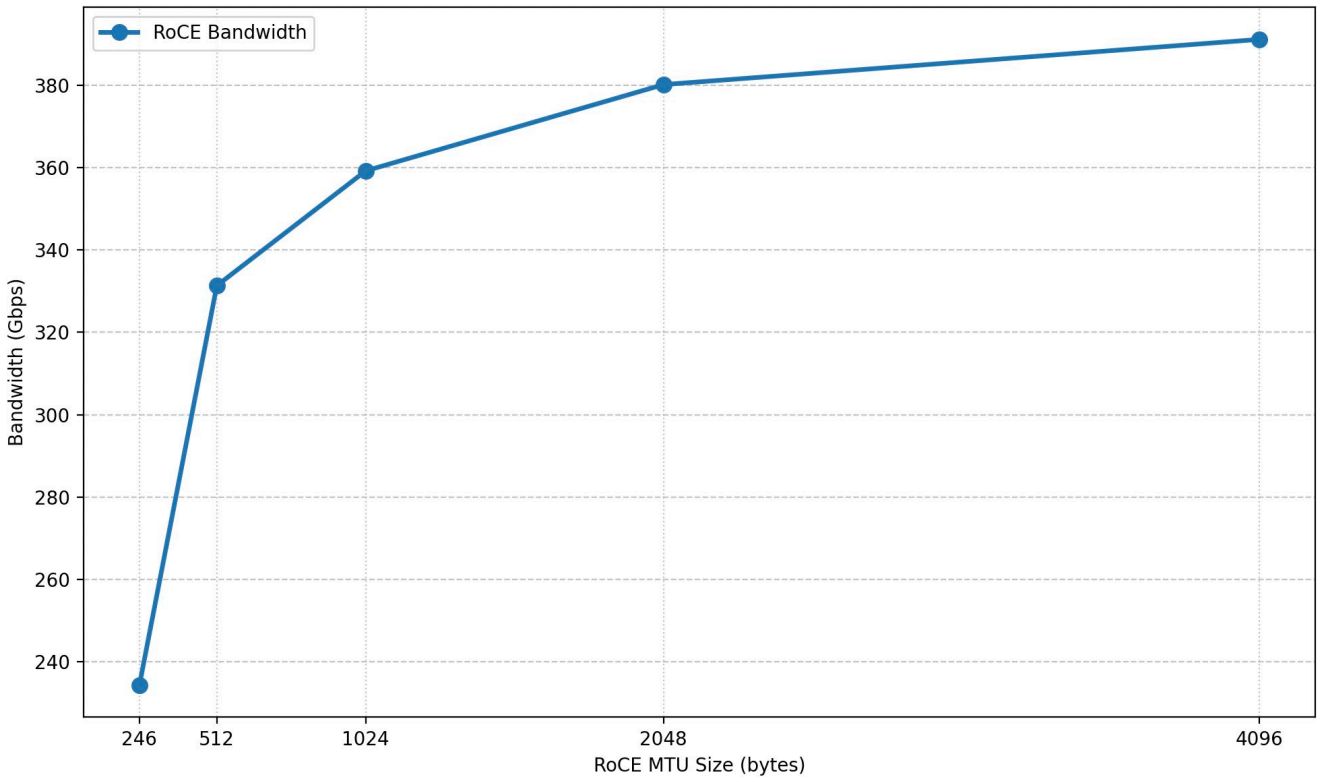
```
nvidia@slate5:~/nvidia-benchmarking-tests/perftest$ numactl --cpunodebind=netdev:ens1f0np0
--localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5 -s 1M -q 32 --sl 3 --tclass 104 --
report_gbits --ipv6 --ipv6-addr -m 4096
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_5
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 4096[B]
 Link type       : Ethernet
 GID index       : 5
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x068d PSN 0x2600b6 RKey 0x1fff00 VAddr 0x007eef9fbc2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x068e PSN 0x613b68 RKey 0x1fff00 VAddr 0x007eef9fcc2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x068f PSN 0x173c92 RKey 0x1fff00 VAddr 0x007eef9fdc2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0690 PSN 0xe9579 RKey 0x1fff00 VAddr 0x007eef9fec2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0691 PSN 0x7e1098 RKey 0x1fff00 VAddr 0x007eef9ffc2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0692 PSN 0x4df61a RKey 0x1fff00 VAddr 0x007eefa00c2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0693 PSN 0x2ee671 RKey 0x1fff00 VAddr 0x007eefa01c2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0694 PSN 0xf2e016 RKey 0x1fff00 VAddr 0x007eefa02c2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0695 PSN 0xd40702 RKey 0x1fff00 VAddr 0x007eefa03c2000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x0696 PSN 0x31455c RKey 0x1fff00 VAddr 0x007eefa04c2000
```

```
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0697 PSN 0x2c65d0 RKey 0x1fff00 VAddr 0x007eefa05c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0698 PSN 0x5afa23 RKey 0x1fff00 VAddr 0x007eefa06c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x0699 PSN 0xc4e0d3 RKey 0x1fff00 VAddr 0x007eefa07c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x069a PSN 0xa13584 RKey 0x1fff00 VAddr 0x007eefa08c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x069b PSN 0xe7e5f2 RKey 0x1fff00 VAddr 0x007eefa09c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x069c PSN 0xb623ce RKey 0x1fff00 VAddr 0x007eefa0ac2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x069d PSN 0xc4cdcb RKey 0x1fff00 VAddr 0x007eefa0bc2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x069e PSN 0x379929 RKey 0x1fff00 VAddr 0x007eefa0cc2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x069f PSN 0xf64b50 RKey 0x1fff00 VAddr 0x007eefa0dc2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a0 PSN 0x8a59fd RKey 0x1fff00 VAddr 0x007eefa0ec2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a1 PSN 0xc7f81a RKey 0x1fff00 VAddr 0x007eefa0fc2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a2 PSN 0xecfae1 RKey 0x1fff00 VAddr 0x007eefa10c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a3 PSN 0x6e6f31 RKey 0x1fff00 VAddr 0x007eefa11c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a4 PSN 0xf312e5 RKey 0x1fff00 VAddr 0x007eefa12c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a5 PSN 0x3bfa82 RKey 0x1fff00 VAddr 0x007eefa13c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a6 PSN 0x31a925 RKey 0x1fff00 VAddr 0x007eefa14c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a7 PSN 0x983ae2 RKey 0x1fff00 VAddr 0x007eefa15c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a8 PSN 0x1df1bc RKey 0x1fff00 VAddr 0x007eefa16c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06a9 PSN 0xaa4f2b RKey 0x1fff00 VAddr 0x007eefa17c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06aa PSN 0xd85e51 RKey 0x1fff00 VAddr 0x007eefa18c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06ab PSN 0x9ffc93 RKey 0x1fff00 VAddr 0x007eefa19c2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x06ac PSN 0xd9c386 RKey 0x1fff00 VAddr 0x007eefa1ac2000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
remote address: LID 0000 QPN 0x071b PSN 0x9a6970 RKey 0x1fff00 VAddr 0x007b8c3374b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x071c PSN 0xf761d1 RKey 0x1fff00 VAddr 0x007b8c3384b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x071d PSN 0xdd7a11 RKey 0x1fff00 VAddr 0x007b8c3394b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x071e PSN 0x546b55 RKey 0x1fff00 VAddr 0x007b8c33a4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x071f PSN 0xa43a7b RKey 0x1fff00 VAddr 0x007b8c33b4b000
```

```
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0720 PSN 0xfd4a8b RKey 0x1fff00 VAddr 0x007b8c33c4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0721 PSN 0x68ed58 RKey 0x1fff00 VAddr 0x007b8c33d4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0722 PSN 0x58e73b RKey 0x1fff00 VAddr 0x007b8c33e4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0723 PSN 0x55dc8d RKey 0x1fff00 VAddr 0x007b8c33f4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0724 PSN 0x1a054 RKey 0x1fff00 VAddr 0x007b8c3404b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0725 PSN 0xb4649e RKey 0x1fff00 VAddr 0x007b8c3414b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0726 PSN 0x8dff10 RKey 0x1fff00 VAddr 0x007b8c3424b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0727 PSN 0xf07486 RKey 0x1fff00 VAddr 0x007b8c3434b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0728 PSN 0x6f7784 RKey 0x1fff00 VAddr 0x007b8c3444b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0729 PSN 0x7c7328 RKey 0x1fff00 VAddr 0x007b8c3454b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x072a PSN 0x133b02 RKey 0x1fff00 VAddr 0x007b8c3464b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x072b PSN 0xa6da26 RKey 0x1fff00 VAddr 0x007b8c3474b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x072c PSN 0xfe2ab1 RKey 0x1fff00 VAddr 0x007b8c3484b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x072d PSN 0x18e56f RKey 0x1fff00 VAddr 0x007b8c3494b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x072e PSN 0xdb5ff9 RKey 0x1fff00 VAddr 0x007b8c34a4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x072f PSN 0x6e5f9c RKey 0x1fff00 VAddr 0x007b8c34b4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0730 PSN 0x144c72 RKey 0x1fff00 VAddr 0x007b8c34c4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0731 PSN 0xa9fcb8 RKey 0x1fff00 VAddr 0x007b8c34d4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0732 PSN 0x732c2a RKey 0x1fff00 VAddr 0x007b8c34e4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0733 PSN 0xa047ad RKey 0x1fff00 VAddr 0x007b8c34f4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0734 PSN 0x68b33e RKey 0x1fff00 VAddr 0x007b8c3504b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0735 PSN 0xaf8a51 RKey 0x1fff00 VAddr 0x007b8c3514b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0736 PSN 0x460ac9 RKey 0x1fff00 VAddr 0x007b8c3524b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0737 PSN 0xc4347d RKey 0x1fff00 VAddr 0x007b8c3534b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0738 PSN 0xf52172 RKey 0x1fff00 VAddr 0x007b8c3544b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x0739 PSN 0x244469 RKey 0x1fff00 VAddr 0x007b8c3554b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x073a PSN 0x7510db RKey 0x1fff00 VAddr 0x007b8c3564b000
```

```
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           389.51             389.13               0.046388
---------------------------------------------------------------------------------------
```

The following graph demonstrates how higher bandwidth is achieved between accelerators at higher MTU sizes.



