



Nokia Validated Design

AI Cluster with Backend, Frontend, and Storage Networks

3HE-21916-AAAA-TQZZA
Issue 3
February 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.

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice.

No part of this document may be copied, reproduced, modified or transmitted.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2026 Nokia.

Contents

| | | |
|----------|---|-----------|
| 1 | Executive summary | 7 |
| 2 | AI training and inference clusters | 7 |
| 2.1 | Introduction to AI/ML | 7 |
| 2.1.1 | Types of machine learning | 8 |
| 2.1.2 | Collective communication library operations | 9 |
| 2.2 | Types of AI/ML clusters..... | 11 |
| 2.3 | AI-DC cluster architecture and scale | 13 |
| 2.3.1 | Rail optimization | 13 |
| 2.3.2 | Rail-optimized stripe | 14 |
| 2.3.3 | Cluster scale | 15 |
| 2.4 | Components of an AI/ML cluster..... | 16 |
| 2.5 | Traffic flow in AI/ML DC clusters..... | 17 |
| 2.5.1 | Intra-node communication | 17 |
| 2.5.2 | Inter-node communication | 18 |
| 2.5.3 | Inter-stripe communication | 19 |
| 2.6 | Congestion management | 20 |
| 2.6.1 | Congestion scenarios in AI/ML networks | 20 |
| 2.6.1.1 | Scenario 1: Intra-rail simultaneous microburst | 20 |
| 2.6.1.2 | Scenario 2: Inter-stripe spine polarization | 21 |
| 2.6.1.3 | Scenario 3: Inter-stripe leaf-GPU congestion | 22 |
| 2.6.1.4 | Scenario 4: Inter-pod super-spine polarization | 22 |
| 2.6.2 | Datacenter quantized congestion notification..... | 23 |
| 2.6.2.1 | Explicit Congestion Notification..... | 24 |
| 2.6.2.2 | Priority flow control..... | 24 |
| 2.6.2.3 | ECN and PFC threshold | 25 |
| 2.7 | Load balancing | 27 |
| 2.7.1 | Static ECMP hash-based load balancing | 27 |
| 2.7.2 | Dynamic load balancing | 28 |
| 2.7.2.1 | Queue-pair hashing..... | 30 |
| 2.8 | Multitenancy | 30 |
| 2.8.1 | Server isolation..... | 31 |
| 2.8.2 | GPU isolation..... | 32 |
| 3 | Hardware and optics | 33 |
| 4 | Nokia portfolio | 35 |
| 5 | Reference architecture and network orchestration | 37 |
| 5.1 | High-level overview | 37 |
| 5.2 | Design considerations..... | 38 |
| 5.2.1 | Backend design | 38 |
| 5.2.2 | Collapsed frontend and storage design..... | 39 |
| 5.2.3 | Multitenant reference design | 40 |
| 6 | Compute server orchestration | 41 |
| 6.1 | Server resource specifications | 41 |
| 6.2 | SMCI-AMD MI300X server orchestration..... | 41 |

| | | |
|----------|--|-----------|
| 6.2.1 | Upgrade the GPU firmware | 41 |
| 6.2.2 | AMD Instinct MI300X system optimization | 43 |
| 6.2.2.1 | BIOS settings | 44 |
| 6.2.2.2 | GRUB settings | 44 |
| 6.2.2.3 | Operating system settings | 45 |
| 6.2.3 | AMD ROCm installation..... | 46 |
| 6.2.3.1 | Prerequisites | 46 |
| 6.2.3.2 | ROCm installation | 47 |
| 6.2.3.3 | Post-installation | 47 |
| 6.2.4 | AOC-S400G-B1C 400G NIC driver installation | 50 |
| 6.2.4.1 | Verify driver installation | 50 |
| 6.2.4.2 | GPU NIC mapping and configuration..... | 52 |
| 6.3 | IP addressing | 54 |
| 7 | WEKA storage orchestration | 56 |
| 7.1 | Server resource specifications | 56 |
| 7.2 | WEKA server optimization | 56 |
| 7.3 | Network settings..... | 58 |
| 7.4 | Building the storage cluster..... | 60 |
| 7.5 | Creating the storage file system..... | 64 |
| 7.6 | Mounting the WEKA system | 66 |
| 8 | Network feature configuration | 67 |
| 8.1 | Backend network..... | 67 |
| 8.1.1 | Point-to-point interfaces between leafs and spines..... | 68 |
| 8.1.2 | Point-to-point interfaces between leafs and GPUs..... | 69 |
| 8.1.3 | BGP in the fabric for route distribution | 70 |
| 8.1.4 | Dynamic load balancing | 72 |
| 8.1.5 | Route aggregation for GPU subnets | 72 |
| 8.1.6 | IP VRFs for tenant isolation in the fabric | 77 |
| 8.1.7 | Quality of service for RoCEv2 traffic | 80 |
| 8.2 | Converged frontend and storage network..... | 86 |
| 8.2.1 | Point-to-point interfaces between leafs and spines for underlay..... | 86 |
| 8.2.2 | Default network instance | 87 |
| 8.2.3 | BGP configuration for underlay and overlay..... | 87 |
| 8.2.4 | Ethernet segments to GPU servers..... | 89 |
| 8.2.5 | Ethernet segments to WEKA storage nodes..... | 91 |
| 8.2.6 | Ethernet segments to frontend server | 92 |
| 8.2.7 | MAC VRFs for GPU storage NICs, storage cluster, and frontend server..... | 93 |
| 9 | EDA integration | 96 |
| 9.1 | Backend fabric..... | 99 |
| 9.1.1 | Creating a namespace | 99 |
| 9.1.2 | Onboarding TopoNodes using ZTP | 99 |
| 9.1.3 | Creating interfaces | 100 |
| 9.1.4 | Creating TopoLinks | 100 |
| 9.1.5 | AI backend fabric deployment | 101 |
| 9.2 | Converged frontend and storage fabric | 102 |
| 9.2.1 | Creating a namespace | 102 |
| 9.2.2 | Onboarding TopoNodes | 103 |
| 9.2.3 | Creating interfaces | 103 |
| 9.2.4 | Creating TopoLinks | 104 |

| | | |
|-----------|--|------------|
| 9.2.5 | Fabric deployment | 105 |
| 9.2.6 | Virtual networks for bridge domains (MAC VRFs) and VLANs | 106 |
| 9.2.7 | EDA configlets | 108 |
| 10 | Test summary | 112 |
| 11 | Validation | 114 |
| 11.1 | Network validation | 114 |
| 11.1.1 | IPv6 unnumbered validation | 114 |
| 11.1.2 | BGP validation | 115 |
| 11.1.3 | BFD validation | 116 |
| 11.1.4 | GPU subnets route validation | 117 |
| 11.1.5 | IP VRFs import and export validation | 118 |
| 11.1.6 | MAC VRFs validation for frontend/storage fabric | 120 |
| 11.1.7 | LAGs and Ethernet segments validation for frontend/storage fabric..... | 121 |
| 11.1.8 | QoS validation | 122 |
| 11.2 | EDA validation..... | 123 |
| 11.2.1 | EDA pods validation | 123 |
| 11.2.2 | TopoNode validation..... | 124 |
| 11.2.3 | AI backend network validation..... | 125 |
| 11.2.4 | Frontend network validation | 127 |
| 11.2.5 | Data center network interfaces status | 128 |
| 11.2.5.1 | Frontend interfaces | 128 |
| 11.2.5.2 | Backend interfaces..... | 129 |
| 11.2.6 | Virtual networks validation | 131 |
| 11.2.6.1 | Detailed virtual network information..... | 131 |
| 11.3 | Storage server validation | 133 |
| 11.3.1 | WEKA storage status | 133 |
| 11.3.2 | WEKA cluster status | 133 |
| 11.3.3 | WEKA filesystem validation | 134 |
| 11.3.4 | WEKA cluster processes | 134 |
| 11.4 | GPU server validation | 135 |
| 11.4.1 | GPU NICs | 137 |
| 11.4.2 | RoCEv2 statistics | 138 |
| 11.4.3 | Physical layer statistics..... | 140 |
| 11.4.4 | Server networking verification | 140 |
| 12 | Performance tests | 141 |
| 12.1 | RCCL operations test..... | 141 |
| 12.1.1 | Test results for AllReduce collective | 142 |
| 12.1.2 | Test results for ReduceScatter collective | 146 |
| 12.1.3 | Test results for Broadcast collective..... | 150 |
| 12.1.4 | Test results for All-to-All collective..... | 154 |
| 12.2 | Broadcom NIC throughput | 157 |
| 12.2.1 | MTU sweep with ib_send_bw..... | 157 |
| 12.2.1.1 | Test results for RoCE MTU size 256 | 158 |
| 12.2.1.2 | Test results for RoCE MTU size 1024 | 161 |
| 12.2.1.3 | Test results for RoCE MTU size 2048 | 164 |
| 12.2.1.4 | Test results for RoCE MTU size 4096 | 167 |
| 12.2.2 | RDMA read/write using ib_read and ib_write | 170 |
| 12.2.2.1 | Test results for RDMA ib_write_bw..... | 170 |
| 12.2.2.2 | Test results for RDMA ib_read_bw | 173 |

| | | |
|-----------|--|------------|
| 13 | MLCommons benchmarks | 178 |
| 13.1 | Training | 179 |
| 13.1.1 | Llama 2 70B using local storage for dataset | 179 |
| 13.1.2 | Llama 2 70B using remote storage (WEKA-mounted file system) for dataset 188 | |
| 13.2 | Inference | 191 |
| 13.2.1 | Results for single-node inference testing | 193 |
| 13.2.2 | Results for 2-node inference testing | 194 |
| 13.2.3 | Results for 3-node inference testing | 195 |
| 13.2.4 | Results for 4-node inference testing | 196 |
| 13.2.5 | Inference performance scaling | 197 |
| 14 | Telemetry | 198 |
| 14.1 | Architecture | 199 |
| 14.2 | Telemetry tool stack | 199 |
| 14.3 | Setting up telemetry with EDA | 200 |
| 14.3.1 | Installing the Remote Write app | 200 |
| 14.3.2 | Setting up the remote destination in EDA | 201 |
| 14.3.3 | Setting up the exporters in EDA | 202 |
| 14.4 | Set up external time-series database and dashboards | 204 |
| 15 | Digital twin | 207 |

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, and, 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.

The solution described in this NVD is tested on Broadcom Thor 2 network interface cards. See the *AMD Pensando Pollara Validation* on the Nokia Documentation Center for a network-level validation with performance tests for this same solution using AMD Pensando Pollara network interface cards.

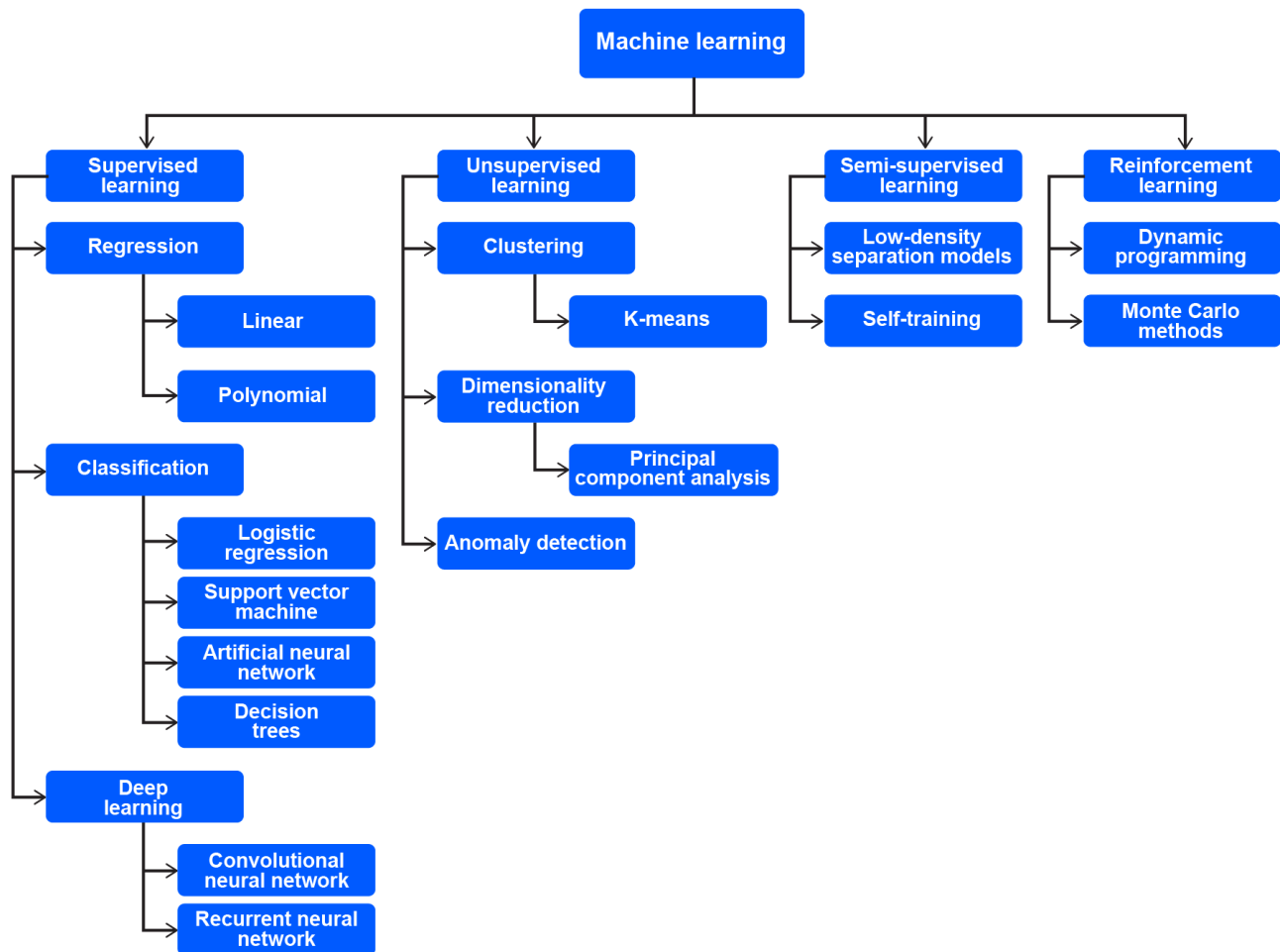
Find more information about NVDs at <https://www.nokia.com/ip-networks/validated-designs/>.

2 AI training and inference clusters

2.1 Introduction to AI/ML

Machine learning is a method by which models learn from data without explicitly being programmed 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



sw4700

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 labelled and then unlabeled data.

Training data: Labelled and unlabeled inputs

Learning process: The model first learns from labelled 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.

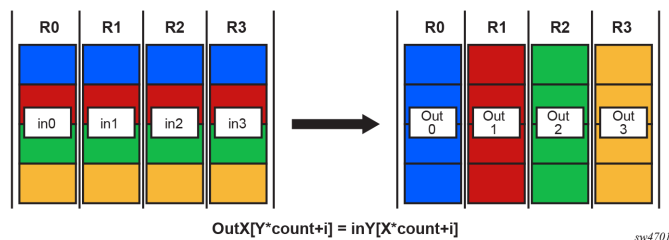
Training data: Unlabeled input

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

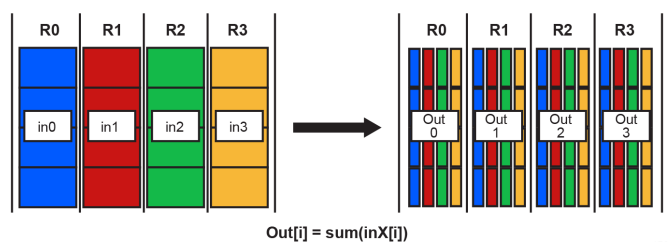
The learning mechanisms described in Section 2.1.1 Types of machine learning need specific workload distribution and synchronization mechanisms. Some of these operations are described below.