12.2.2 RDMA read/write using ib_read and ib_write

RDMA read and write tests, using ib_read_bw and ib_write_bw respectively, help measure throughput of read and write operations. With ib_read_bw, the client issues RDMA READ operations from memory on the server NIC, and with ib_write_bw, the client issues RDMA WRITE operations to memory on the server NIC.

12.2.2.1 Test results for RDMA ib_write_bw

```
nvidia@slate5:~/nvidia-benchmarking-tests/perftest$ numactl --cpunodebind=netdev:ens1f0np0
--localalloc ib_write_bw -d mlx5_5 -n 5000 -F -x 5 -s 1M -q 32 --sl 3 --tclass 104 --
report_gbits --ipv6 --ipv6-addr -m 4096
---------------------------------------------------------------------------------------
                    RDMA_Write BW Test
 Dual-port       : OFF          Device         : mlx5_5
```

```
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 4096[B]
 Link type       : Ethernet
 GID index       : 5
 Max inline data : 0[B]
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x5c87 PSN 0xf76402 RKey 0x1a1d5f VAddr 0x0071abd2c51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c88 PSN 0x453dfb RKey 0x1a1d5f VAddr 0x0071abd2d51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c89 PSN 0x89f3f3 RKey 0x1a1d5f VAddr 0x0071abd2e51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c8a PSN 0x9d5c8f RKey 0x1a1d5f VAddr 0x0071abd2f51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c8b PSN 0xe50c2d RKey 0x1a1d5f VAddr 0x0071abd3051000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c8c PSN 0x147f55 RKey 0x1a1d5f VAddr 0x0071abd3151000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c8d PSN 0xefe35a RKey 0x1a1d5f VAddr 0x0071abd3251000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c8e PSN 0xad7215 RKey 0x1a1d5f VAddr 0x0071abd3351000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c8f PSN 0xcd75f RKey 0x1a1d5f VAddr 0x0071abd3451000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c90 PSN 0xdb17be RKey 0x1a1d5f VAddr 0x0071abd3551000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c91 PSN 0x4d78c0 RKey 0x1a1d5f VAddr 0x0071abd3651000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c92 PSN 0x2a1d8a RKey 0x1a1d5f VAddr 0x0071abd3751000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c93 PSN 0x430a78 RKey 0x1a1d5f VAddr 0x0071abd3851000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c94 PSN 0x68bb8e RKey 0x1a1d5f VAddr 0x0071abd3951000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c95 PSN 0xf7676a RKey 0x1a1d5f VAddr 0x0071abd3a51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c96 PSN 0xb8871c RKey 0x1a1d5f VAddr 0x0071abd3b51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c97 PSN 0x5f9d38 RKey 0x1a1d5f VAddr 0x0071abd3c51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c98 PSN 0x83655b RKey 0x1a1d5f VAddr 0x0071abd3d51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c99 PSN 0x3f9bd1 RKey 0x1a1d5f VAddr 0x0071abd3e51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c9a PSN 0x8b13b3 RKey 0x1a1d5f VAddr 0x0071abd3f51000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5c9b PSN 0x2f01ce RKey 0x1a1d5f VAddr 0x0071abd4051000
```

```
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5c9c PSN 0xea47bc RKey 0x1a1d5f VAddr 0x0071abd4151000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5c9d PSN 0xc2773a RKey 0x1a1d5f VAddr 0x0071abd4251000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5c9e PSN 0xc82184 RKey 0x1a1d5f VAddr 0x0071abd4351000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5c9f PSN 0x339aff RKey 0x1a1d5f VAddr 0x0071abd4451000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca0 PSN 0x96d928 RKey 0x1a1d5f VAddr 0x0071abd4551000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca1 PSN 0x1deaf3 RKey 0x1a1d5f VAddr 0x0071abd4651000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca2 PSN 0x6ebbc3 RKey 0x1a1d5f VAddr 0x0071abd4751000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca3 PSN 0x902aef RKey 0x1a1d5f VAddr 0x0071abd4851000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca4 PSN 0x4f7bfc RKey 0x1a1d5f VAddr 0x0071abd4951000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca5 PSN 0x2ccd2b RKey 0x1a1d5f VAddr 0x0071abd4a51000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
local address: LID 0000 QPN 0x5ca6 PSN 0xcd9775 RKey 0x1a1d5f VAddr 0x0071abd4b51000
GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
remote address: LID 0000 QPN 0x6abf PSN 0xde18aa RKey 0x1a226e VAddr 0x007abea594b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac0 PSN 0x38a64c RKey 0x1a226e VAddr 0x007abea5a4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac1 PSN 0x897a6 RKey 0x1a226e VAddr 0x007abea5b4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac2 PSN 0x1654fd RKey 0x1a226e VAddr 0x007abea5c4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac3 PSN 0x8b33cc RKey 0x1a226e VAddr 0x007abea5d4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac4 PSN 0x76ab3e RKey 0x1a226e VAddr 0x007abea5e4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac5 PSN 0x725ec5 RKey 0x1a226e VAddr 0x007abea5f4b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac6 PSN 0x3893da RKey 0x1a226e VAddr 0x007abea604b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac7 PSN 0xb3a976 RKey 0x1a226e VAddr 0x007abea614b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac8 PSN 0x4da8c0 RKey 0x1a226e VAddr 0x007abea624b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ac9 PSN 0xbe8f64 RKey 0x1a226e VAddr 0x007abea634b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6aca PSN 0x61a627 RKey 0x1a226e VAddr 0x007abea644b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6acb PSN 0xd7b687 RKey 0x1a226e VAddr 0x007abea654b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6acc PSN 0x95eb28 RKey 0x1a226e VAddr 0x007abea664b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6acd PSN 0x7a94c6 RKey 0x1a226e VAddr 0x007abea674b000
GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
remote address: LID 0000 QPN 0x6ace PSN 0xad0c12 RKey 0x1a226e VAddr 0x007abea684b000
```

```
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6acf PSN 0x42cabf RKey 0x1a226e VAddr 0x007abea694b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad0 PSN 0xf850d RKey 0x1a226e VAddr 0x007abea6a4b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad1 PSN 0x49364 RKey 0x1a226e VAddr 0x007abea6b4b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad2 PSN 0xd10281 RKey 0x1a226e VAddr 0x007abea6c4b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad3 PSN 0xf4504e RKey 0x1a226e VAddr 0x007abea6d4b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad4 PSN 0xfd4105 RKey 0x1a226e VAddr 0x007abea6e4b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad5 PSN 0x90a485 RKey 0x1a226e VAddr 0x007abea6f4b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad6 PSN 0xdd3fa9 RKey 0x1a226e VAddr 0x007abea704b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad7 PSN 0xd921f6 RKey 0x1a226e VAddr 0x007abea714b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad8 PSN 0x7dad89 RKey 0x1a226e VAddr 0x007abea724b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ad9 PSN 0x17f176 RKey 0x1a226e VAddr 0x007abea734b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ada PSN 0xf7a6c0 RKey 0x1a226e VAddr 0x007abea744b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6adb PSN 0xadf9df RKey 0x1a226e VAddr 0x007abea754b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6adc PSN 0xb5c4f5 RKey 0x1a226e VAddr 0x007abea764b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6add PSN 0x4c0867 RKey 0x1a226e VAddr 0x007abea774b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ade PSN 0x2b44ca RKey 0x1a226e VAddr 0x007abea784b000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           389.82             389.82               0.046470
---------------------------------------------------------------------------------------
```

12.2.2.2 Test results for RDMA ib_read_bw

```
nvidia@slate5:~/nvidia-benchmarking-tests/perftest$ numactl --cpunodebind=netdev:ens1f0np0
--localalloc ib_read_bw -d mlx5_5 -n 5000 -F -x 5 -s 1M -q 32 --sl 3 --tclass 104 --
report_gbits --ipv6 --ipv6-addr -m 4096
---------------------------------------------------------------------------------------
                    RDMA_Read BW Test
 Dual-port       : OFF          Device         : mlx5_5
 Number of qps   : 32           Transport type : IB
 Connection type : RC           Using SRQ      : OFF
 PCIe relax order: ON           Lock-free      : OFF
 ibv_wr* API     : ON           Using Enhanced Reorder      : OFF
 CQ Moderation   : 1
 CQE Poll Batch  : Dynamic
 Mtu             : 4096[B]
```

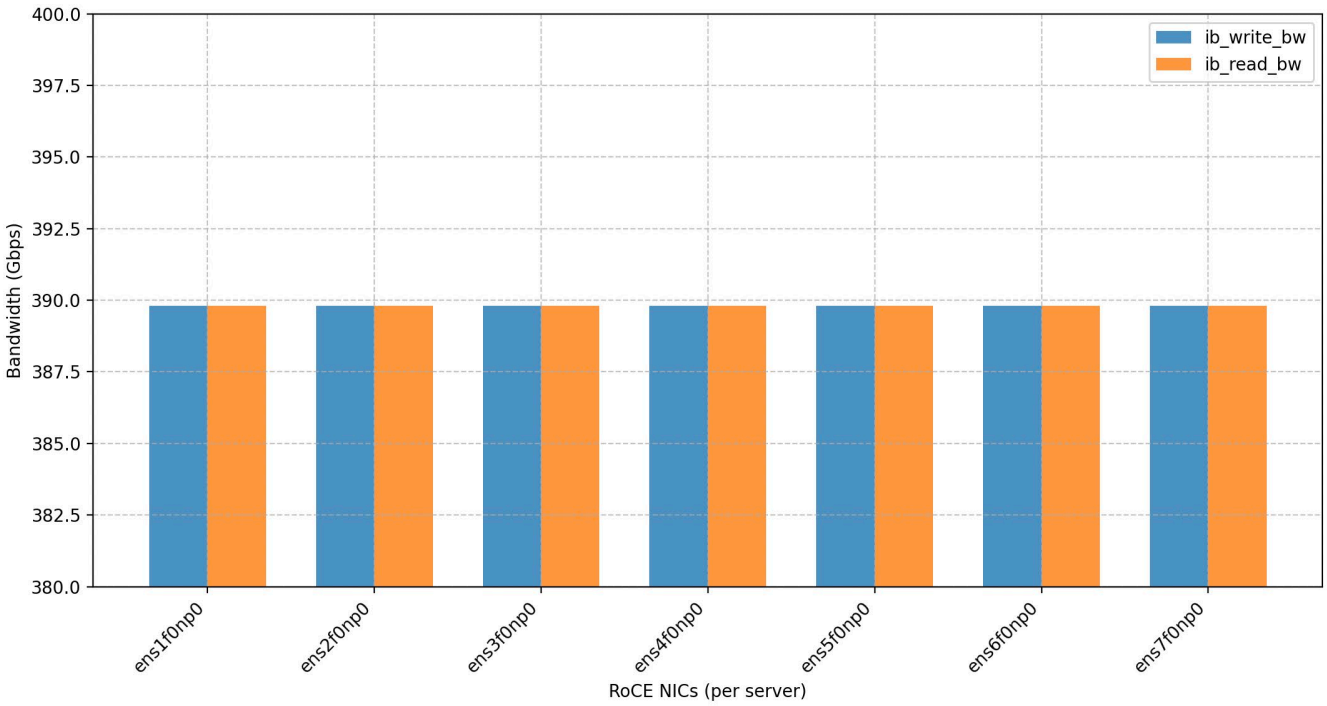
```
 Link type       : Ethernet
 GID index       : 5
 Outstand reads  : 16
 rdma_cm QPs     : OFF
 Data ex. method : Ethernet
---------------------------------------------------------------------------------
 local address: LID 0000 QPN 0x5ca7 PSN 0x2d2f56 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563938c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5ca8 PSN 0xd3d588 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563939c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5ca9 PSN 0xdb9c32 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756393ac6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5caa PSN 0x150c99 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756393bc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cab PSN 0xc58938 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756393cc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cac PSN 0x9c123a OUT 0x10 RKey 0x1a1d60 VAddr
0x00756393dc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cad PSN 0x35a011 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756393ec6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cae PSN 0x15a936 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756393fc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5caf PSN 0xd669a2 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563940c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb0 PSN 0x51037c OUT 0x10 RKey 0x1a1d60 VAddr
0x007563941c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb1 PSN 0x11970 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563942c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb2 PSN 0xcd3543 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563943c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb3 PSN 0x6cd73 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563944c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb4 PSN 0xcab5a4 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563945c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb5 PSN 0x303392 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563946c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb6 PSN 0xbcf0ee OUT 0x10 RKey 0x1a1d60 VAddr
0x007563947c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
```

```
 local address: LID 0000 QPN 0x5cb7 PSN 0x74e46b OUT 0x10 RKey 0x1a1d60 VAddr
0x007563948c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb8 PSN 0x4fb49 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563949c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cb9 PSN 0x11d2f0 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756394ac6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cba PSN 0x9cd91d OUT 0x10 RKey 0x1a1d60 VAddr
0x00756394bc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cbb PSN 0x5ed8ba OUT 0x10 RKey 0x1a1d60 VAddr
0x00756394cc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cbc PSN 0x7a5f01 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756394dc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cbd PSN 0x16d0d1 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756394ec6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cbe PSN 0xda6405 OUT 0x10 RKey 0x1a1d60 VAddr
0x00756394fc6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cbf PSN 0x1c4522 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563950c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc0 PSN 0x3d2f45 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563951c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc1 PSN 0x811682 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563952c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc2 PSN 0x1534dc OUT 0x10 RKey 0x1a1d60 VAddr
0x007563953c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc3 PSN 0xc0a3cb OUT 0x10 RKey 0x1a1d60 VAddr
0x007563954c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc4 PSN 0xe22671 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563955c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc5 PSN 0x16f233 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563956c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 local address: LID 0000 QPN 0x5cc6 PSN 0xae18a6 OUT 0x10 RKey 0x1a1d60 VAddr
0x007563957c6000
 GID: 253:00:00:02:00:09:00:01:00:02:00:01:00:00:00:02
 remote address: LID 0000 QPN 0x6adf PSN 0x63507b OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca38d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae0 PSN 0xc33049 OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca48d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
```

```
 remote address: LID 0000 QPN 0x6ae1 PSN 0x3aa65f OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca58d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae2 PSN 0x2ccfc2 OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca68d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae3 PSN 0x677ad OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca78d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae4 PSN 0x5ab80b OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca88d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae5 PSN 0x8000e OUT 0x10 RKey 0x1a2200 VAddr
0x007589ca98d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae6 PSN 0x21d5ef OUT 0x10 RKey 0x1a2200 VAddr
0x007589caa8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae7 PSN 0xbfaa67 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cab8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae8 PSN 0x6e955d OUT 0x10 RKey 0x1a2200 VAddr
0x007589cac8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6ae9 PSN 0x145c3d OUT 0x10 RKey 0x1a2200 VAddr
0x007589cad8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6aea PSN 0xd48c OUT 0x10 RKey 0x1a2200 VAddr
0x007589cae8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6aeb PSN 0x693588 OUT 0x10 RKey 0x1a2200 VAddr
0x007589caf8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6aec PSN 0x8fe495 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb08d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6aed PSN 0x7db62f OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb18d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6aee PSN 0xd29bc7 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb28d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6aef PSN 0x3998d0 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb38d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af0 PSN 0xf4884a OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb48d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af1 PSN 0xabc25d OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb58d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af2 PSN 0xbfb886 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb68d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
```

```
 remote address: LID 0000 QPN 0x6af3 PSN 0xbc4e6f OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb78d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af4 PSN 0xe21b12 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb88d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af5 PSN 0xe52a0e OUT 0x10 RKey 0x1a2200 VAddr
0x007589cb98d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af6 PSN 0x1f30fe OUT 0x10 RKey 0x1a2200 VAddr
0x007589cba8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af7 PSN 0x1b4127 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cbb8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af8 PSN 0xd6fb66 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cbc8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6af9 PSN 0xcea68f OUT 0x10 RKey 0x1a2200 VAddr
0x007589cbd8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6afa PSN 0x833865 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cbe8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6afb PSN 0x113b20 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cbf8d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6afc PSN 0x84f3a2 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cc08d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6afd PSN 0xee5610 OUT 0x10 RKey 0x1a2200 VAddr
0x007589cc18d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
 remote address: LID 0000 QPN 0x6afe PSN 0xd82bbf OUT 0x10 RKey 0x1a2200 VAddr
0x007589cc28d000
 GID: 253:00:00:02:00:09:00:01:00:02:00:02:00:00:00:02
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
 1048576    160000           389.81             389.80               0.046468
---------------------------------------------------------------------------------------
```

The following graph shows the **ib_read_bw** and **ib_write_bw** throughput values across all eight NICs of an NVIDIA H200 server.



13 MLCommons benchmarks

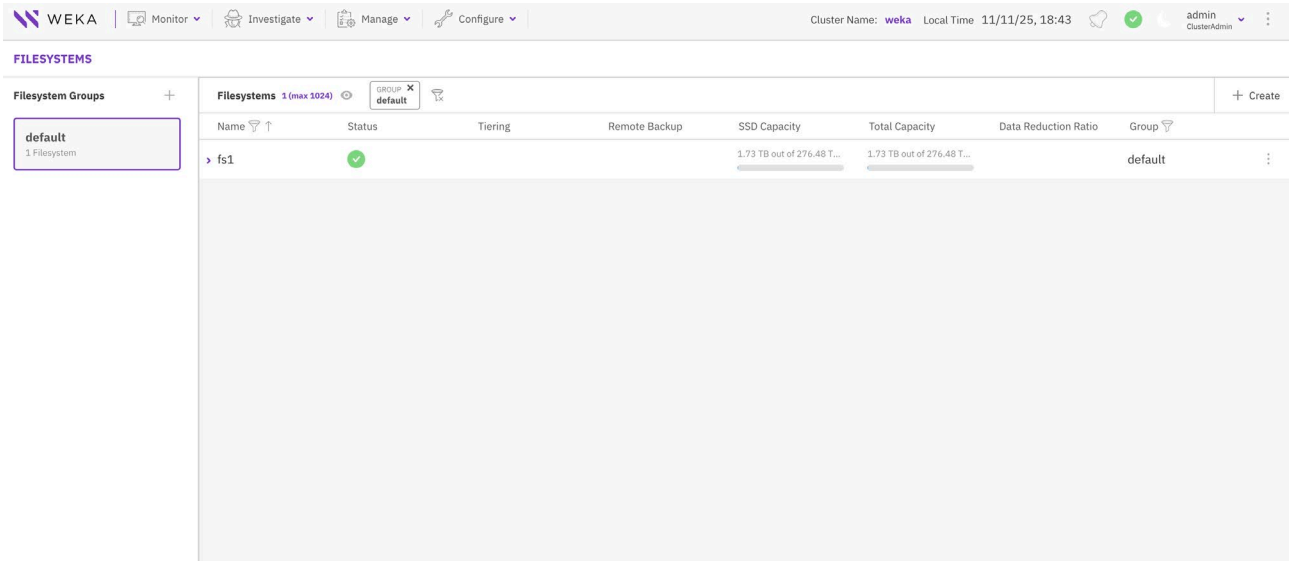
MLCommons benchmarking – Training	
Llama 2 70B – 32 accelerators	4-node training – 6.96 minutes
MLCommons benchmarking – Inference	
BERT – 8 accelerators – offline mode	1-node inference - 73568.6 samples/second

13.1 Training

Llama 2 70B can be benchmarked on NVIDIA H200 GPU clusters for multi-node training using the MLPerf-style LoRA container. This container includes everything needed (scripts and configuration files) to download the dataset and schedule Llama 2 70B jobs via SLURM. First, download the Llama 2 70B-LoRA training dataset, GovReport, to the centralized WEKA storage location. (GovReport is a dataset for long document summarization consisting of reports written by government research agencies.)