All-to-ALL



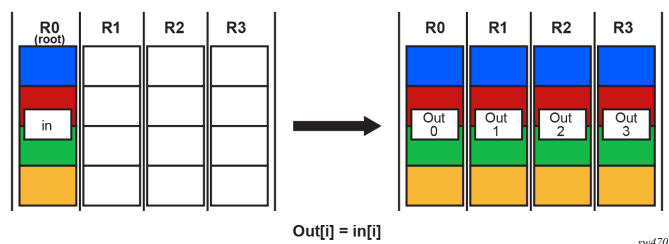
The All-to-All operation gathers N values from k ranks into an output buffer of size $k \times N$ and distributes that result to all ranks.

AllReduce



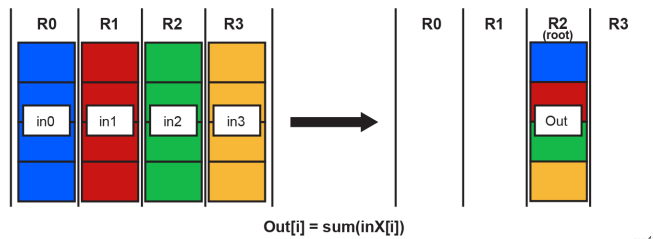
The AllReduce operation performs reductions on data (for example, sum, min, max) across devices and stores the result in the receive buffer of every rank.

Broadcast



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.

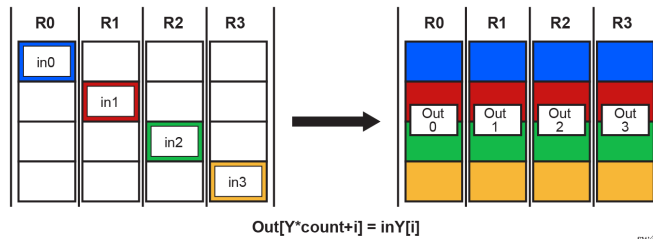
Reduce



The Reduce operation performs the same operation as AllReduce but stores the result only in the receive buffer of a specified root rank.

sw4704

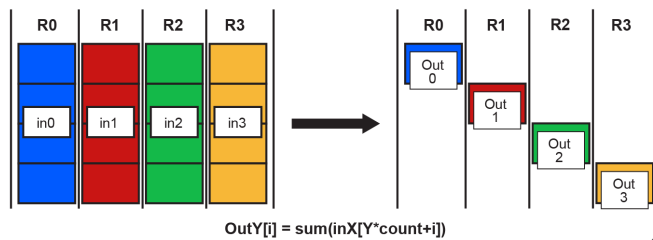
AllGather



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.

sw4705

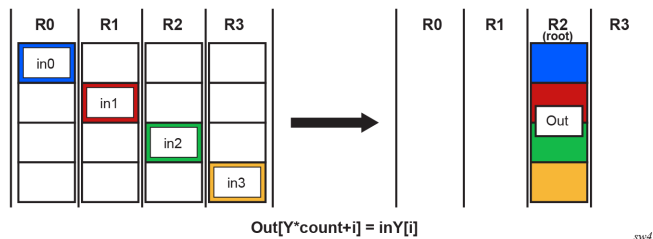
ReduceScatter



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.

sw4706

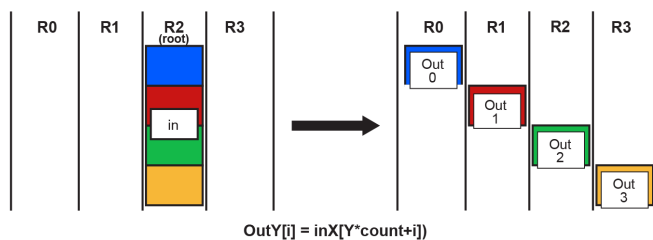
Gather



The Gather operation gathers N values from k ranks into an output buffer on the root rank of size $k \times N$.

sw4707

Scatter



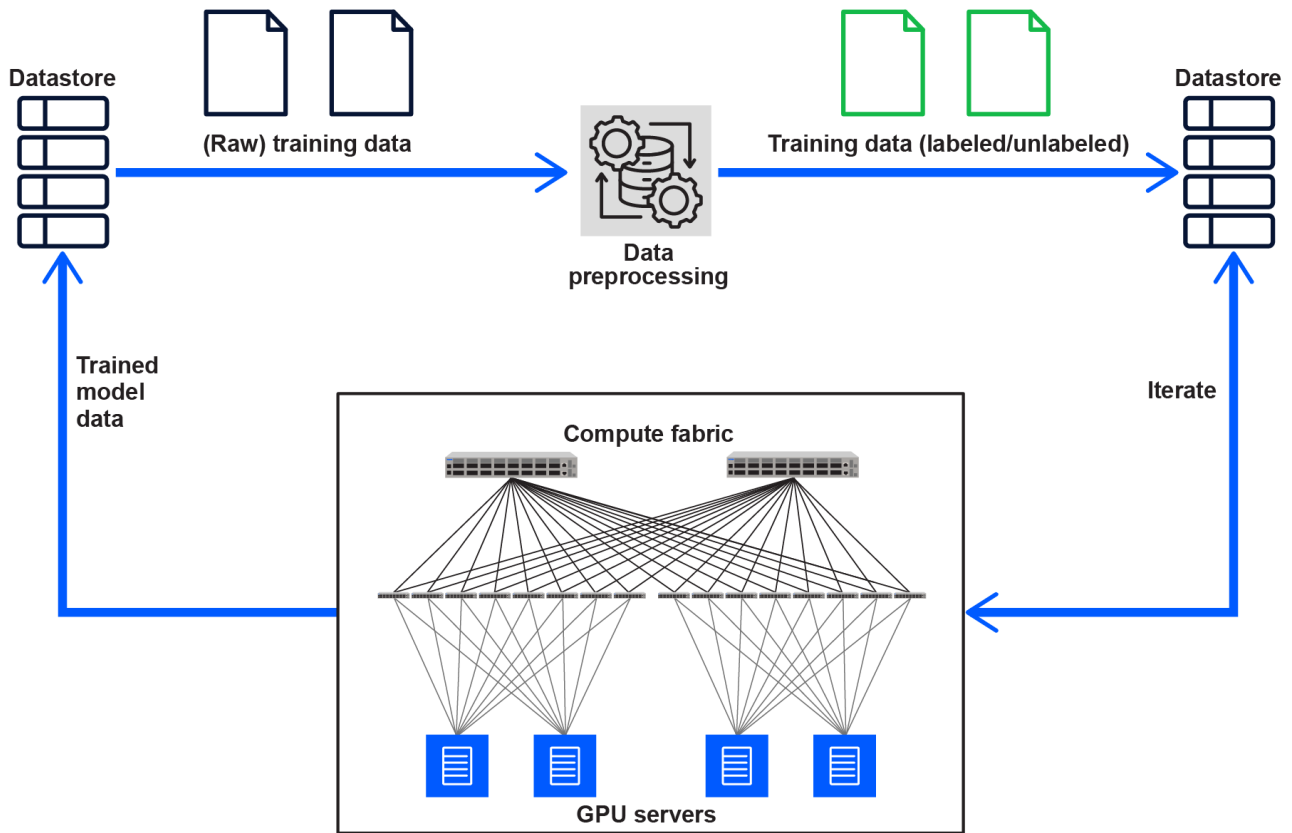
The Scatter operation distributes a total of $N \times k$ values from the root rank to k ranks, each rank receiving N values.

sw4708

Table 1 Collective communications library operations

2.2 Types of AI/ML clusters

Training cluster

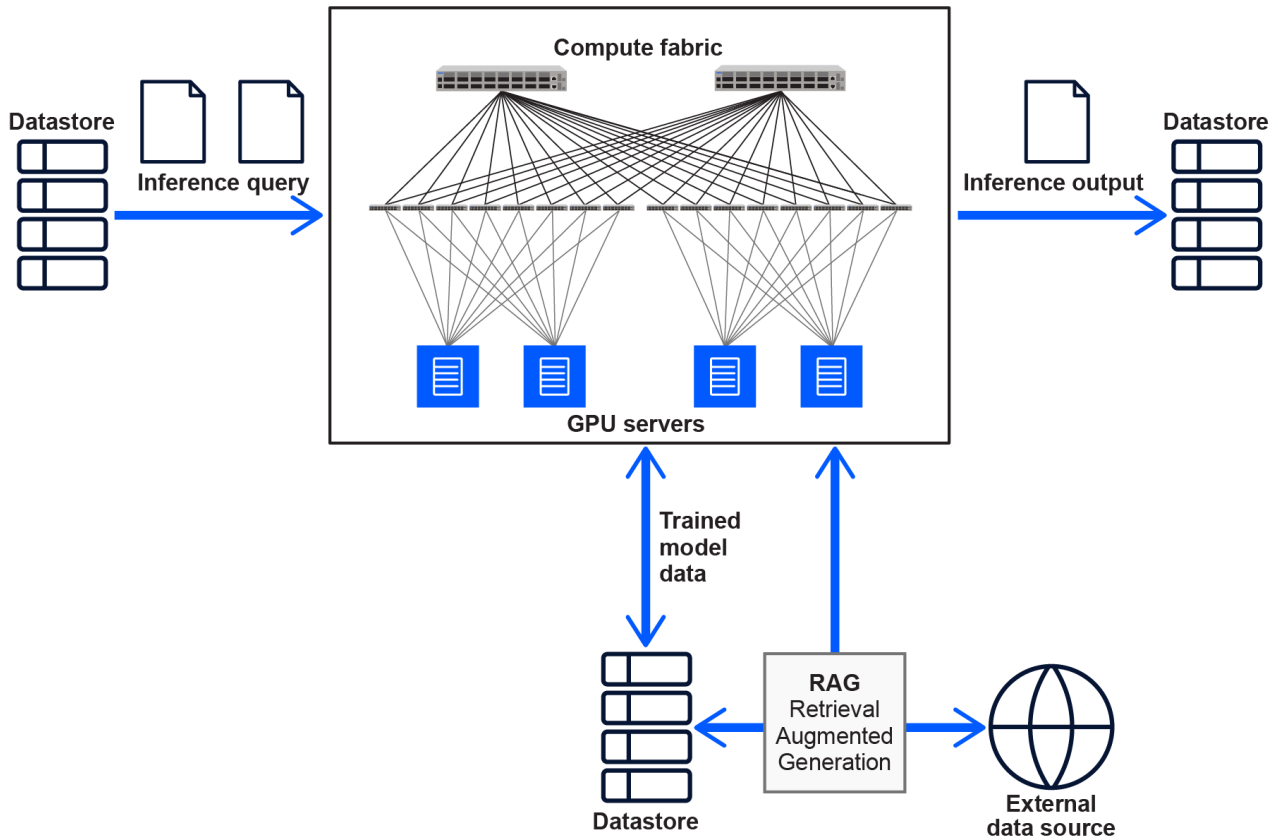


sw4709

Figure 2. Training cluster operation

The primary function of a training cluster is to train the learning models with different forms of data (labelled and un-labelled), 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



sw4710

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 both training and an inference hybrid cluster.

2.3 AI-DC cluster architecture and scale

2.3.1 Rail optimization

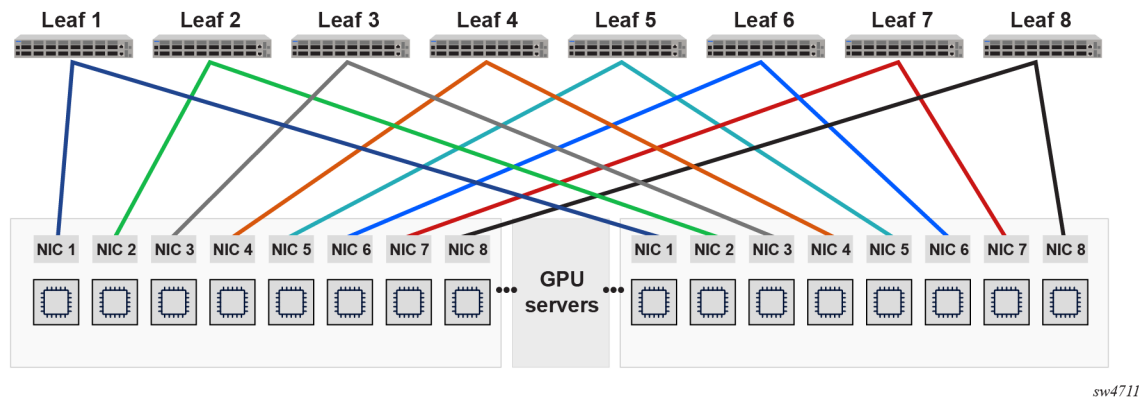


Figure 4. Rail-optimized fabric

sw4711

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

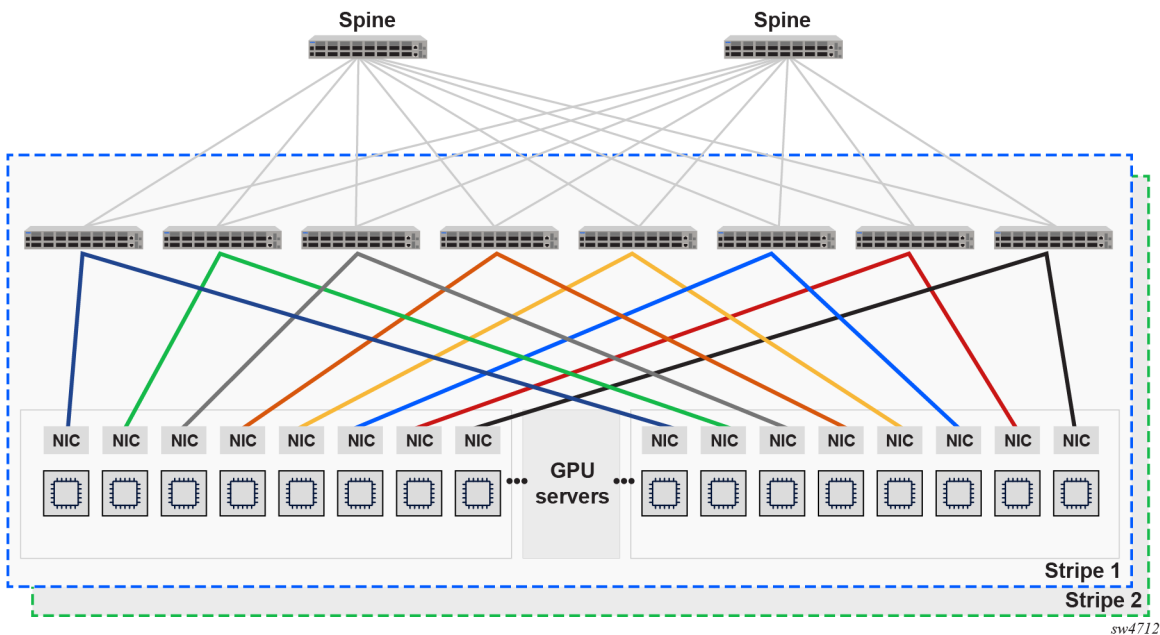


Figure 5. Rail-optimized stripe

A stripe is an atomic unit of a GPU cluster.