Before downloading, ensure WEKA is mounted on all NVIDIA H200 GPU servers and that a directory has been created to store the Llama data in the WEKA-mounted file system. With an 8-node WEKA storage cluster configured in the converged frontend/storage fabric, the GPUs can mount the WEKA filesystem as follows.

1. After the WEKA cluster is up and operational, configure a file system on WEKA.

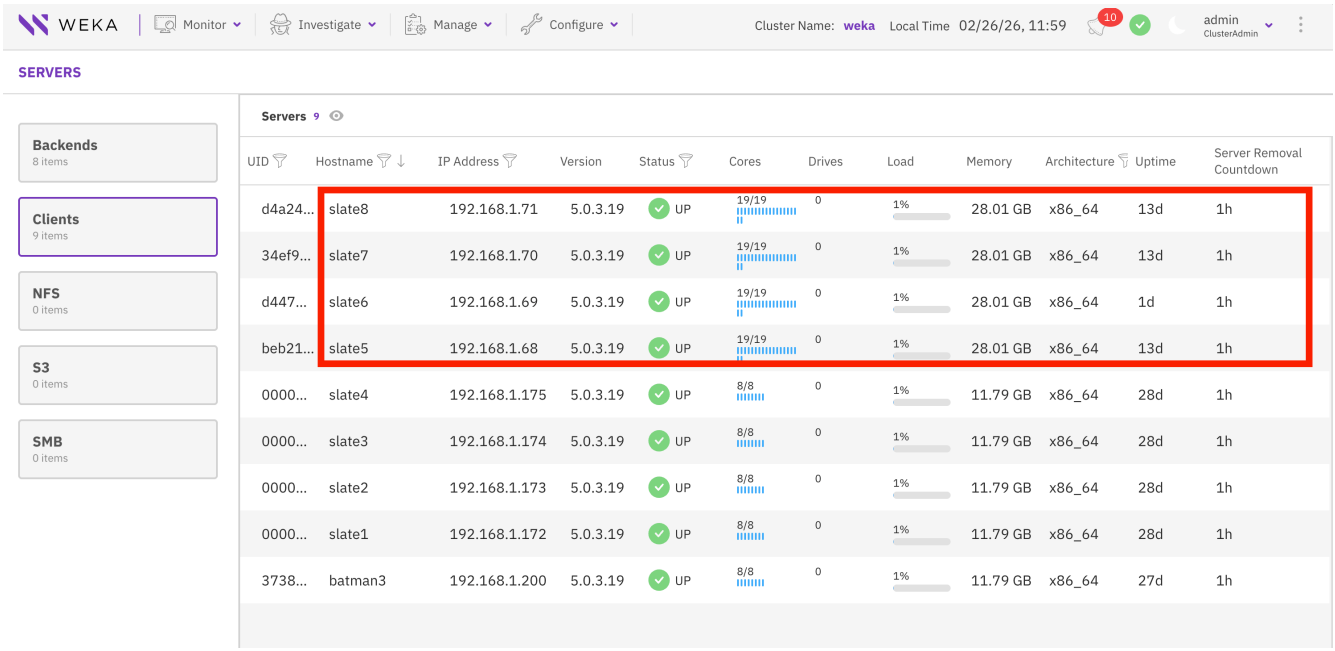


2. Mount the file system on the GPUs by providing redundant WEKA frontend IP addresses. In addition, create a directory on this mounted file system so that the dataset can be stored here and be accessible to all GPUs.

```
nvidia@slate5:~$ sudo mount -t wekafs -o net=udp 192.168.1.189,192.168.1.190/fs1 /mnt/weka
-o num_cores=19
wekafs_mount_helper: Mounting 192.168.1.189,192.168.1.190/fs1 on /mnt/weka
Basing mount on container client
Creating Weka client container 'client' in version 5.0.3.19
Preparing version 5.0.3.19 of container client
Base port was not explicitly provided, the container will use 14000
Successfully set new container staging resources
Applying resources
Starting container 'client'
Mounted tmpfs dir for cleanup files: /opt/weka/data/agent/tmpfss/cleanup
Set permissions on mounted tmpfs dir: /opt/weka/data/agent/tmpfss/cleanup
Waiting for container 'client' to join cluster
client: Container "slate5":"client" allocated 7030 out of 7030 required hugepages after 1
retries
client: Container "slate5":"client" allocated 6327 out of 6327 required hugepages after 1
retries
client: Allocated 26714MB hugepages memory from 2 NUMA nodes for "slate5":"client"
client: Bandwidth of "slate5":"client" set to unlimited
client: WekaFS driver attached by "NodeId<65534>" on "slate5":"client"
Container "client" is ready (pid = 242323)
```

```
wekafs_mount_helper: Executing `mount --no-canonicalize -i -t wekafs -o
inode_bits=auto,dentry_max_age_positive=1000,dentry_max_age_negative=0,readahead_kb=32768,
container_name=client,writecache,relatime,rw,relatime_threshold=0,token=<redacted>,192.168
.1.190/fs1 /mnt/weka`
Cgroups v1 not found, running without cgroups
Mount completed successfully

# create a directory on the mounted WEKA file system to store the llama2 dataset
nvidia@slate5:~$ sudo mkdir -p /mnt/weka/nvidia-training/llama2/
```

3. View the client list on the WEKA UI to confirm that all clients (slate5 through slate8) are up and the expected number of cores have been allocated per client.



Download the GovReport using a docker MLPerf container and store the data in WEKA storage.

4. After it is downloaded, verify that the Llama 2 data is available in the WEKA storage directory you specified earlier when instantiating the container.

```
nvidia@headnode:/home/nvidia# ll /mnt/weka/nvidia-training/llama2/gov_report/
total 195972
drwxr-xr-x 1 root root         0 Jan 30 08:27 ./
drwxr-xr-x 1 root root         0 Jan 30 08:27 ../
-rw-r--r-- 1 root root 192151239 Jan 30 08:27 train.npy
-rw-r--r-- 1 root root   8521136 Jan 30 08:27 validation.npy
nvidia@headnode:/home/nvidia# ll /mnt/weka/nvidia-training/llama2/model/
total 134721896
drwx------ 1 root root            0 Jan 30 08:30 ./
drwxr-xr-x 1 root root            0 Jan 30 08:27 ../
-rwxr-xr-x 1 root root       499723 Jan 30 08:30
b385e3e0cf1a46b19f6a7b2f240a7652_tokenizer.model*
-rw-r--r-- 1 root root 137954713600 Jan 30 08:30 llama2-70b.nemo
```

```
-rw-r--r-- 1 root root         2994 Jan 30 08:30 model_config.yaml
drwxr-xr-x 1 root root            0 Jan 30 08:32 model_weights/
```

5. After all the Llama 2 data has been downloaded, configure the benchmarking scripts that will be used by SLURM to schedule the training job. These scripts have been provided and copied from the MLPerf container to the SLURM head node headnode. For a 4-node MLPerf test using this model, the following parameter script is used.

```
nvidia@headnode:/training/llama2-70b/scripts# cat
configs/config_DGXH200_4x8x1xtp4pp1cp1.sh
#!/bin/bash

# Copyright (c) 2024, NVIDIA CORPORATION.  All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

source $(dirname ${BASH_SOURCE[0]})/config_common.sh

# hyperparameters
export NCCL_TEST=0
export LR=0.0005
export MAX_STEPS=896
export MINIBS=1
export TP=4
export PP=1
export CP=1
export SP=1
export TP_COMM_OVERLAP=1
export VBOOST_VALUE=1
export GPU_ARCH="h"

export FP8=True
export FP8_AMAX_ALGO=max
export FP8_REDUCE_AMAX=True
export FP8_AMAX_HISTORY=32

export SKIP_EVALS=3
export HYDRA_FULL_ERROR=1
export CUDA_DEVICE_MAX_CONNECTIONS=1

# system parameters
export DGXNNODES=4
export SHARP=True
```

```
export WALLTIME_RUNANDTIME=500
export WALLTIME=$((5 + ${NEXP:-1} * ($WALLTIME_RUNANDTIME + 5)))
export MLPERF_NUM_NODES=$DGXNNODES

# additional parameters
export SBATCH_NETWORK=sharp
export NCCL_SOCKET_IFNAME=bond0.100
export NCCL_IB_GID_INDEX=5
export NCCL_IB_HCA=mlx5_0,mlx5_1,mlx5_2,mlx5_3,mlx5_4,mlx5_5,mlx5_6,mlx5_9
export NCCL_IB_DISABLE=0
export OMPI_MCA_pml=ucx
export OMPI_MCA_btl=^openib
export UCX_TLS="shm,rc_mlx5,rc_x,self,tcp"
export
UCX_NET_DEVICES=ens1f0np0.1000,ens2f0np0.1000,ens3f0np0.1000,ens4f0np0.1000,ens5f0np0.1000
,ens6f0np0.1000,ens7f0np0.1000,ens8f0np0.1000
export NCCL_IB_SL=3
export NCCL_IB_TC=104
```

6. Schedule a training job. From the SLURM head node, execute the following **sbatch** command.

```
root@headnode:~#
root@headnode:~# export DATADIR="/mnt/weka/nvidia-training/llama2/gov_report"
root@headnode:~# export MODEL="/mnt/weka/nvidia-training/llama2/model/"
root@headnode:~# export LOGDIR="/training/llama2-70b/logs"
root@headnode:~# export CONT=<CONTAINER>
root@headnode:~# source /training/llama2-70b/scripts/configs/config_DGXH200_4x8x1xtp4pp1cp1.sh
root@headnode:~# export NEXP=1
root@headnode:~# export DGXNGPU=8
root@headnode:~# sbatch --exclusive --gres=gpu:8 -N${DGXNNODES} --ntasks-per-node=${DGXNGPU} --
time=${WALLTIME} /training/llama2-70b/scripts/run.sub
Submitted batch job 92
```

7. Verify if the training job has been scheduled for execution on all four nodes (slate5 through slate8). The full log of the run can be monitored with **tail -f slurm-92.out**.

```
root@headnode:~# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                92  compute1  run.sub     root  R       0:23      4 slate[5-8]
root@headnode:~#
root@headnode:~#
root@headnode:~# scontrol show job 92
JobId=92 JobName=run.sub
   UserId=root(0) GroupId=root(0) MCS_label=N/A
   Priority=1 Nice=0 Account=root QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:01:25 TimeLimit=08:30:00 TimeMin=N/A
   SubmitTime=2026-02-26T13:22:45 EligibleTime=2026-02-26T13:22:45
   AccrueTime=2026-02-26T13:22:45
   StartTime=2026-02-26T13:22:45 EndTime=2026-02-26T21:52:45 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2026-02-26T13:22:45 Scheduler=Main
```

```
   Partition=compute1 AllocNode:Sid=headnode:590698
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=slate[5-8]
   BatchHost=slate5
   NumNodes=4 NumCPUs=1536 NumTasks=32 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   ReqTRES=cpu=32,mem=9287276M,node=4,billing=32,gres/gpu=32
   AllocTRES=cpu=1536,mem=9287276M,node=4,billing=1536,gres/gpu=32
   Socks/Node=* NtasksPerN:B:S:C=8:0:*:1 CoreSpec=*
   MinCPUsNode=8 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=NO Contiguous=0 Licenses=(null) Network=sharp
   Command=/training/llama2-70b/scripts/run.sub
   WorkDir=/root
   StdErr=/root/slurm-92.out
   StdIn=/dev/null
   StdOut=/root/slurm-92.out
TresPerNode=gres/gpu:8
```

Job completion times were monitored across 20 Llama 2 70B runs on a 4-node cluster, averaging 6.96 minutes.

```
llama2-70b/logs/260222232605285048232_1.log -> 6.723600 minutes
llama2-70b/logs/260222232605285048232_2.log -> 7.618750 minutes
llama2-70b/logs/260222232605285048232_3.log -> 7.579783 minutes
llama2-70b/logs/260222232605285048232_4.log -> 7.635550 minutes
llama2-70b/logs/260222232605285048232_5.log -> 6.755133 minutes
llama2-70b/logs/260222232605285048232_6.log -> 6.738466 minutes
llama2-70b/logs/260222232605285048232_7.log -> 6.714066 minutes
llama2-70b/logs/260222232605285048232_8.log -> 7.622350 minutes
llama2-70b/logs/260222232605285048232_9.log -> 5.827183 minutes
llama2-70b/logs/260222232605285048232_10.log -> 6.731566 minutes
llama2-70b/logs/260222232605285048232_11.log -> 7.636683 minutes
llama2-70b/logs/260222232605285048232_12.log -> 6.756366 minutes
llama2-70b/logs/260222232605285048232_13.log -> 7.623983 minutes
llama2-70b/logs/260222232605285048232_14.log -> 6.716950 minutes
llama2-70b/logs/260222232605285048232_15.log -> 6.737333 minutes
llama2-70b/logs/260222232605285048232_16.log -> 6.748283 minutes
llama2-70b/logs/260222232605285048232_17.log -> 6.752366 minutes
llama2-70b/logs/260222232605285048232_18.log -> 6.730333 minutes
llama2-70b/logs/260222232605285048232_19.log -> 6.726183 minutes
llama2-70b/logs/260222232605285048232_20.log -> 6.771500 minutes
```

13.2 Inference

Inference testing was conducted on a single-node NVIDIA H200 GPU system using the BERT-large model for question-answering workloads, following MLPerf-style benchmarks.

This evaluates inference for BERT on a single node, as detailed below. The test achieved 73 568.6 samples per second (QPS) in MLPerf Offline mode.

```
<snip>
```

```
I20260218 15:01:06.989776 11314 bert_core_vs.cc:433] Setup complete
I20260218 15:01:08.842027 11314 main_bert.cc:184] Starting running actual test.

================================================
MLPerf Results Summary
================================================
SUT name : BERT SERVER
Scenario : Offline
Mode     : PerformanceOnly
Samples per second: 73568.6
Result is : VALID
  Min duration satisfied : Yes
  Min queries satisfied : Yes
  Early stopping satisfied: Yes

================================================
Additional Stats
================================================
Min latency (ns)                : 11292714922
Max latency (ns)                : 674635703234
Mean latency (ns)               : 47517832009
50.00 percentile latency (ns)   : 446260744938
90.00 percentile latency (ns)   : 644573784466
95.00 percentile latency (ns)   : 662158287243
97.00 percentile latency (ns)   : 668024343368
99.00 percentile latency (ns)   : 672794392533
99.90 percentile latency (ns)   : 674469295367

================================================
Test Parameters Used
================================================
samples_per_query : 49632000
target_qps : 75200
target_latency (ns): 0
max_async_queries : 1
min_duration (ms): 600000
max_duration (ms): 0
min_query_count : 1
max_query_count : 0
qsl_rng_seed : 3066443479025735752
sample_index_rng_seed : 10688027786191513374
schedule_rng_seed : 14962580496156340209
accuracy_log_rng_seed : 0
accuracy_log_probability : 0
accuracy_log_sampling_target : 0
print_timestamps : 0
performance_issue_unique : 0
performance_issue_same : 0
performance_issue_same_index : 0
performance_sample_count : 10833

8 warnings encountered. See detailed log.

No errors encountered during test.
```

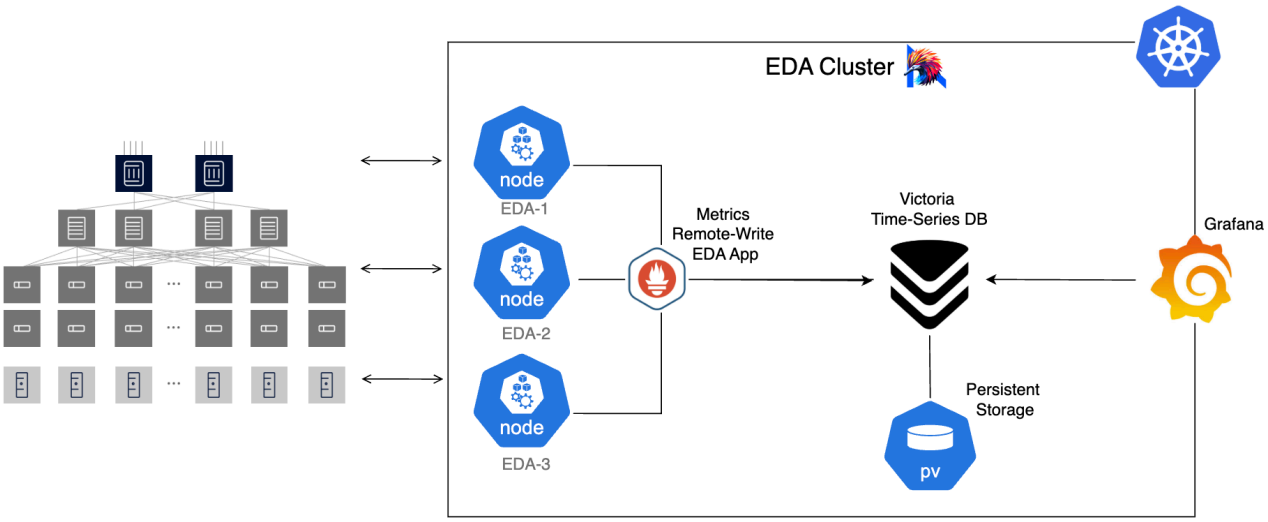
```
I20260218 15:12:33.713733 11314 main_bert.cc:190] Finished running actual test.
[2026-02-18 15:12:38,098 harness.py:114 INFO] setting vboost to gpu default
[2026-02-18 15:12:38,098 __init__.py:46 INFO] Running command: sudo nvidia-smi boost-
slider --vboost 0
sudo: unable to resolve host mlperf-inference-root-x86-64-23751: Name or service not known
Successfully set vboost slider with value 0 for GPU 0
Successfully set vboost slider with value 0 for GPU 1
Successfully set vboost slider with value 0 for GPU 2
Successfully set vboost slider with value 0 for GPU 3
Successfully set vboost slider with value 0 for GPU 4
Successfully set vboost slider with value 0 for GPU 5
Successfully set vboost slider with value 0 for GPU 6
Successfully set vboost slider with value 0 for GPU 7
[2026-02-18 15:12:38,265 run_harness.py:166 INFO] Result: result_samples_per_second:
73568.6, Result is VALID, 10-min runtime requirement met: True

======================== Result summaries: ========================

 nokia_TRT-custom_k_99_MaxP-Offline:
   bert-99:
     performance: result_samples_per_second: 73568.6, Result is VALID, 10-min runtime
requirement met: True


======================== Extra Perf Stats: ========================
```

14 Telemetry

Nokia Event-Driven Automation (EDA) is built on open standards such that integration with third-party tools becomes seamless. EDA provides end-to-end live streaming of telemetry data across data center network fabric, which is essential for proactive monitoring and performance assurance. The metrics collected from EDA can be visualized in the built-in EDA Dashboard Builder, which is a no-code approach to build custom dashboards to display the source data that you define, in a variety of possible layouts. Alternatively, the metrics can be exported to external systems and tools.