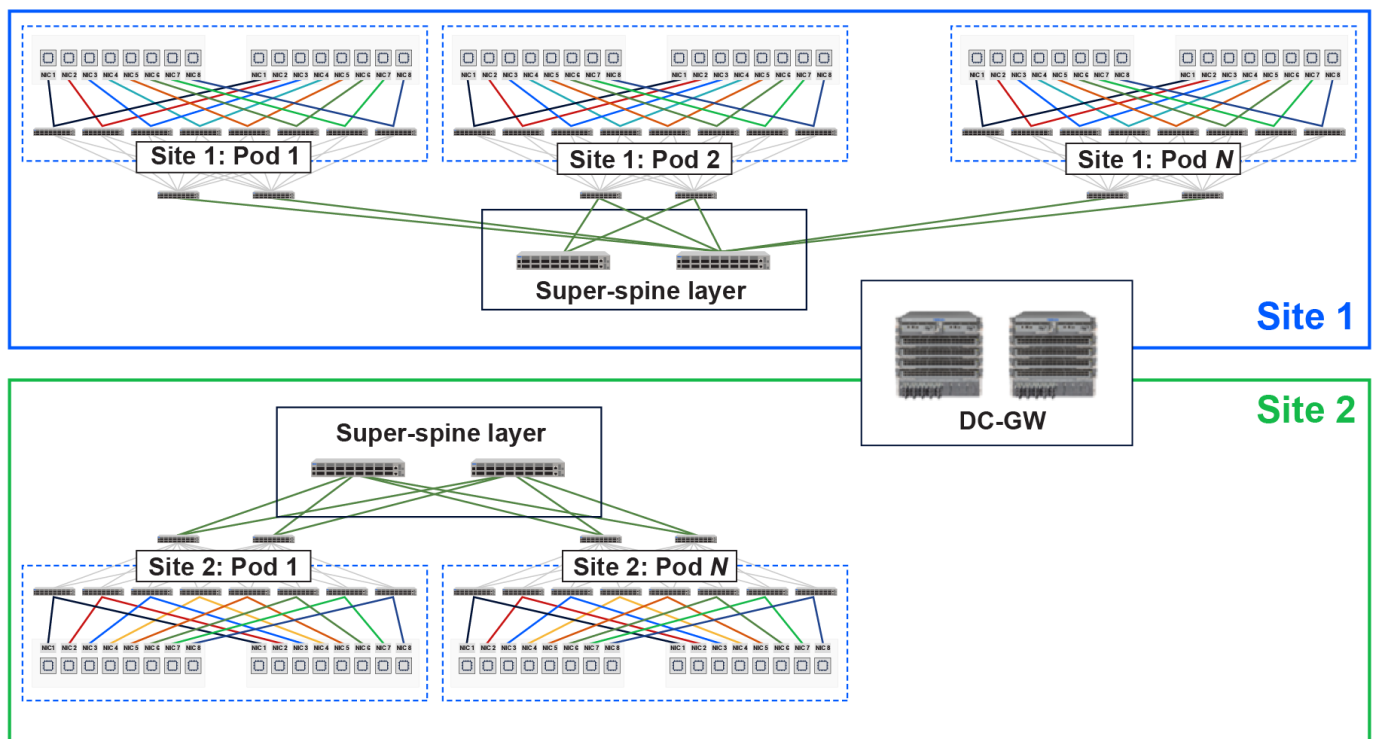
Earlier, we discussed rail optimization which resulted in eight leafs being the design choice for rail optimization because the GPU servers could accommodate eight GPUs.

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 32 port switches in the leaf role, assuming an oversubscription ratio of 1, the number of GPUs in 1 stripe will be 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



sw4713

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 leaves 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 $64 \times 64 = 4096$ 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

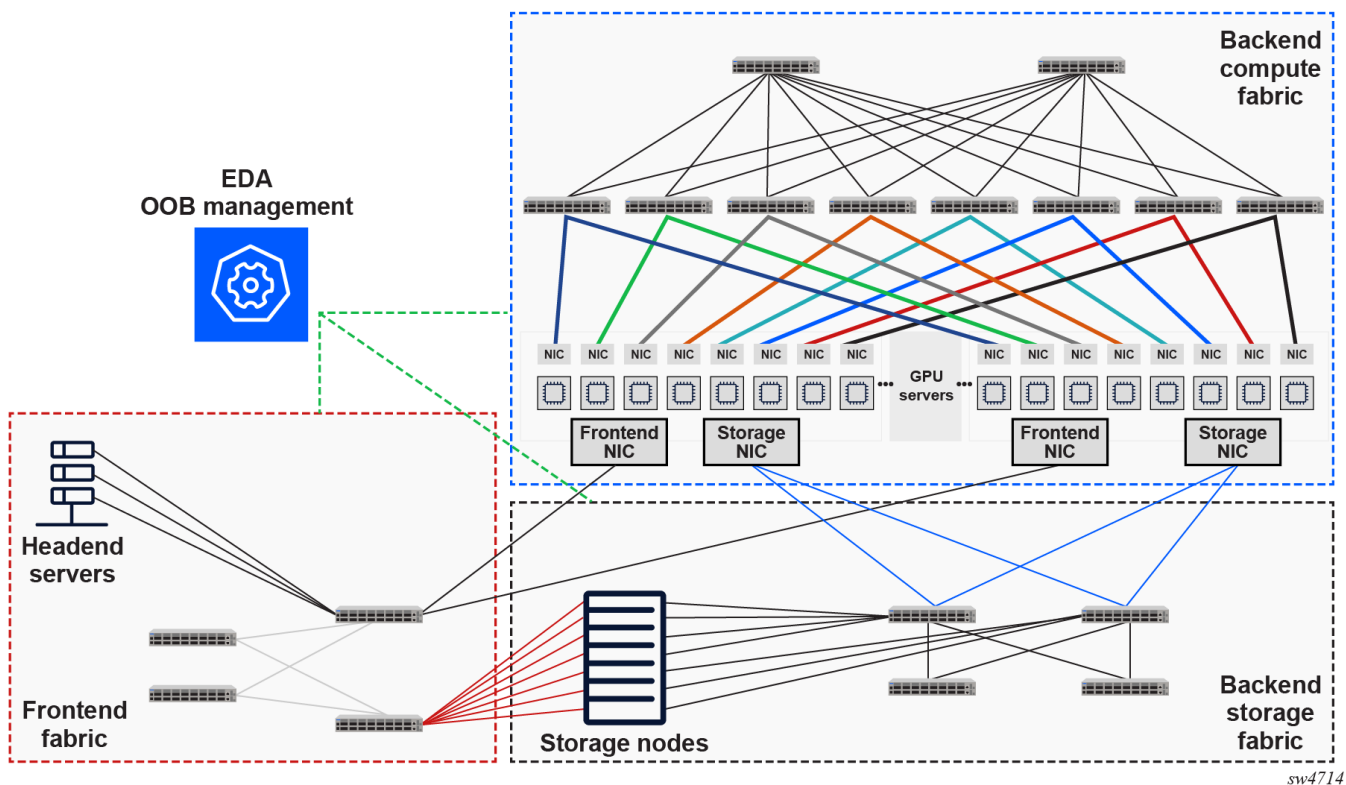


Figure 7. Components of an AI/ML cluster

Backend compute fabric is fabric that hosts the GPU servers.

The GPU direct ports on the server, which is usually equal to the number of GPUs in the higher-end training servers (such as the AMD Instinct MI300X and comparable Nvidia servers, such as the H200) 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, and the headend along with MCP serves and RAG gateways.

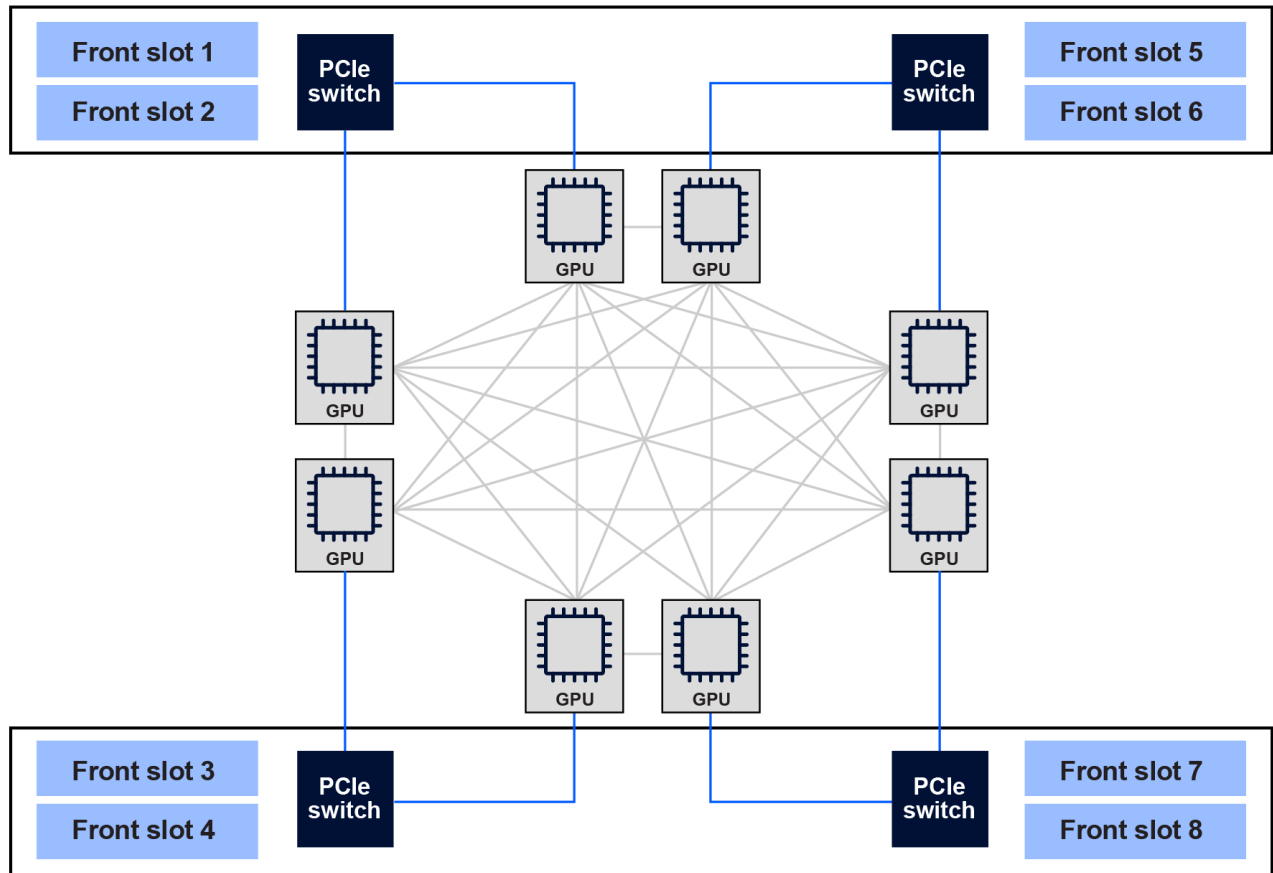
The front-end fabric is also connected to the dedicated storage portions to manage volumes and checkpoints.

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/PURE storage are examples of dedicated storage node vendors. The number of 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

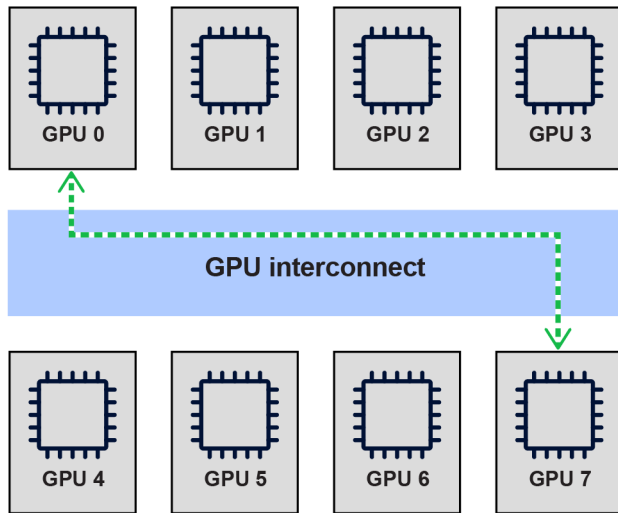


sw4716

Figure 8. Internal network architecture of a GPU server

Figure 8 shows the GPU switching reference architecture for the SMC1 GPU server capable of hosting the AMD MI300X-series GPU.

As shown, the GPUs have a full mesh connectivity with each other via the infinity fabric, which means every GPU in the server has 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.



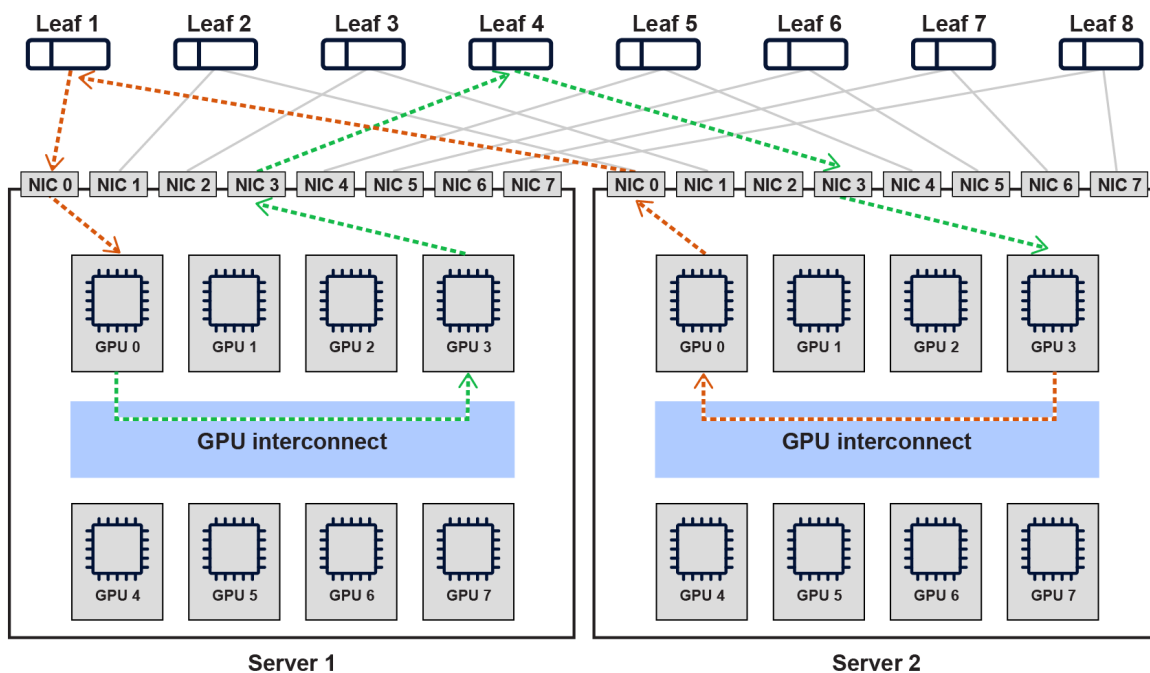
sw4715

Figure 9. Intra-node communication

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

Figure 9 shows that each AMD MI300X 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 Infinity fabric or its evolution “Accelerated fabric link” and switch it to GPU7.