Here we have taken the approach to export the metrics collected by EDA to an external system for storage and visualization. The telemetry stack is deployed using Helm charts in the same EDA Kubernetes cluster.

14.1 Architecture



EDA primarily collects the metrics from data center switches centrally though its built-in mechanism using the gRPC/gNMI subscribe method. Then, the metrics data can either be scraped locally from an external system or can be pushed to remote system. In this document, we use the remote-write method. The Remote Write app, which can be installed from the built-in EDA App Store, allows users or operators to select network and EDA metrics of interest to export to remote servers, such as Prometheus or VictoriaMetrics.

14.2 Telemetry tool stack

The telemetry tool stack provides an end-to-end pipeline for collecting, transporting, storing, and visualizing metrics from the AI and data-center network infrastructure. It leverages cloud-native, Prometheus-compatible components to enable open, scalable, and high-performance telemetry ingestion.

Together, the following tools deliver real-time visibility and operational insights across frontend, backend, and storage networks.

1. EDA Remote Write app: This app is an Export Kind Kubernetes app available for installation in the EDA App Store, which is used to stream telemetry metrics to external systems using the Prometheus Remote Write protocol. This app enables integration with time-series databases for metric storage
2. VictoriaMetrics Time-Series Database: VictoriaMetrics is a high-performance time-series database optimized for large-scale metric ingestion. This database natively supports Prometheus-compatible queries and remote write/read APIs and is known for fast data compression and minimal resource usage. VictoriaMetrics is compatible with Prometheus remote read and write and it supports PromQL so that the Grafana dashboards are compatible.

3. Grafana: Grafana is a visualization platform used for building dashboards and monitoring metrics and logs. Grafana integrates with multiple data sources such as Prometheus, VictoriaMetrics, Kafka, and Loki.
4. Alloy: Alloy is a monitoring pipeline tool for collecting processing logs and telemetry data. This tool provides transformations and integration for Grafana and Loki.
5. Loki: Loki is a log aggregation system that streamlines to display logs in a structured format. Loki processes the logs and stores them by indexing metadata instead of the raw full text format, which results in fast, effective log queries.
6. Kafka: Kafka is used to receive real-time alarms from EDA. The Kafka bus ingests and processes events in a publish-subscribe pattern to various topics.

14.3 Setting up telemetry with EDA

The following sections guide you through steps to set up telemetry with EDA. The general workflow is as follows:

1. Install the Remote Write app in EDA.
2. Set up the remote destination in EDA.
3. Set up the exporters in EDA.
4. Set up external time-series database and dashboards.

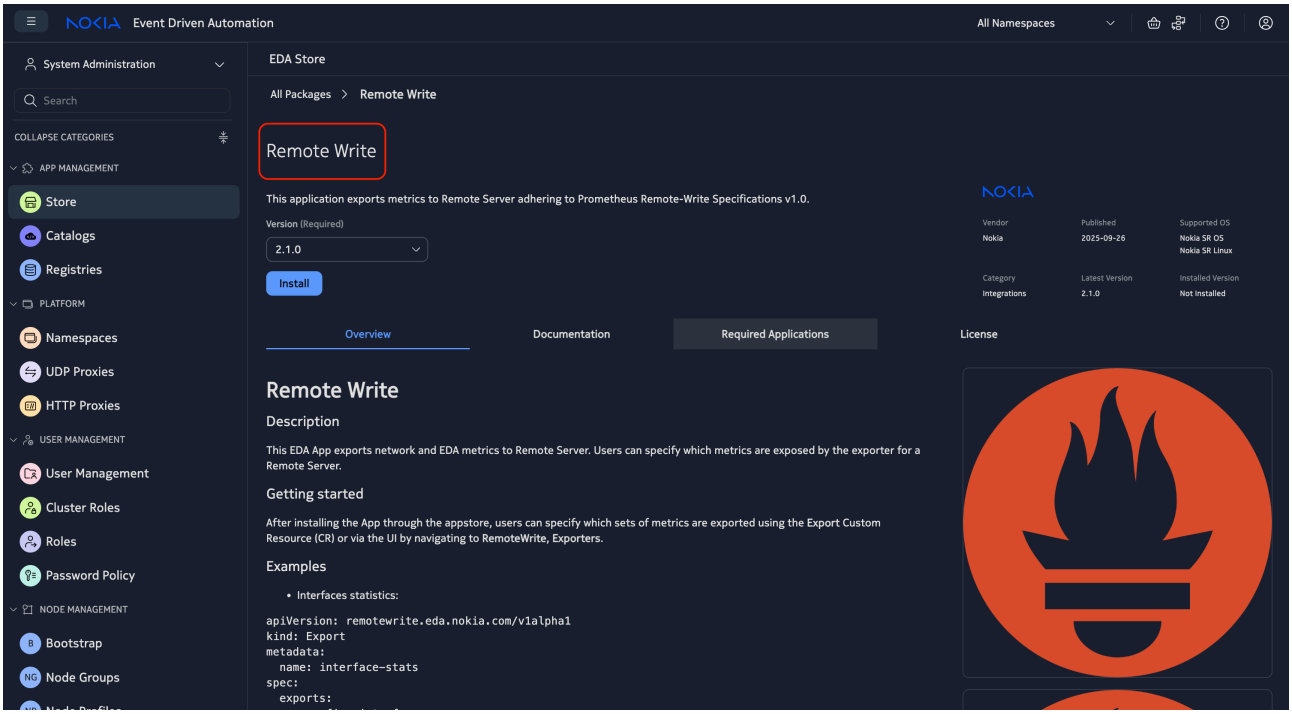
14.3.1 Installing the Remote Write app

Install the Remote Write app using either the Kubernetes API or the EDA UI:

- Using the **kubectl** interface and the following Kubernetes manifest file to install the app:

```
cat << 'EOF' | kubectl apply -f -
apiVersion: core.eda.nokia.com/v1
kind: Workflow
metadata:
  name: remote-write-app
  namespace: eda-system
spec:
  type: app-installer
  input:
    operation: install
    apps:
      - app: remote-write
        catalog: eda-catalog-builtin-apps
        vendor: nokia
        version:
          type: semver
          value: v2.1.0
EOF
```

- Using the EDA UI, navigate to System Administration > EDA Store > Remote Write, then click the Install button:



14.3.2 Setting up the remote destination in EDA

Remote destination information must be updated in EDA. Remote destination is the location where the metrics are pushed from EDA. Details such as TLS, authentication, timeout, buffer size, flush interval, and so on can all be set.

The destination can be set either to cluster-wide, called **`ClusterDestination`** (without namespace, which is systemwide limited), or **`Destination`** (namespace limited).

This can be set up using either the Kubernetes **`kubectl`** API, shown below, or using the EDA UI.