2.5.2 Inter-node communication



sw4717

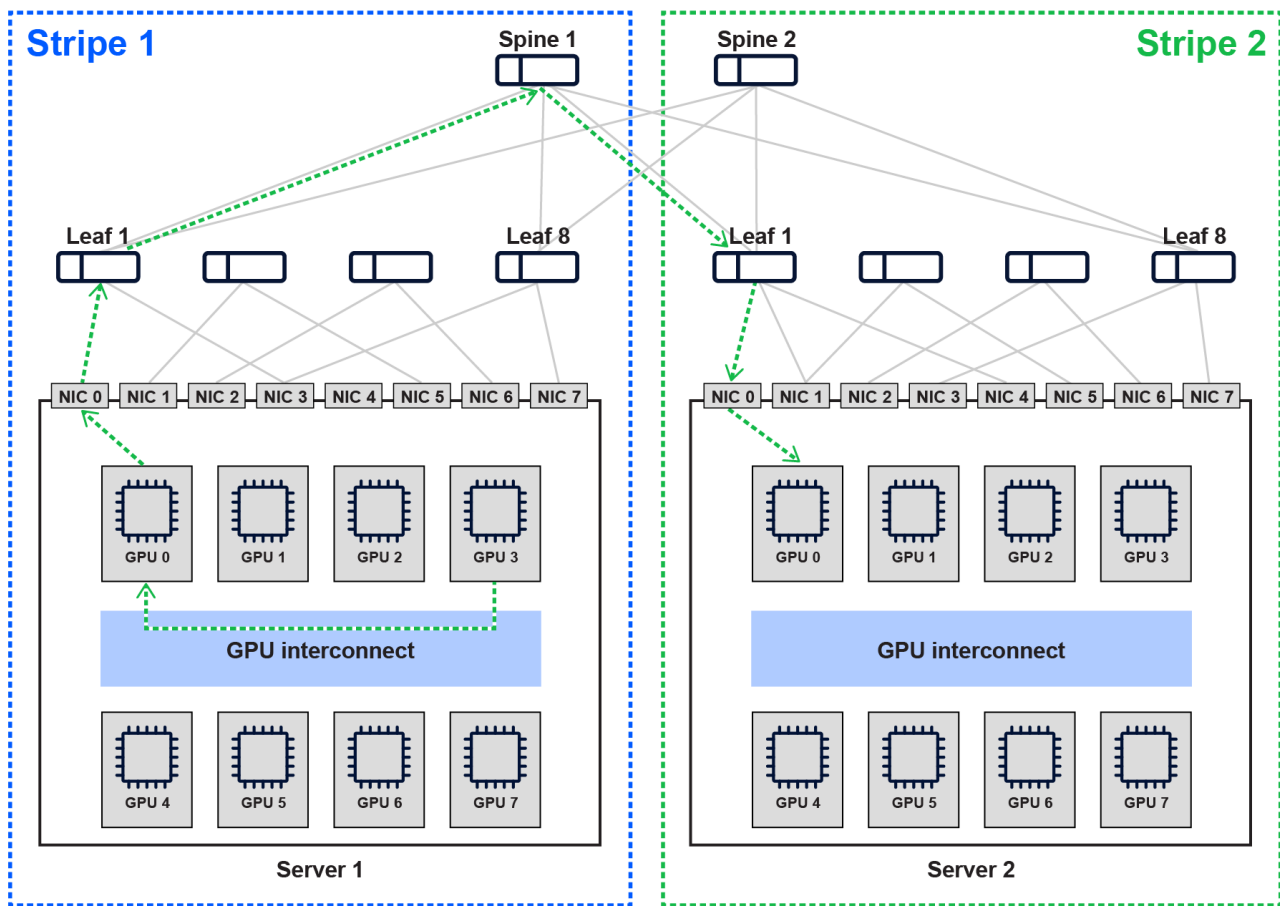
Figure 10. Inter-node communication

Inter-node communication speaks about 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 then 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 then to GPU 0 on Server 1, as indicated by the orange path.

2.5.3 Inter-stripe communication



sw4718

Figure 11. Inter-stripe communication

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

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 nature of the learning model and applications. Some of these scenarios have been listed below.

2.6.1.1 Scenario 1: Intra-rail simultaneous microburst

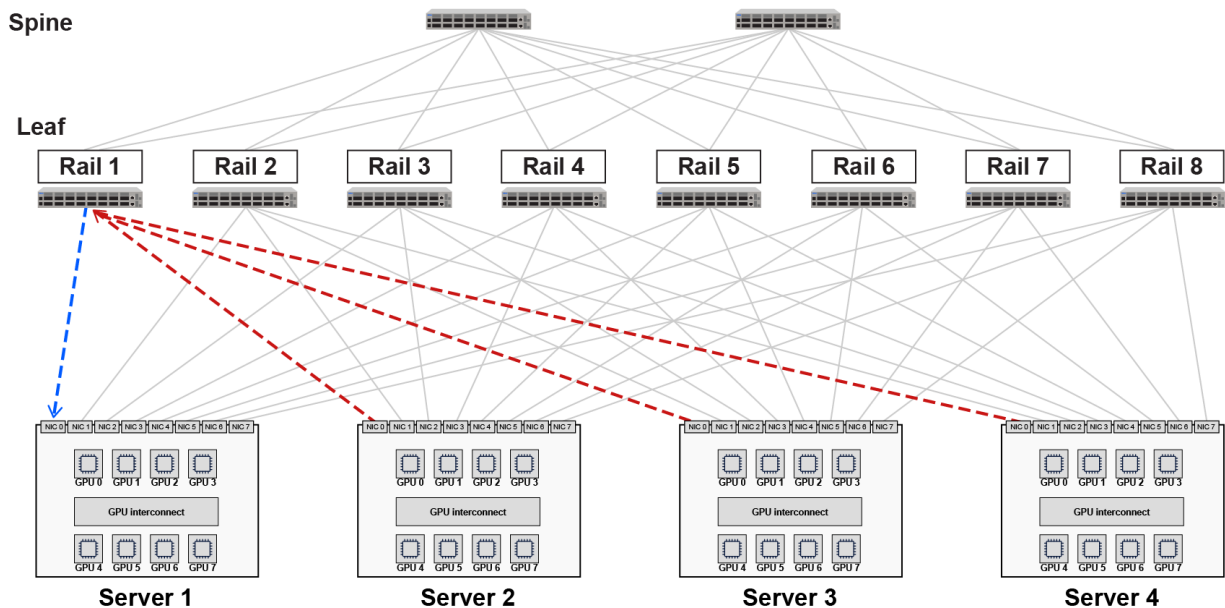


Figure 12. Intra-rail intra-stripe congestion

sw4719

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.

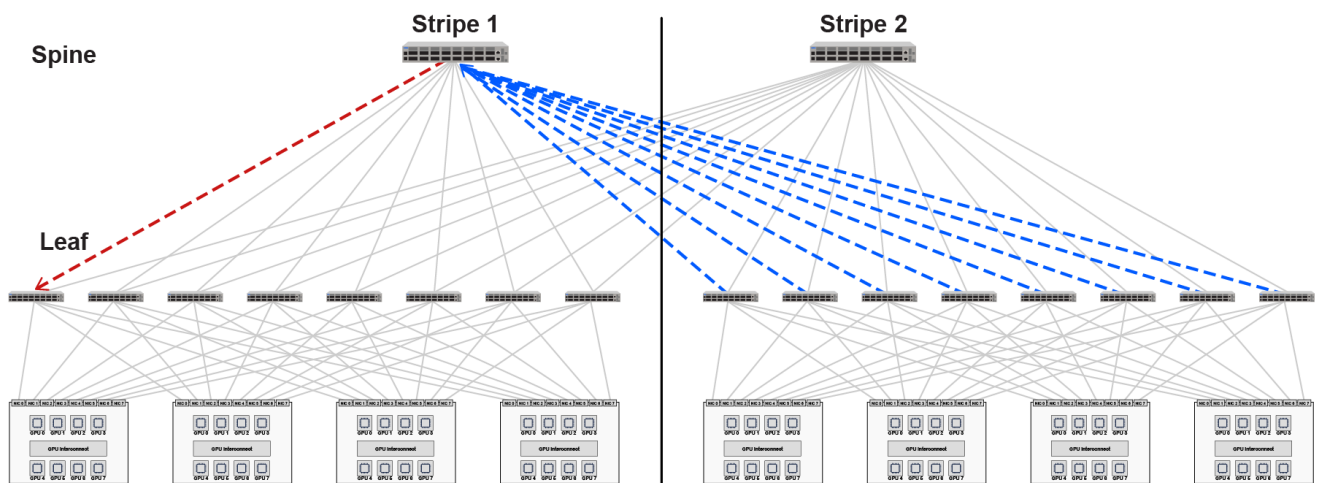


Note: This communication is done via RoCEv2 for Ethernet-based fabrics.

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

If all GPU 0s decide to synchronize with GPU 0 of Server 1 at the same time, and send 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

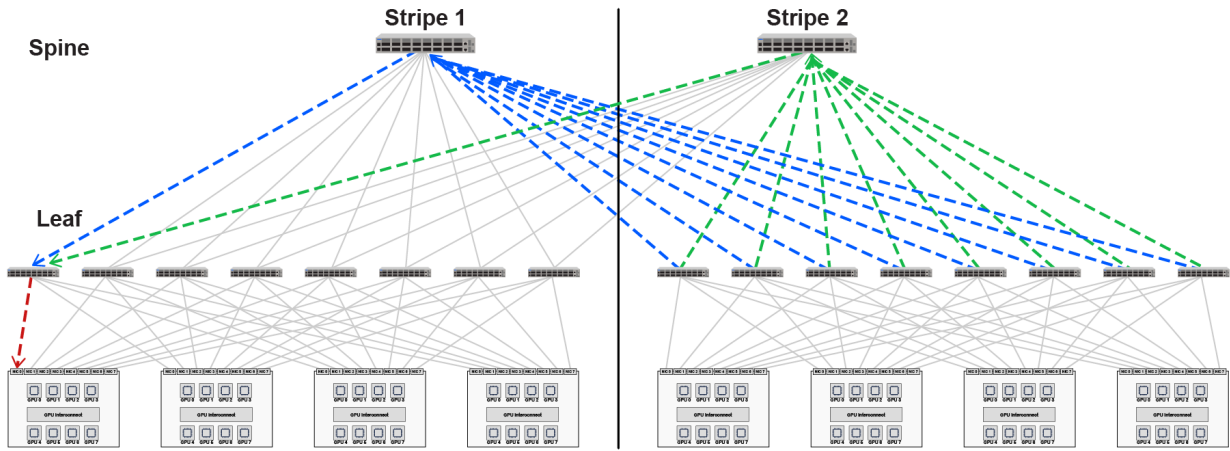


sw4720

Figure 13. Inter-stripe spine polarization

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 it 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

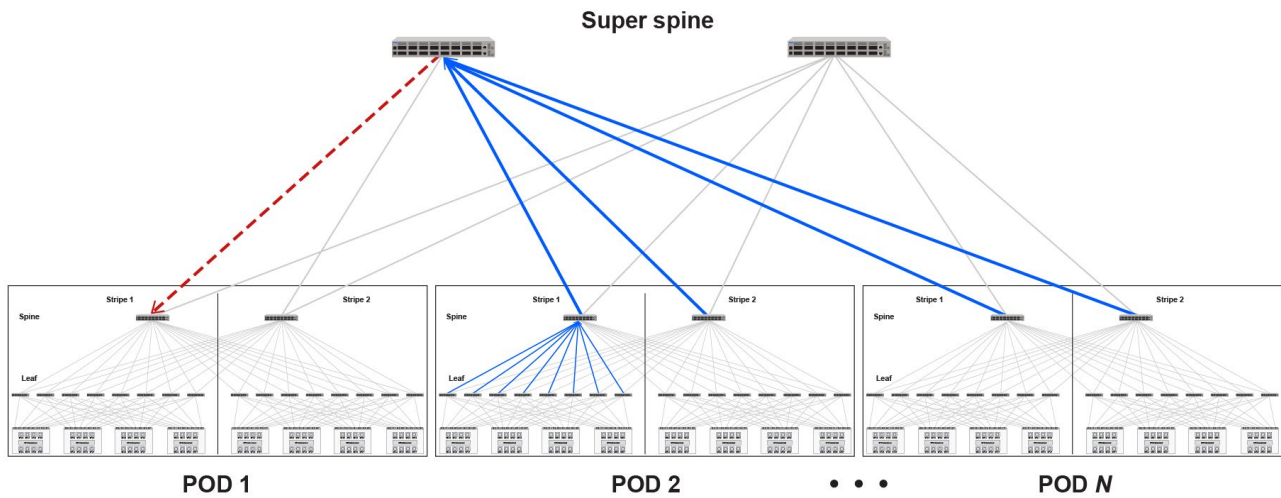


sw4721

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



sw4722

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; this causes a scenario where the rate of incoming traffic to a spine can exceed the rate at which the spine can send it out, and this can cause congestion



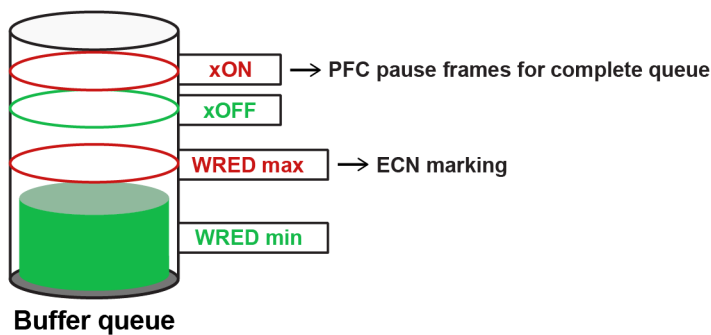
Note: Many probable congestion scenarios can be avoided by using non-blocking architectures and maintaining an oversubscription ratio of 1.

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.

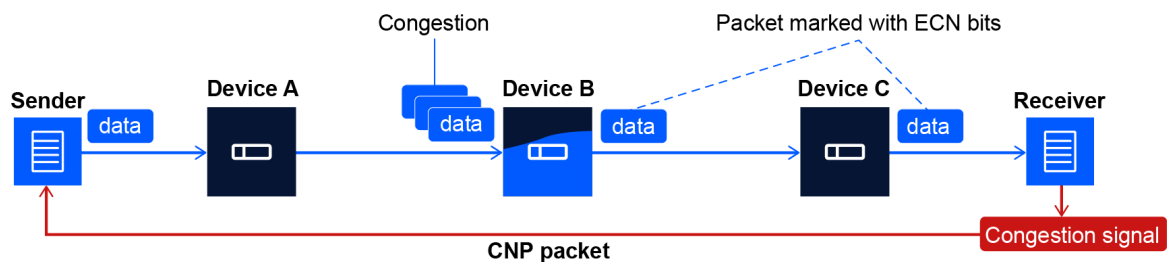


Buffer queue

sw4724

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

2.6.2.1 Explicit Congestion Notification



sw4725

Figure 17. ECN workflow

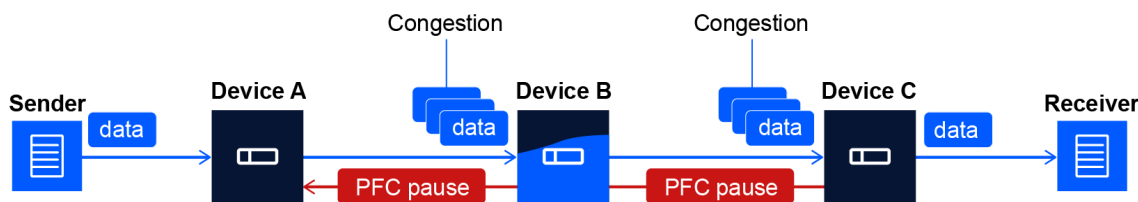
Explicit Congestion Notification (ECN) RFC 3168 facilitates end-to-end congestion management without dropping packets and is ideal for lossless AI fabrics.

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



sw4726

Figure 18. PFC workflow

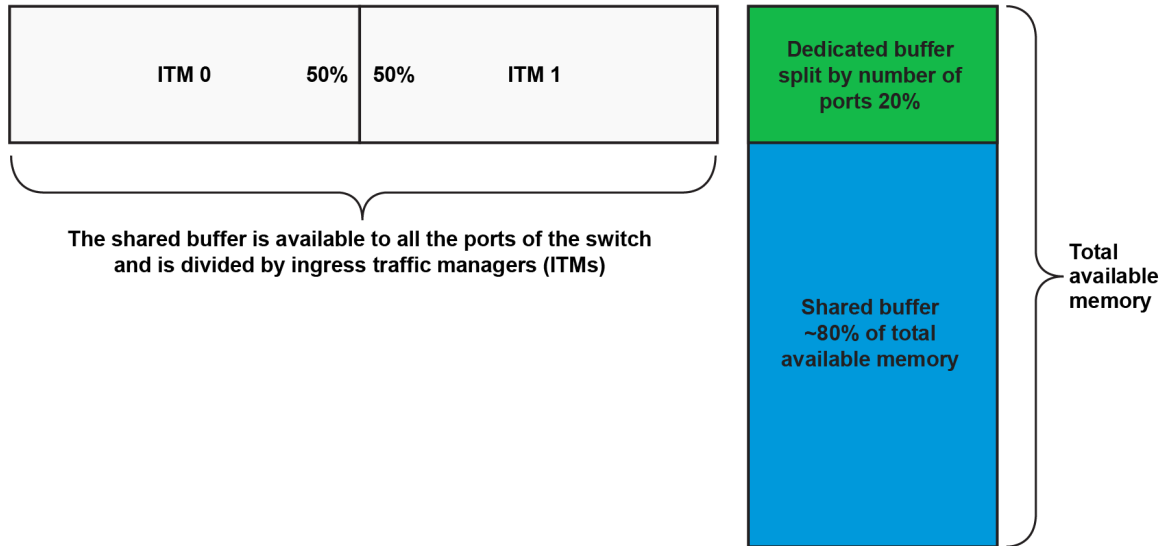
PFC, unlike ECN, works on a per-segment basis. As shown in Figure 18, data is being sent from the sender to the receiver via network Devices A, B, and C. Note that Devices B and C are facing congestion. PFC pause frames will be sent from Device C to B, and from Device B to A. If either Device C or B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



sw4723

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.

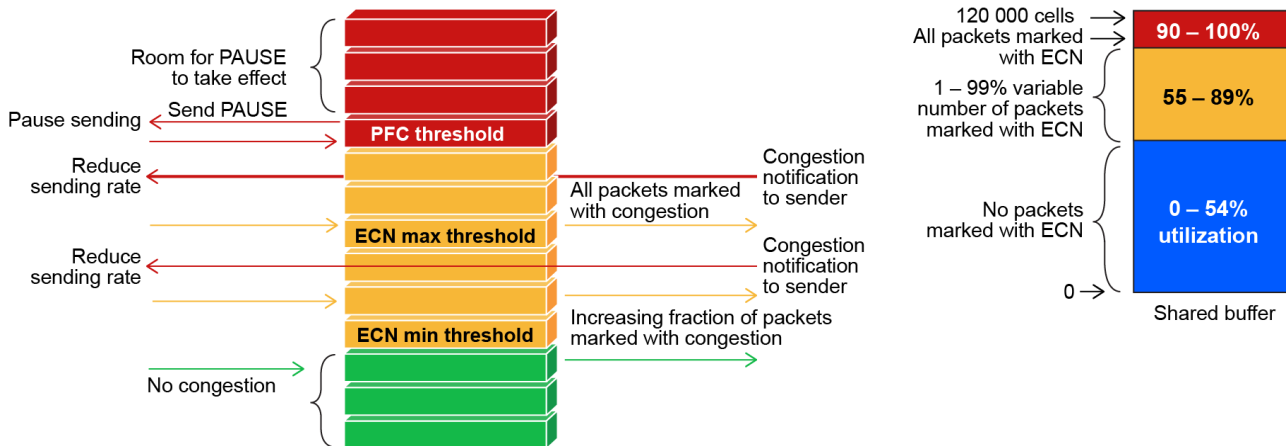
The total available memory of the router is split into two kinds of buffers: the dedicated buffer and the shared buffer. The dedicated buffer is then further allocated equally to all available ports.

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:

$40 \text{ mb (per ITM)} / 16 \text{ ports} \times (2 \text{ queues per port})$



sw4727

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

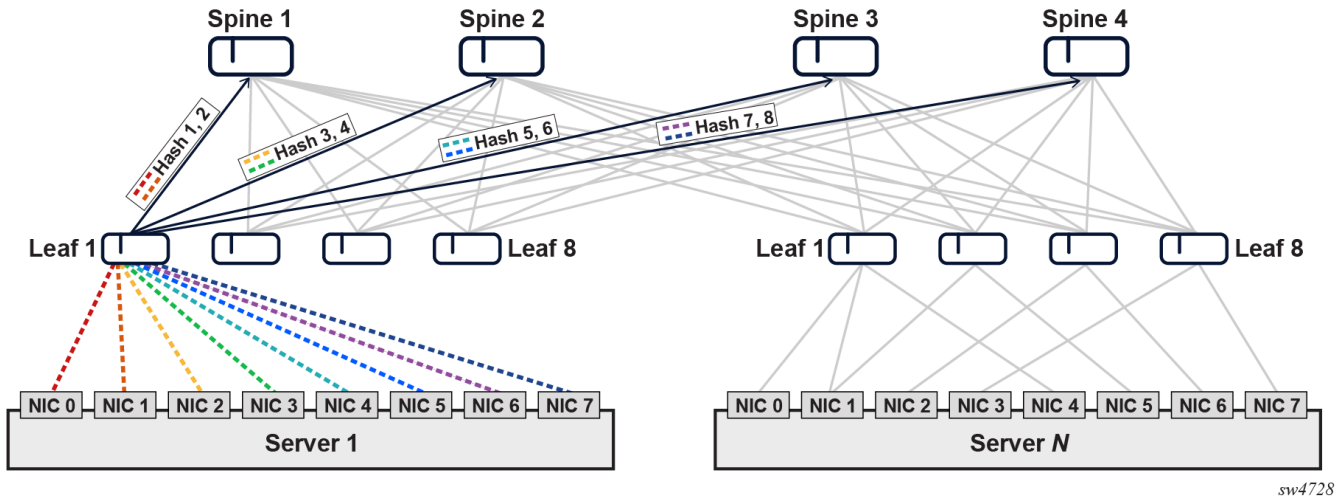


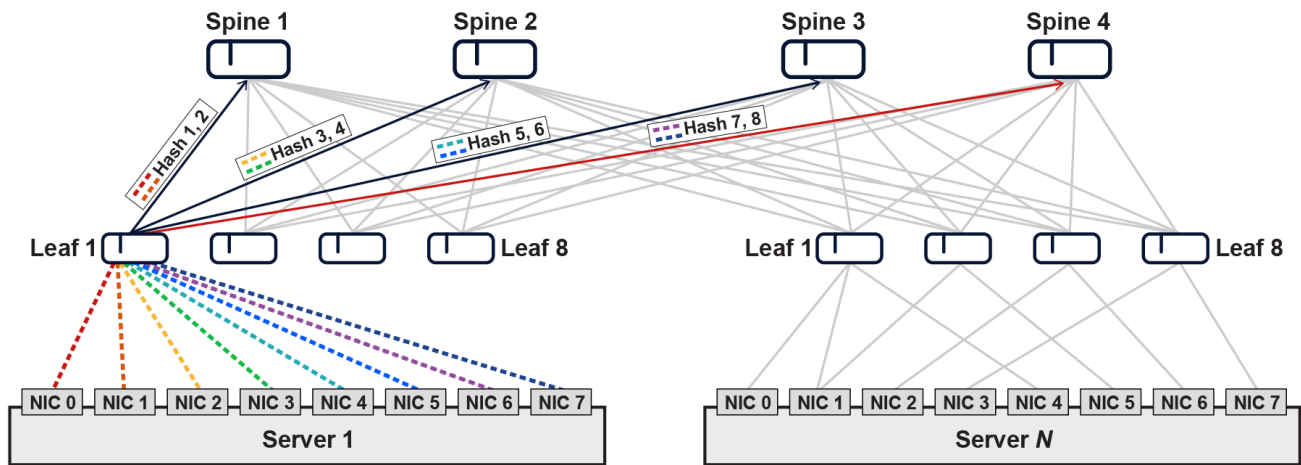
Figure 21. Static ECMP hash-based load balancing – normal state

sw4728

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, 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 the course of 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.



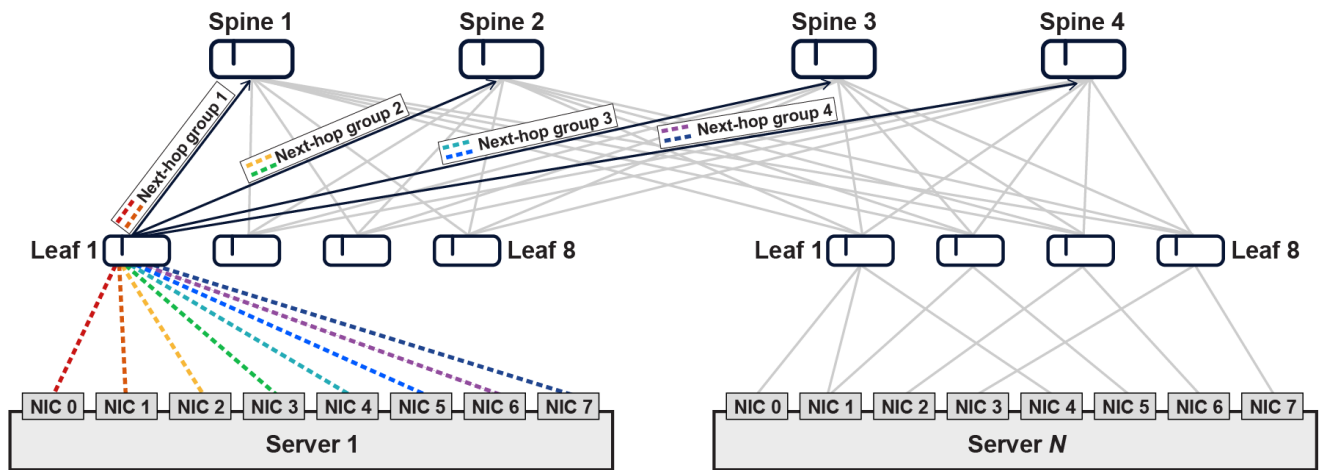
sw4729

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.



sw4730

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.

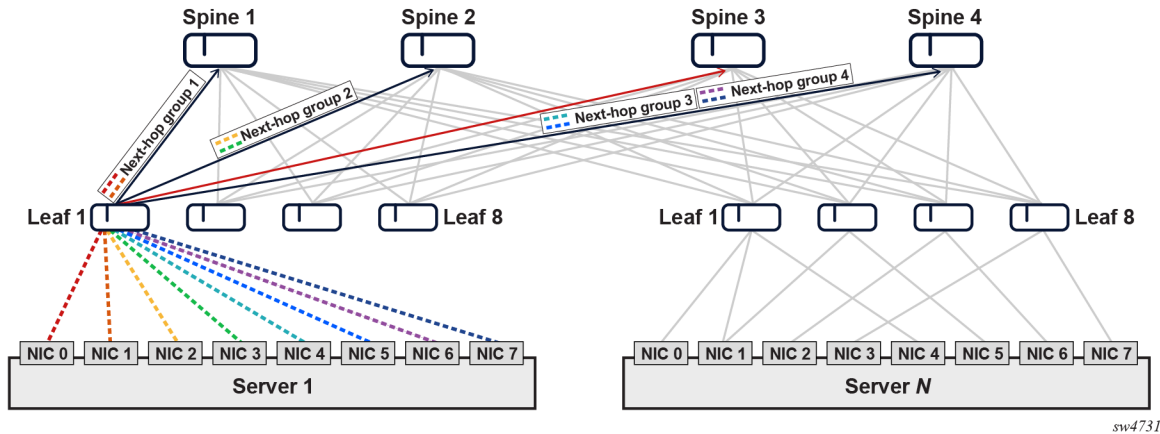


Figure 24. DLB – congested state

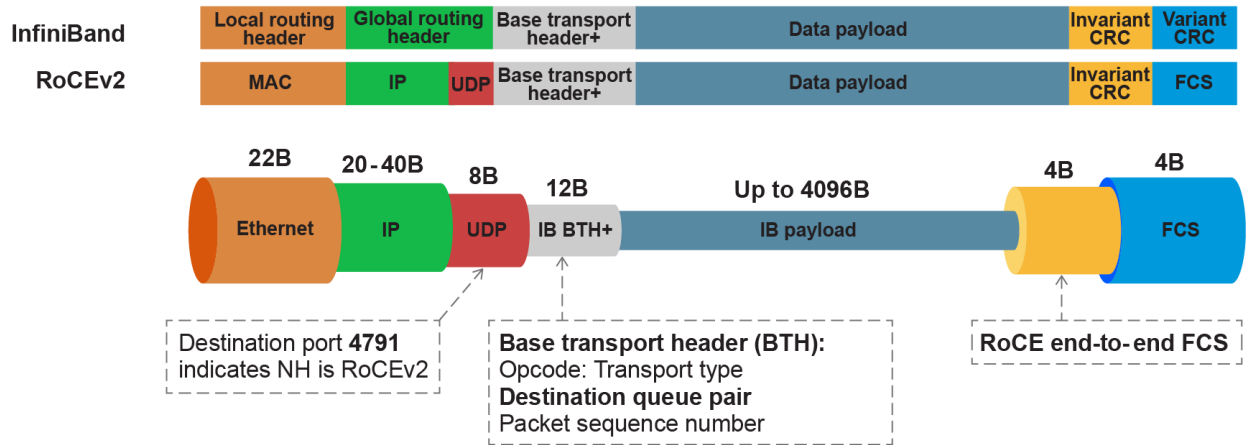
When the link becomes congested, the sub flow idle time (the time interval between flows in a flow group) exceeds the inactivity timer, and this makes it eligible for reassignment to another member link. See section 8.1.4 for examples.



Note: If the inactivity timer is not optimized to the environment and traffic pattern, it results in the flows jumping between members 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



sw4732

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.