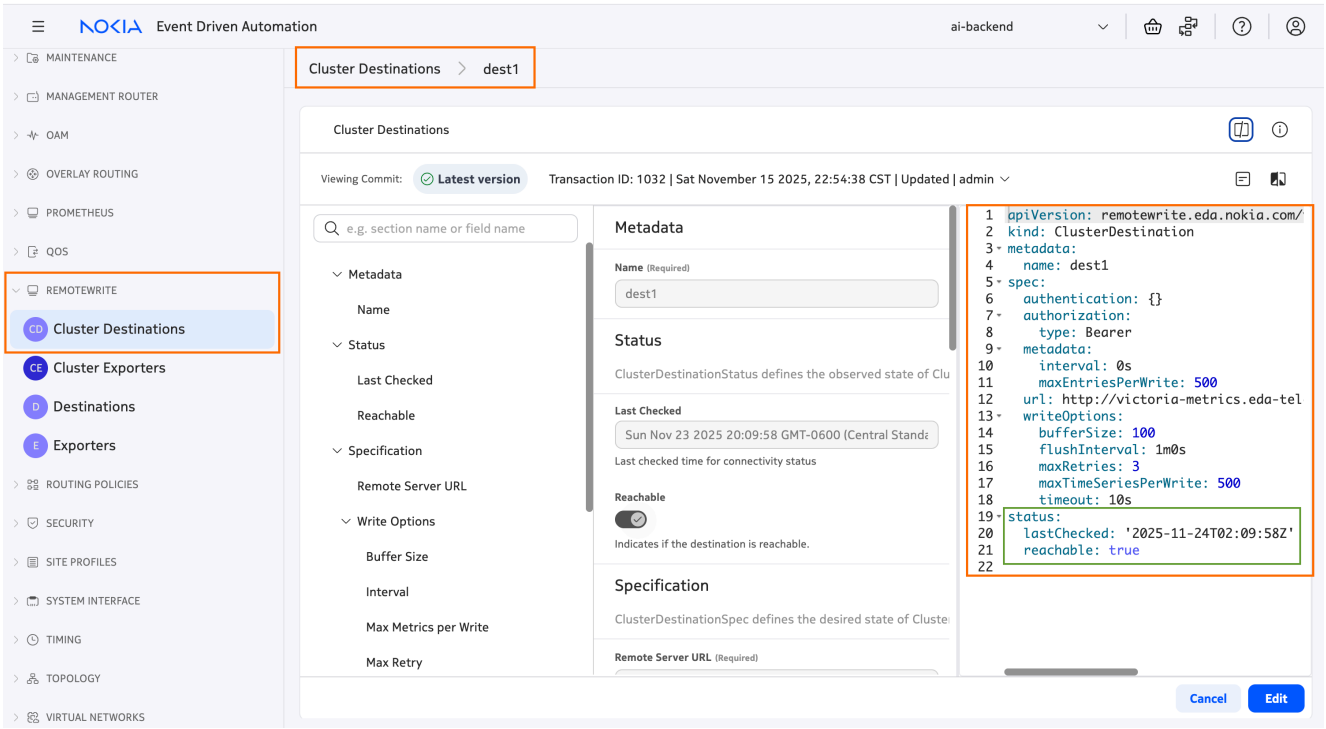
```
apiVersion: remotewrite.eda.nokia.com/v1alpha1
kind: ClusterDestination
metadata:
  name: dest1
spec:
  authentication: {}
  authorization:
    type: Bearer
  metadata:
    interval: 0s
    maxEntriesPerWrite: 500
  url: http://victoria-metrics.eda-telemetry.svc.cluster.local:8428/api/v1/write
  writeOptions:
```

```
    bufferSize: 100
    flushInterval: 1m0s
    maxRetries: 3
    maxTimeSeriesPerWrite: 500
    timeout: 10s
```

The URL/API of the remote metrics database, which could be Prometheus or VictoriaMetrics time series database remote-write API, is as follows:

```
http://victoria-metrics.eda-telemetry.svc.cluster.local:8428/api/v1/write
```

After the destination is configured, ensure the destination is reachable, which can be validated under the **`status`** section highlighted below.



14.3.3 Setting up the exporters in EDA

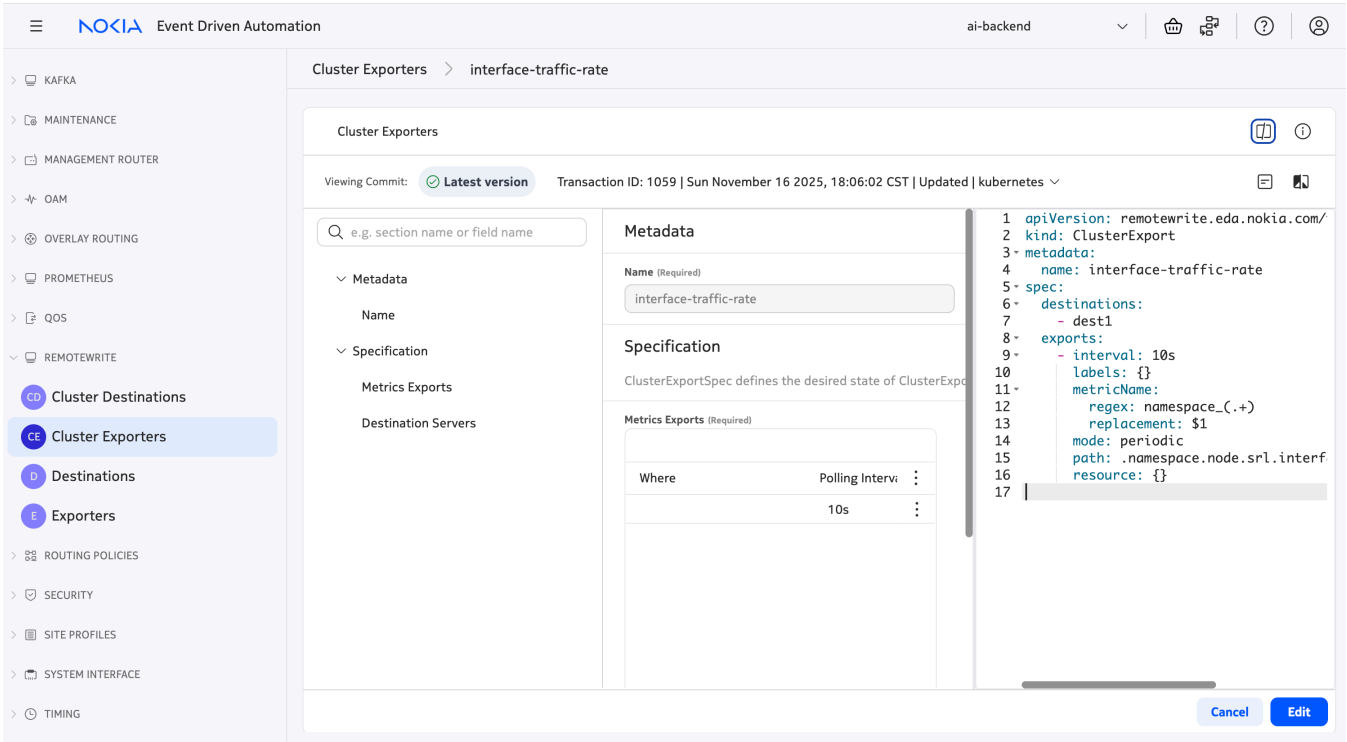
The exporters let the operators publish configured metrics. This can be set up using the kubectl API or EDA-UI. The exporters can be system-wide, called **`ClusterExporters`**, or the namespace-limited **`Exporter`**. Exporters allow us to perform post-processing such as filtering, adding additional labels, mapping, and so on, on the metrics before pushing them to the remote destination.

Export options	Description
path (required)	This is the EQL path of the configured metrics, which can be retrieved from EDA Queries
mode	This is pushing mode: periodic, on-change, periodic-on-change
interval	Polling interval of the metrics
fields	In the metrics path, we can filter by specified fields
labels	User-defined labels can be added for further classification
mappings	Transform field values using regex and numeric replacements
metricName	Rename metrics using regex

The **`Export`** or **`ClusterExport`** has two main components: the first one is destinations, which defines the remote-write target, and the second one is the actual metrics, which will be exported under the section of `exports` and formatting options.

```
apiVersion: remotewrite.eda.nokia.com/v1alpha1
kind: ClusterExport
metadata:
  name: interface-traffic-rate
spec:
  destinations:
    - dest1
  exports:
    - interval: 10s
      labels: {}
      metricName:
        regex: namespace_(.+)
        replacement: $1
      mode: periodic
      path: .namespace.node.srl.interface.traffic-rate
```

EDA UI also lets us set up the export under the Main > REMOTEWRITE > ClusterExporters or Exporters.



14.4 Set up external time-series database and dashboards

The Helm charts install all the necessary tools and configurations with dashboards. In this lab, we are using VictoriaMetrics, which is compatible with v1.0 Prometheus and is used for large data ingestion.

Note: Make sure the destination is reachable to the time-series database.

Installation

The telemetry stack is all packed into a single Helm chart and can be installed using the bash script. Running the script installs all the necessary dependencies and packages, including Grafana dashboards.

```
cse@d2vm-4 ~/telemetry-ai./install_telemetry_stack.sh
Installing telemetry-stack helm chart...
NAME: telemetry-stack
LAST DEPLOYED: Mon Nov 24 04:23:19 2025
NAMESPACE: eda-telemetry
STATUS: deployed
REVISION: 1
TEST SUITE: None

Step-1: Run the following command to access Grafana:
```

```
kubectl port-forward -n eda-telemetry service/grafana 3000:3000 --address=0.0.0.0
>/dev/null 2>&1 & disown
kubectl port-forward -n eda-telemetry svc/victoria-metrics 8428:8428 --address=0.0.0.0
>/dev/null 2>&1 & disown

Step-2: Open your browser and access the following URLs:
You can open Grafana at http://<eda-host-ip>:3000
You can open VictoriaMetrics at http://<eda-host-ip>:8428/vmui
```

After the package has been installed and all the pods are deployed, ensure they are in the running status before forwarding the ports for external access (as described in Step-1).