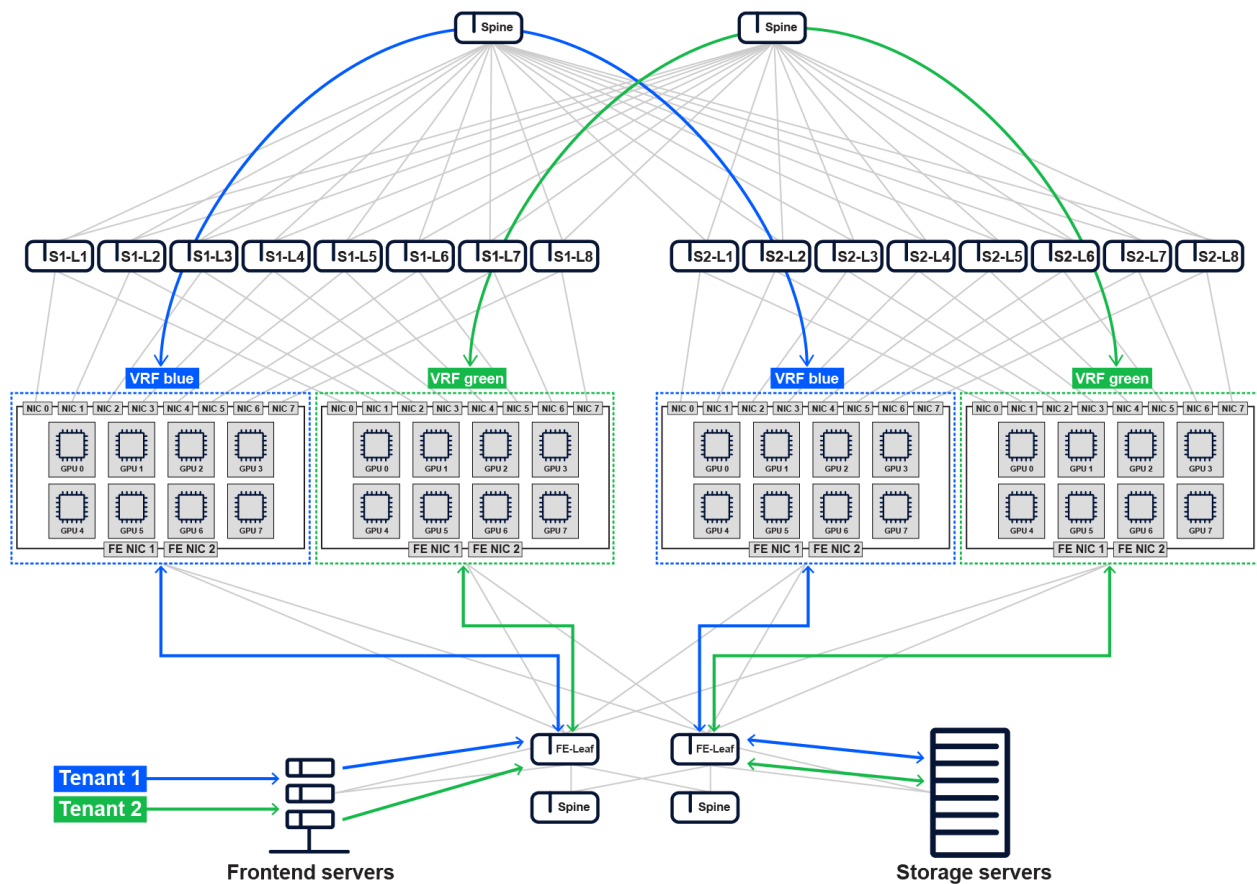
To add more variance into each flow, with quadratic probing (QP) hashing the queue pairs are also added to the hash calculation such that even if the source and the destination are the same for two flows, the calculation generates a different hash based on the send and receive queues. QP hashing helps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



sw4733

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 or not, 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

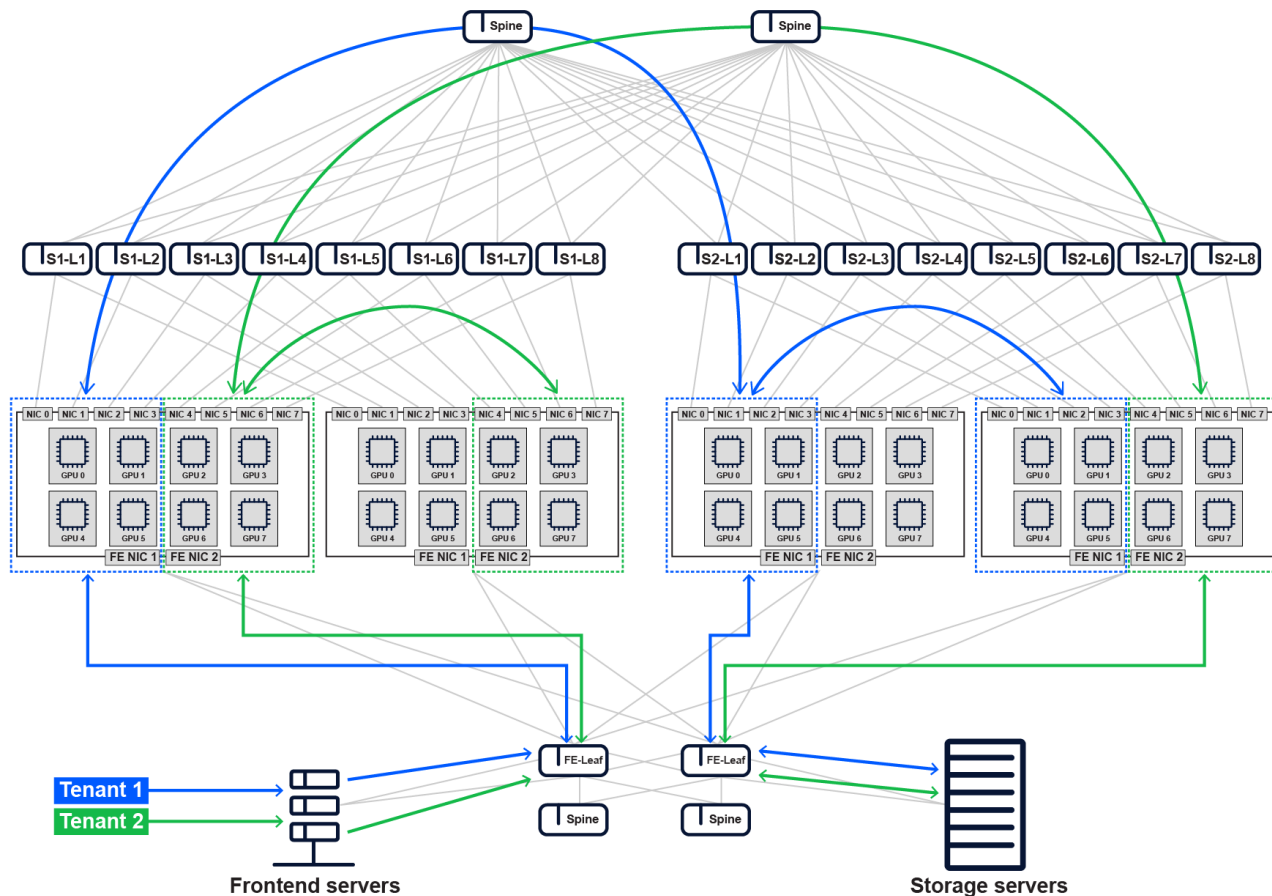


Figure 27. GPU isolation

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 must be disabled, and the optimization must 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.

sw4734



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

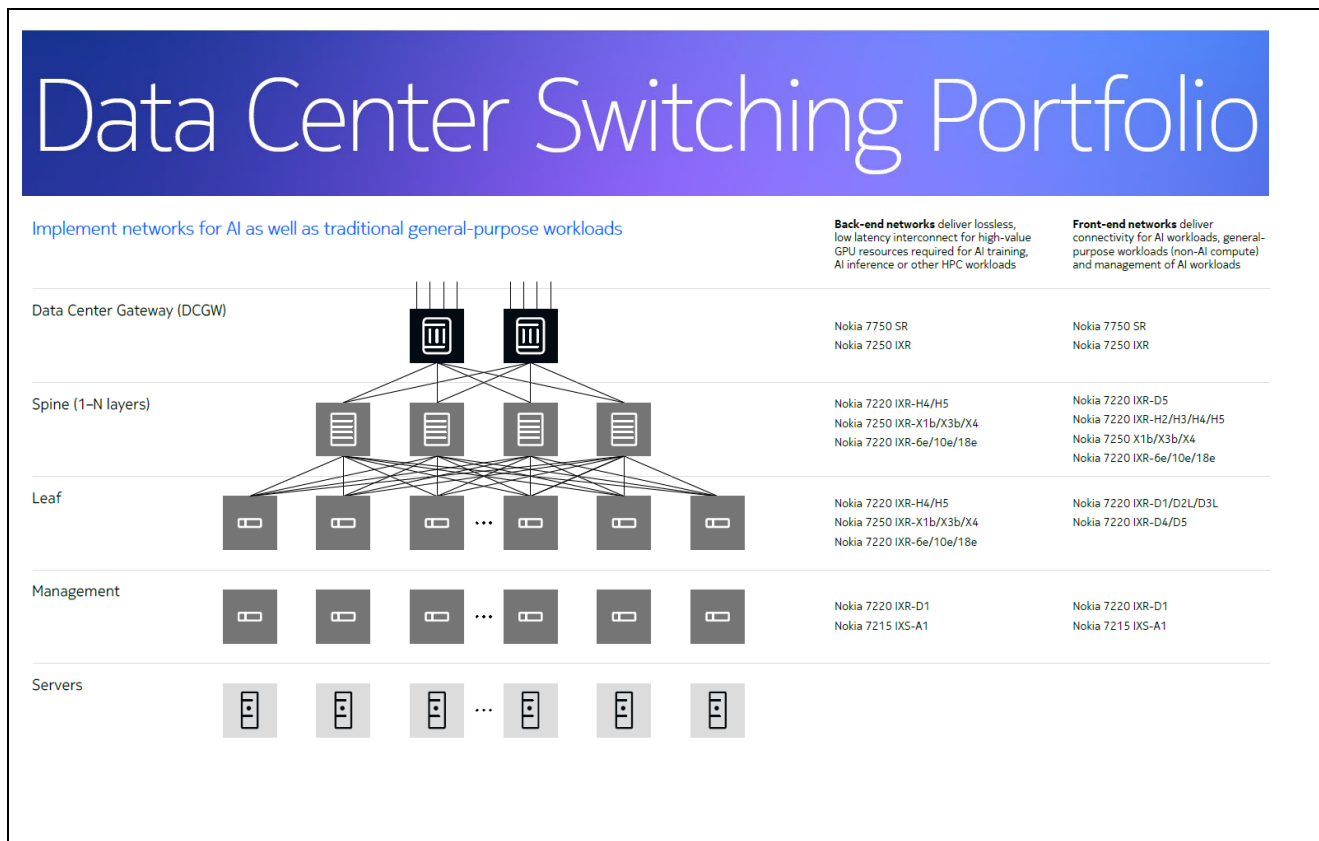
The table below shows the hardware that has been used in the validation process. The selection of optics was determined by laboratory availability.

| Connection | | | | Connectivity option |
|------------|---|---------------------------|-----------|---|
| Endpoints | | Speed | Port type | DAC, AOC, or transceiver and cable |
| 1 | A | AOC-S400G-B1C BCM57608 | 400G | QSFP-DD Transceiver: <ul style="list-style-type: none"> 3HE21007AARA01 QSFPDD 400G-SR4 Cable: MPO-12 |
| | B | 7220 IXR-H4-32D | 400G | QSFP-DD Transceiver: <ul style="list-style-type: none"> 3HE21007AARA01 QSFPDD 400G-SR4 Cable: MPO-12 |
| 2 | A | AOC-S400G-B1C BCM57608 | 400G | QSFP-DD Transceiver: <ul style="list-style-type: none"> 3HE15272AARA01 QSFPDD 400G-FR4 Cable: LC SM fiber |
| | B | 7220 IXR-H4-32D | 400G | QSFP-DD Transceiver: <ul style="list-style-type: none"> 3HE15272AARA01 QSFPDD 400G-FR4 Cable: LC SM fiber |
| 3 | A | AOC-S400G-B1C BCM57608 | 400G | QSFP-DD Transceiver: <ul style="list-style-type: none"> 3HE15271AARA01 QSFPDD 400G-DR4 Cable: MPO-12 |

| | | | | | |
|----|---|---------------------------|------|---------|---|
| | B | 7220 IXR-H4-32D | 400G | QSFP-DD | Transceiver: <ul style="list-style-type: none"> • 3HE15271AARA01 • QSFDD 400G-DR4 Cable: MPO-12 |
| 4 | A | AOC-S400G-B1C BCM57608 | 400G | QSFP-DD | AOC: <ul style="list-style-type: none"> • Eoptolink • EOLD-8HG-PCT-05C1 • 800G QSFPDD to 2 x 400G QSFP112 Breakout AOC 5m |
| | B | 7220 IXR-H5-32D | 800G | QSFP-DD | |
| 5 | A | AOC-S400G-B1C BCM57608 | 400G | QSFP-DD | AOC: <ul style="list-style-type: none"> • Eoptolink • EOLD-8HG-PCT-05C1 • 800G QSFPDD to 2 x 400G QSFP112 Breakout AOC 5m |
| | B | 7220 IXR-H5-64D | 800G | QSFP-DD | |
| 6 | A | 7220 IXR-H4-32D | 400G | QSFP-DD | Transceiver: <ul style="list-style-type: none"> • 3HE21007AARA01 • QSFDD 400G-SR4 Cable: MPO-12 |
| | B | 7220 IXR-H5-32D | 400G | QSFP-DD | Transceiver: <ul style="list-style-type: none"> • 3HE21007AARA01 • QSFDD 400G-SR4 Cable: MPO-12 |
| 7 | A | 7220 IXR-H4-32D | 400G | QSFP-DD | Transceiver: <ul style="list-style-type: none"> • Eoptolink • EOLD-854HG-01M46 • 400G-VR4 Cable: MPO-12 |
| | B | 7220 IXR-H5-640 | 800G | OSFP | Transceiver: <ul style="list-style-type: none"> • Eoptolink • EOLO-858HG-01-D • 2 x 400G-VR4 OSFP Cable: MPO-12 |
| 8 | A | 7220 IXR-H5-32D | 800G | QSFP-DD | DAC: <ul style="list-style-type: none"> • LUXSHARE-TECH • LQ8DD020-SD-R • 800G DAC |
| | B | 7220 IXR-H5-32D | 800G | QSFP-DD | |
| 9 | A | 7220 IXR-H5-32D | 800G | QSFP-DD | AOC: <ul style="list-style-type: none"> • Eoptolink • EOLO-8HG-PCT-05G • 800G OSFP to QSFP-DD 5m |
| | B | 7220 IXR-H5-640 | 800G | OSFP | |
| 10 | A | 7220 IXR-D5 | 400G | QSFP-DD | Transceiver: <ul style="list-style-type: none"> • Eoptolink • EOLD-854HG-01M46 • 400G-VR4 Cable: MPO-12 |
| | B | 7220 IXR-H5-640 | 800G | OSFP | Transceiver: <ul style="list-style-type: none"> • Eoptolink |













| | | | | | |
|----|---|------------------------------------|------|-----------|--|
| | | | | | <ul style="list-style-type: none"> EOLO-858HG-01-D 2 x 400G-VR4 OSFP Cable: MPO-12 |
| 11 | A | AOC-S200G-B2C BCM57608 | 200G | QSFP56 | AOC: <ul style="list-style-type: none"> Eoptolink EOLD-4HG-PCT-05C45 400G QSFP56-DD to 2 x 200G QSFP56 5m |
| | B | 7220 IXR-D5 | 400G | QSFP56-DD | |
| 12 | A | AOC-CX7AH0078-DTZ | 200G | QSFP112 | DAC: <ul style="list-style-type: none"> Amphenol NDYRYH-C105 400G QSFP-DD to 2 x 200G QSFP-DD |
| | B | 7220 IXR-D5 | 400G | QSFP-DD | |
| 13 | A | Intel Ethernet Controller XL710 | 40G | QSFP+ | Transceiver: <ul style="list-style-type: none"> 3HE07928AAAA01 QSFP+ 40G-SR4 Cable: MPO-12 |
| | B | 7220 IXR-D5 | 40G | QSFP+ | |

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)

| | | | | |
|---|---|---|--|--|
|  <p>7220 IXR-D1</p> <ul style="list-style-type: none"> • 88 Gb/s (FD); fixed; 1RU • 4 x 10G SFP+ • 48 x 10/100/1000M RJ45 |  <p>7220 IXR-D2L</p> <ul style="list-style-type: none"> • 2.0 Tb/s (FD); fixed; 1RU • 8 x 100G QSFP28 • 48 x 25G SFP28 • 2 x 10G SFP+ |  <p>7220 IXR-D3L</p> <ul style="list-style-type: none"> • 3.2 Tb/s (FD); fixed; 1RU • 32 x 100G QSFP28 • 2 x 10G SFP+ |  <p>7220 IXR-D4</p> <ul style="list-style-type: none"> • 6.0 Tb/s (FD); fixed; 1RU • 8 x 400G QSFP-DD • 28 x 100G QSFP28 |  <p>7220 IXR-D5</p> <ul style="list-style-type: none"> • 12.8 Tb/s (FD); fixed; 1RU • 32 x 400G QSFP-DD • 2 x 10G SFP+ |
|  <p>7220 IXR-H2</p> <ul style="list-style-type: none"> • 12.8 Tb/s (FD); fixed; 4RU • 128 x 100G QSFP28 |  <p>7220 IXR-H3</p> <ul style="list-style-type: none"> • 12.8 Tb/s (FD); fixed; 1RU • 32 x 400G QSFP-DD • 2 x 10G SFP+ |  <p>7220 IXR-H4-32D</p> <ul style="list-style-type: none"> • 12.8 Tb/s (FD); fixed; 1RU • 32 x 400G QSFP-DD • 1 x 10G SFP+ |  <p>7220 IXR-H4</p> <ul style="list-style-type: none"> • 25.6 Tb/s (FD); fixed; 2RU • 64 x 400G QSFP-DD • 2 x 10G SFP+ |  <p>7220 IXR-H5-32D</p> <ul style="list-style-type: none"> • 25.6 Tb/s (FD); fixed; 1RU • 32 x 800G OSFP • 2 x 10G SFP+ |
|  <p>7220 IXR-H5-64D</p> <ul style="list-style-type: none"> • 51.2 Tb/s (FD); fixed; 2RU • 64 x 800G QSFP-DD • 2 x 10G SFP+ |  <p>7220 IXR-H5-64O</p> <ul style="list-style-type: none"> • 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)

| | | | | |
|---|---|---|---|--|
|  <p>7250 IXR-X1b</p> <ul style="list-style-type: none"> • 7.2 Tb/s (FD); fixed; 1RU • 24 x 100G QSFP28 • 12 x 400G QSFP-DD |  <p>7250 IXR-X4 QSFP-DD</p> <ul style="list-style-type: none"> • 25.6 Tb/s (FD); fixed; 1RU • 32 x 800G QSFP-DD |  <p>7250 IXR-6e</p> <ul style="list-style-type: none"> • 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 |  <p>7250 IXR-10e</p> <ul style="list-style-type: none"> • 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 |  <p>7250 IXR-18e</p> <ul style="list-style-type: none"> • 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 |
|  <p>7250 IXR-X3b</p> <ul style="list-style-type: none"> • 14.4 Tb/s (FD); fixed; 1RU • 36 x 400G QSFP-DD |  <p>7250 IXR-X4 OSFP</p> <ul style="list-style-type: none"> • 25.6 Tb/s (FD); fixed; 1RU • 32 x 800G OSFP | | | |

Management Switches

- Fixed-configuration 7218 Interconnect System (IXS) platform designed for data center fabric management connectivity
- Integrated redundant fans and PSUs
- Powered by SR Linux and community SONiC



7218 IXS-A1

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

Service Router 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® open-source NOS leverages the strength of a large ecosystem and community
- Offers 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

Explore the Nokia Data Center Fabric portfolio

Explore the Nokia IP Routing portfolio

Data center networks design hub

- Access to a rich library of resources
- Nokia Validated Designs (NVDs) are rigorously tested for reliable deployment
- Reference designs deliver alternate architectures and product capabilities
- Technical briefs provide clear insights on complex networking technologies

5 Reference architecture and network orchestration

5.1 High-level overview

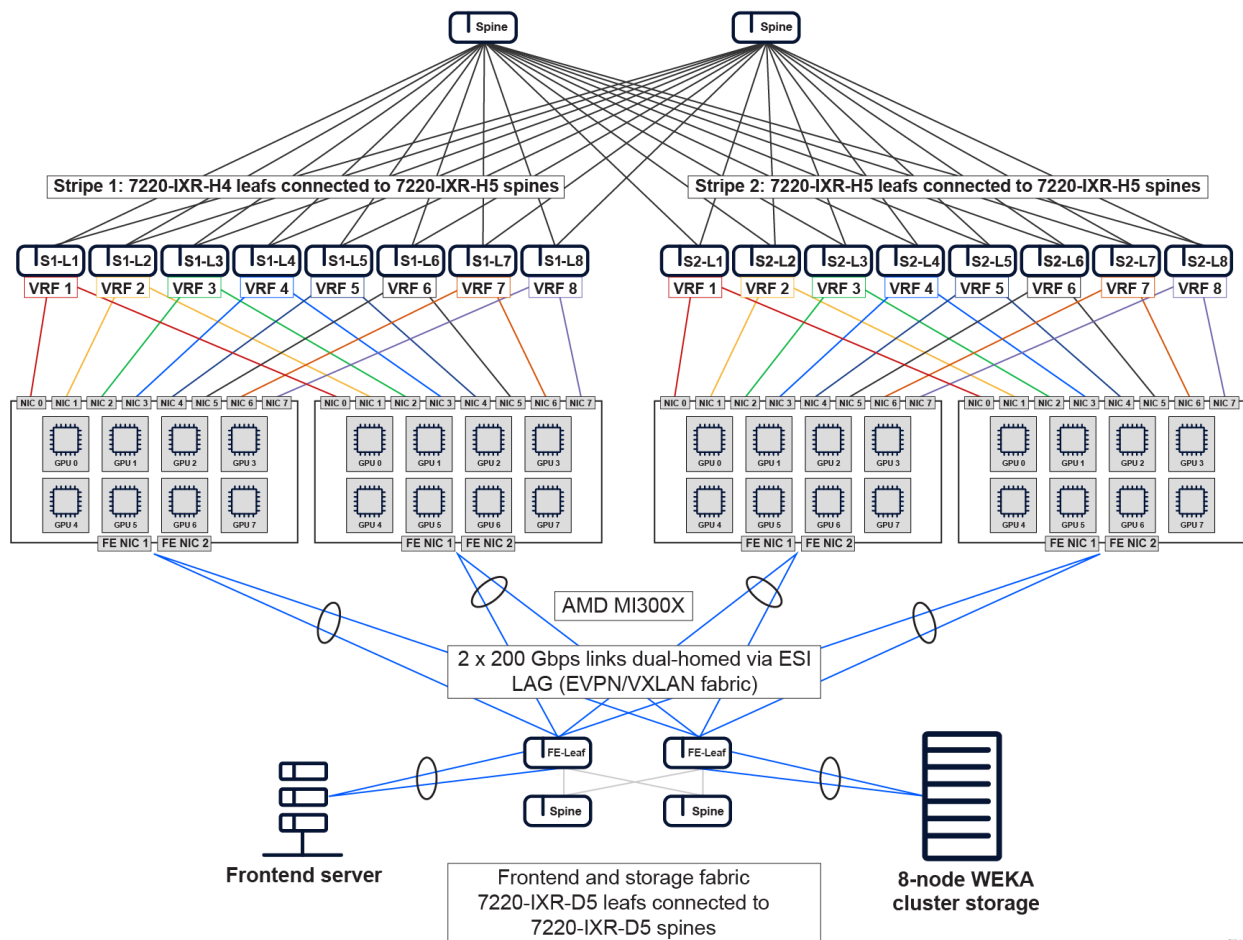


Figure 28. HLD – Nokia hybrid training and inference cluster design

sw4735

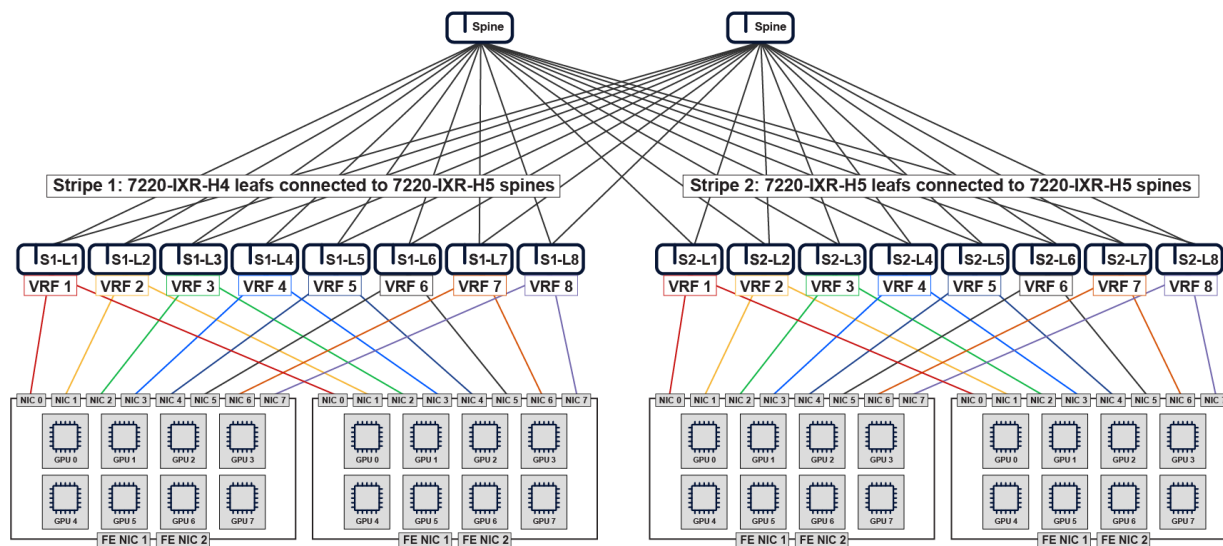
Figure 28 shows the reference architecture for the Nokia Validated Design that is described in this document. It is a rail-optimized, two-stripe design hosting four SMCI servers that have eight AMD MI300X GPUs per server.

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-D5 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



sw4737

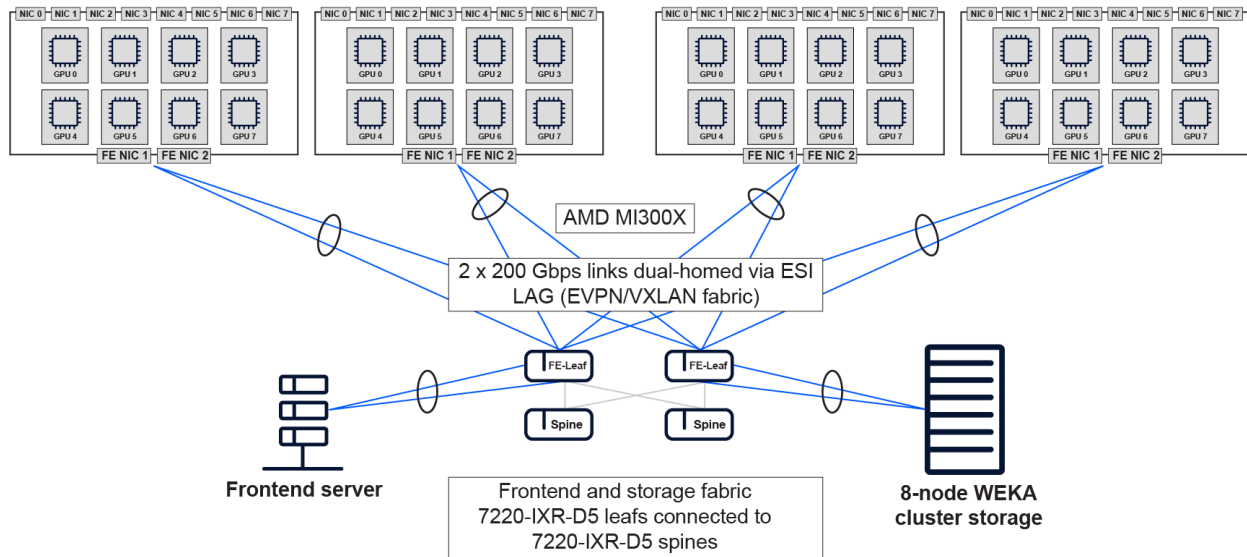
Figure 29. Backend compute fabric

- 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



sw4738

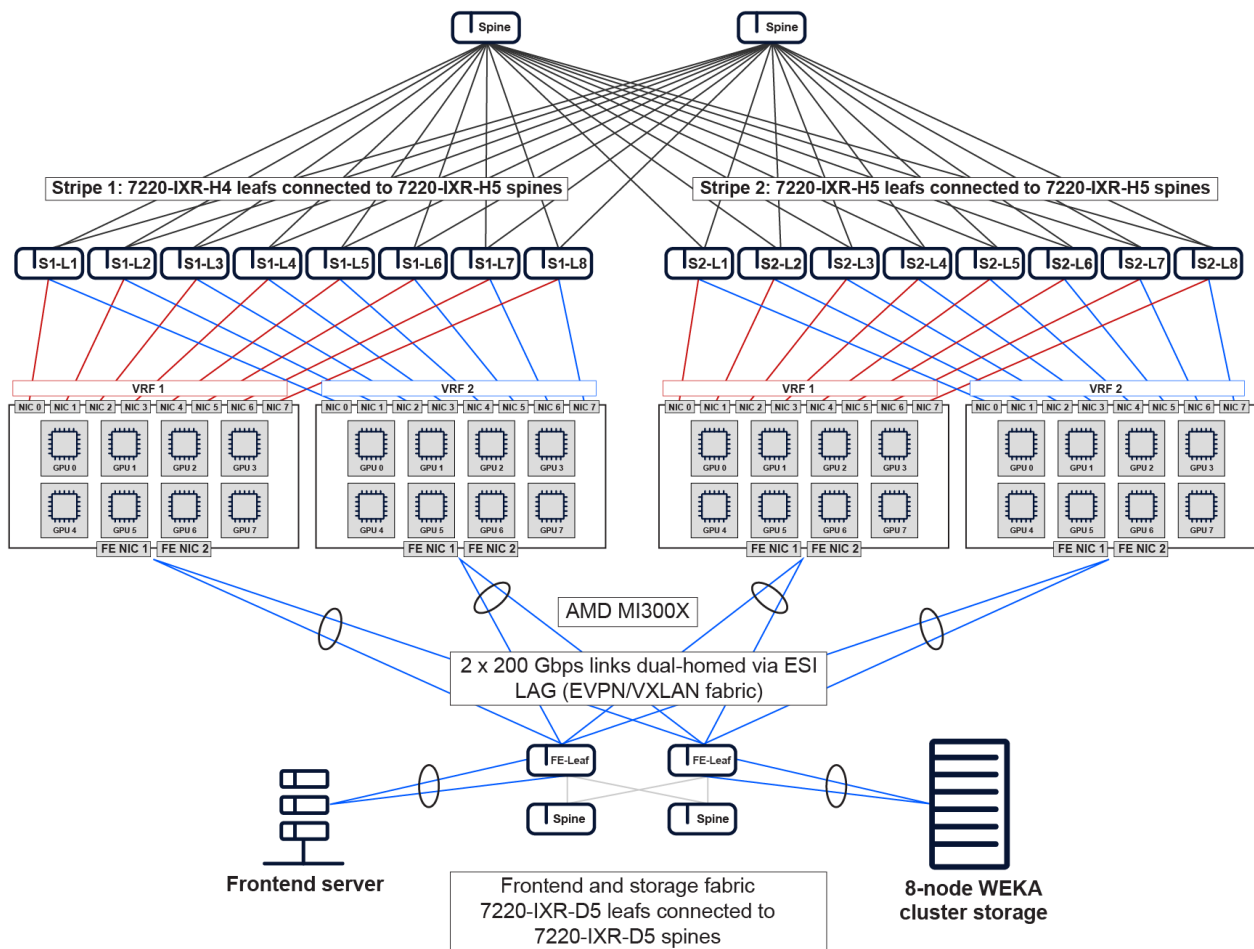
Figure 30. Combined backend storage and frontend fabric