Verification

All the pods and services of telemetry are deployed in a separate namespace called **eda-telemetry**, which can be verified using the following command:

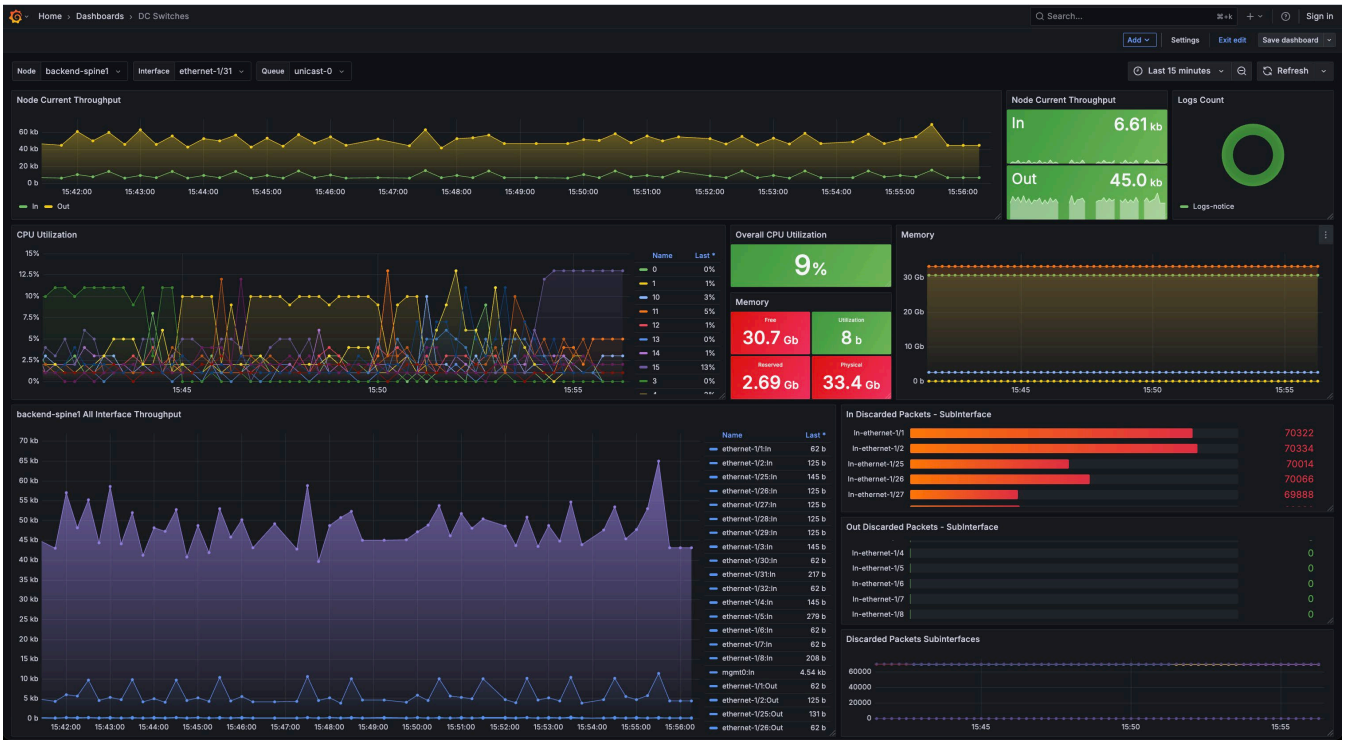
```
cse@eda-node~ kubectl get all -n eda-telemetry

NAME                                        READY   STATUS    RESTARTS        AGE
pod/alloy-b5648665-ltzfw                    1/1     Running   1 (7d22h ago)   7d22h
pod/grafana-6554f68f46-c9vdk                1/1     Running   0               7d22h
pod/kafka-6fbf94cbcb-fklxq                  1/1     Running   0               7d22h
pod/loki-7449c899b8-776ds                   1/1     Running   0               7d22h
pod/vms-victoria-metrics-single-server-0    1/1     Running   0               7d22h

NAME                        TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)                          AGE
service/alloy               LoadBalancer   10.96.3.201     172.18.255.4   12345:30283/TCP,1514:31393/UDP   7d22h
service/grafana             ClusterIP      10.96.61.43     <none>         3000/TCP                         7d22h
service/kafka               ClusterIP      10.96.64.132    <none>         9092/TCP                         7d22h
service/loki                ClusterIP      10.96.115.214   <none>         3100/TCP                         7d22h
service/victoria-metrics    ClusterIP      None            <none>         8428/TCP                         7d22h

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/alloy       1/1     1            1           7d22h
deployment.apps/grafana     1/1     1            1           7d22h
deployment.apps/kafka       1/1     1            1           7d22h
deployment.apps/loki        1/1     1            1           7d22h

NAME                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/alloy-b5648665         1         1         1       7d22h
replicaset.apps/grafana-6554f68f46     1         1         1       7d22h
replicaset.apps/kafka-6fbf94cbcb       1         1         1       7d22h
replicaset.apps/loki-7449c899b8        1         1         1       7d22h

NAME                                                READY   AGE
statefulset.apps/vms-victoria-metrics-single-server   1/1     7d22h
```

The Grafana and Victoria Metrics UI can be accessed by port-forwarding it.

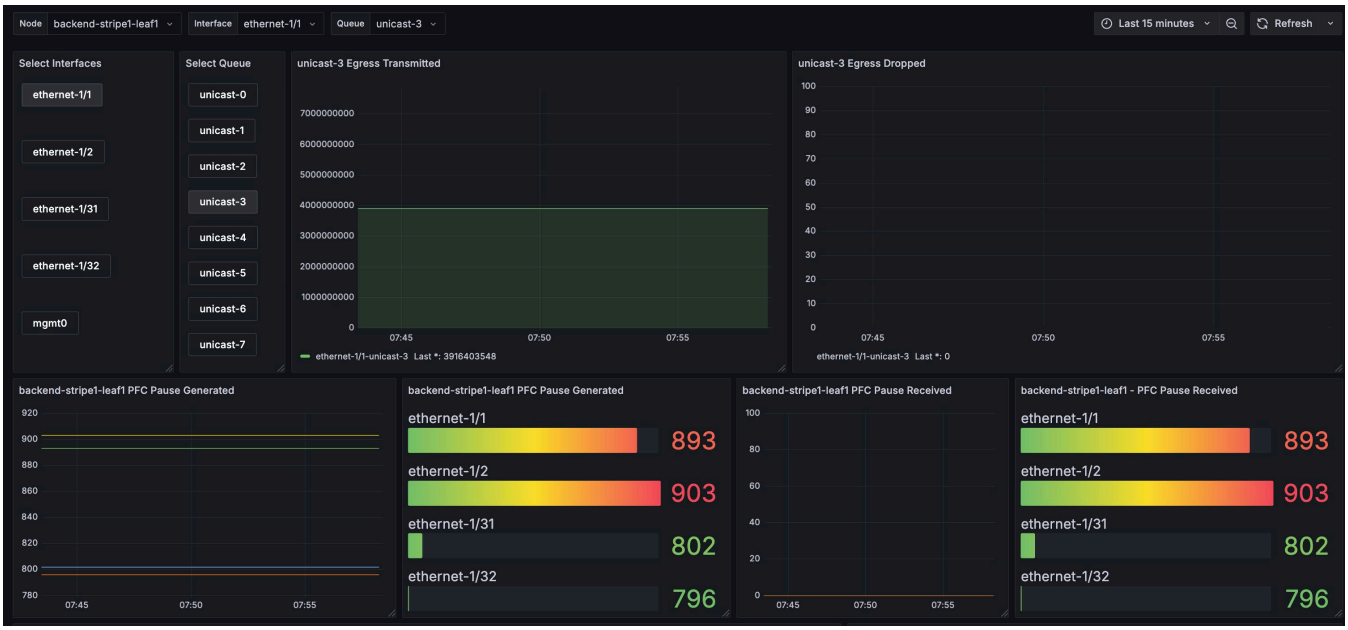
```
kubectl port-forward -n eda-telemetry service/grafana 3000:3000 --address=0.0.0.0
>/dev/null 2>&1 & disown
kubectl port-forward -n eda-telemetry svc/victoria-metrics 8428:8428 --address=0.0.0.0
>/dev/null 2>&1 & disown
```

There are pre-built dashboards in Grafana, as shown below:

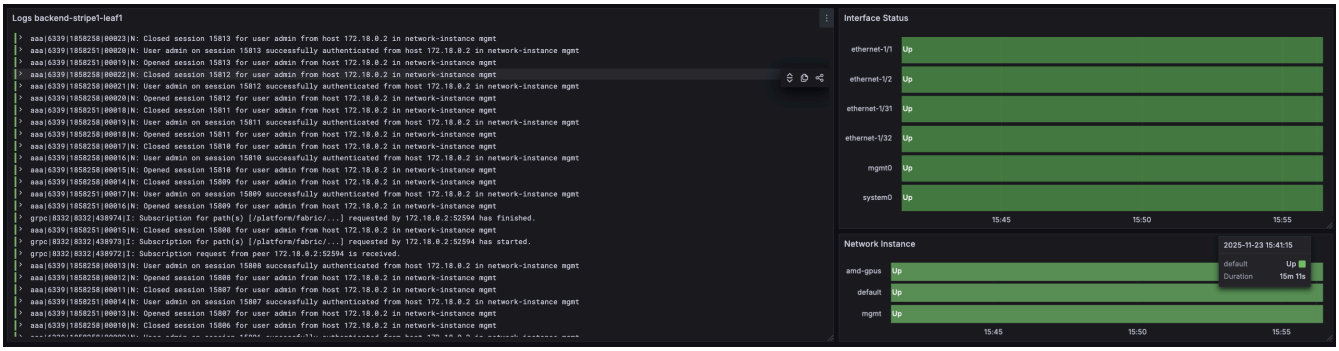
View 1: This dashboard shows statistics collectively from all the data center nodes. At the top right, the display shows the total traffic trend. Below that, the display shows the Top Talkers interfaces in both directions, and to the left, individual nodes with their traffic trend.



View 2: This dashboard shows individual node details, with the top left showing the traffic trend, CPU, and memory utilization, traffic trend per individual interfaces, and to the bottom right it shows the discarded packet count



View 3: This dashboard shows PFC and queue statistics for individual nodes and their interfaces with their queues.



View 4: This dashboard shows logs and alarms to the left and interfaces and VRF status to the right.

15 Digital twin

Digital twins are an integral part of Day-0 through Day-2 operations, providing the operations and deployment teams with the opportunity to continuously validate the look and feel of any deployment. These virtual fabrics also grant the ability to learn and play with technologies and designs—in this case, a prescriptive two-stripe, rail-optimized, AI fabric that has been validated and tuned to provide maximum efficiency and redundancy.

A digital twin of this NVD, deployed using containerlab and containerized SR Linux, can be found here:

https://github.com/nokia/nokia-validated-designs/tree/main/validated-designs/ai-dc/two-stripe-rail-optimized-nvidia