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 leaves.
- 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



sw4736

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

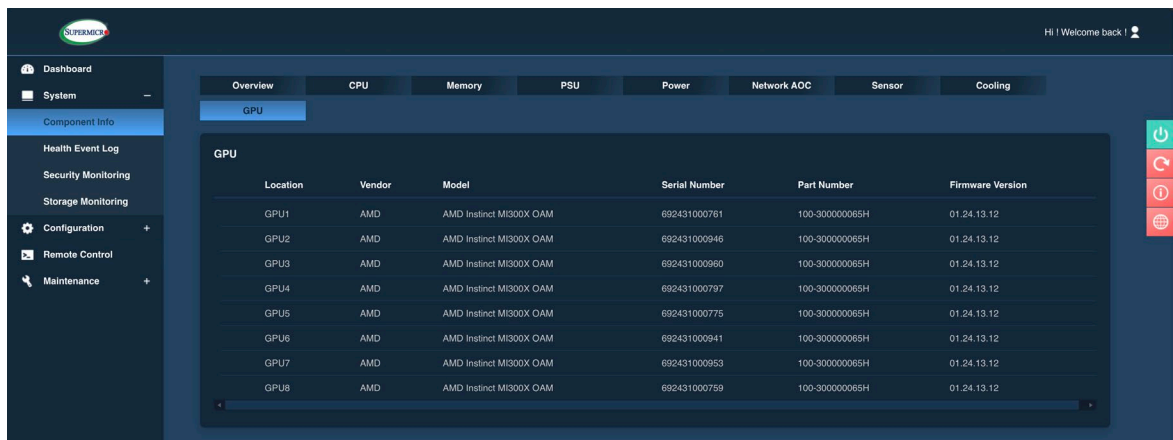
| Component | Specification |
|---------------------|---|
| CPU | 2 x AMD Genoa 9654 (96 cores, 192 threads each) |
| Memory | 24 x 96 GB DDR5-5600 (total 2.3 TB) |
| Disk storage | 2 x Samsung 1.9 TB NVMe drives |
| Networking | 8 x 400G Thor 2 Broadcom NIC 1 x 2-port 200G Broadcom NIC 1 x 2-port 10G RJ45 Intel NIC |
| GPU | 8 x AMD Instinct MI300X |

6.2 SMCI-AMD MI300X server orchestration

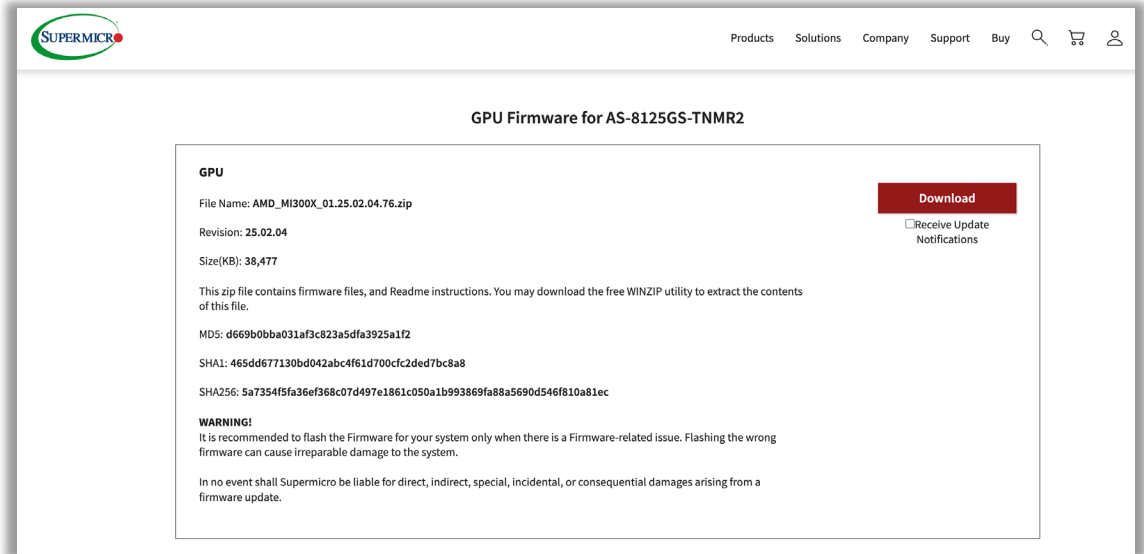
This section outlines the AMD specific orchestration steps that need to be performed in order to have the servers updated to deploy the RCCL collective and workloads for benchmarking.

6.2.1 Upgrade the GPU firmware

1. View the current firmware version on the server manager tool, as shown below.

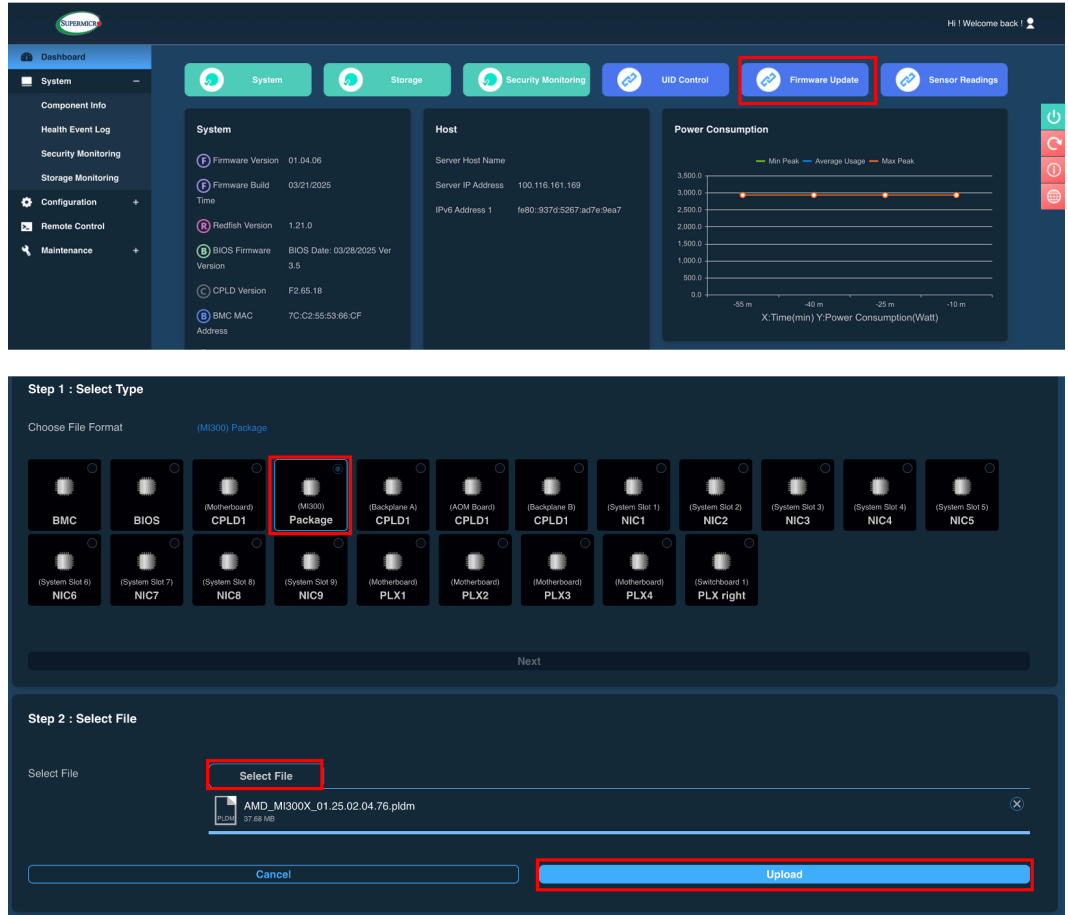


2. Download the latest GPU firmware from the Supermicro website:
<https://www.supermicro.com/en/support/resources/downloadcenter/firmware/AS-8125GS-TNMR2/GPU>



3. From the dashboard shown below, perform the following steps:
 - a. Navigate to the firmware update section.
 - b. Select the GPU package that is required.
 - c. Upload the firmware that was downloaded in the previous step.

d. Perform a cold reboot (shut down the server by cutting power).



6.2.2 AMD Instinct MI300X system optimization

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

<https://instinct.docs.amd.com/projects/amdgpu-docs/en/latest/system-optimization/mi300x.html>

6.2.2.1 BIOS settings

| BIOS setting location [under Advanced tab] | Parameter | Value |
|--|-------------------------------|----------|
| PCIe/PCI/PnP Configuration | Above 4G decoding | Enabled |
| PCIe/PCI/PnP Configuration | SRIOV support | Enabled |
| CPU Configuration | Global C-State Control | Auto |
| NB Configuration | IOMMU | Enabled |
| PCIe/PCI/PnP Configuration | PCIe Ten Bit Tag | Auto |
| NB Configuration → XGMI Configuration | XGMI Link Width Control | Manual |
| NB Configuration → XGMI Configuration | XGMI Force Link Width Control | Force |
| NB Configuration → XGMI Configuration | XGMI Force link width | 2 |
| NB Configuration → XGMI Configuration | XGMI Link Max Speed (auto) | Auto |
| NB Configuration | DF Cstates | Auto |
| NB Configuration | ACS Enable | Disabled |
| PCIe/PCI/PnP Configuration | Re-Size BAR Support | Enabled |

6.2.2.2 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:

- pci=realloc=off**
 With this setting, Linux can unambiguously detect all GPUs of the MI300X-based system because this setting disables the automatic reallocation of PCI resources. It's used when Single Root I/O Virtualization (SR-IOV) Base Address Registers (BARs) have not been allocated by the BIOS. This setting can help avoid potential issues with certain hardware configurations.
- 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.



Note: Though SR-IOV or PCI passthrough are not used for this setup, we followed the best practices provided by AMD documentation.

1. Update GRUB to use the new modified configuration.

```
cse@slate4:~$ cat /etc/default/grub
...
GRUB_CMDLINE_LINUX=" pci=realloc=off iommu=pt
numa_balancing=disable pci=bfsort"
...
```

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
```

6.2.2.3 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 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
```

5. Tune sysfs using the following commands:

```
sudo apt-get install -y linux-tools-common
sudo cpupower idle-set -d 2
```

```
sudo cpupower frequency-set -g performance
```

6.2.3 AMD ROCm installation

ROCm is AMD's open software platform that enables parallel programming for running GPU-accelerated applications, especially in AI, HPC, and data science. ROCm is AMD's alternative to Nvidia's CUDA and is similar in purpose to OpenCL.

Like CUDA, ROCm enables GPU parallelism for high-performance computing and AI workloads. It is an open-source platform and supports heterogeneous programming across CPU and GPUs.

The AMD ROCm 6.4.3 Quick Start Installation Guide can be found at:

<https://rocm.docs.amd.com/projects/install-on-linux/en/docs-6.4.3/install/quick-start.html>

Follow the guide to ensure that the supported OS and kernel are in use.

6.2.3.1 Prerequisites

Confirm that the kernel and OS match the ROCm 6.4.3 system requirements, shown below.

<https://rocm.docs.amd.com/projects/install-on-linux/en/docs-6.4.3/reference/system-requirements.html#supported-operating-systems>

| Operating system | Kernel | Glibc | Support |
|------------------|----------------------|-------|---------|
| Ubuntu 24.04.2 | 6.8 [GA], 6.11 [HWE] | 2.39 | ✓ |
| Ubuntu 22.04.5 | 5.15 [GA], 6.8 [HWE] | 2.35 | ✓ |
| RHEL 9.6 | 5.14+ | 2.34 | ✓ |
| RHEL 9.4 | 5.14+ | 2.34 | ✓ |
| RHEL 8.10 | 4.18.0+ | 2.28 | ✓ |
| SLES 15 SP7 | 6.11.0+ | 2.38 | ✓ |
| SLES 15 SP6 | 6.5.0+ | 2.38 | ✓ |
| Oracle Linux 9 | 5.15.0 (UEK) | 2.35 | ✓ [2] |
| Oracle Linux 8 | 5.15.0 (UEK) | 2.28 | ✓ [2] |
| Azure Linux 3.0 | 6.6.60 | 2.38 | ✓ [3] |
| Debian 12 | 6.1 | 2.36 | ✓ [4] |

```
cse@slate2:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.3 LTS
Release:        24.04
Codename:       noble

cse@slate2:~$ uname -smrv
Linux 6.8.0-85-generic #85-Ubuntu SMP PREEMPT_DYNAMIC Tue Apr 15 19:04:15 UTC 2025 x86_64
```

6.2.3.2 ROCm installation

Perform the ROCm installation steps, found here:

<https://rocm.docs.amd.com/projects/install-on-linux/en/docs-6.4.3/install/quick-start.html>



Note: The system must be rebooted to apply all the settings.

```
wget https://repo.radeon.com/amdgpu-install/6.4.3/ubuntu/noble/amdgpu-install_6.4.60403-1_all.deb
sudo apt install ./amdgpu-install_6.4.60403-1_all.deb
sudo apt update
sudo apt install python3-setuptools python3-wheel
sudo usermod -a -G render,video $LOGNAME # Add the current user to the render and video groups
sudo apt install rocm
```

6.2.3.3 Post-installation

Perform the post-installation checks. The detailed installation checks can be found here:

<https://rocm.docs.amd.com/projects/install-on-linux/en/docs-6.4.3/install/post-install.html#>

1. Configure the system linker by indicating where to find the shared objects (.so files) for the ROCm applications.

```
cse@slate4:~$ sudo tee --append /etc/ld.so.conf.d/rocm.conf <<EOF
/opt/rocm/lib
/opt/rocm/lib64
EOF
sudo ldconfig
```

2. List all the ROCm commands that are supported.

```
cse@slate4:~$ sudo update-alternatives --display rocm
rocm - auto mode
  link best version is /opt/rocm-6.4.3
  link currently points to /opt/rocm-6.4.3
  link rocm is /opt/rocm
/opt/rocm-6.4.3 - priority 649637140
```

3. Add ROCm binaries to the `$PATH` env variable.

```
export PATH=$PATH:/opt/rocm-6.4.3/bin/ #Add also to /etc/environment
```

4. Verify kernel-mode driver installation.

```
cse@slate4:~$ dkms status
amdgpu/6.12.12-2194681.24.04, 6.8.0-85-generic, x86_64: installed
```

Post installation instructions:

1. Verify that ROCm is functioning as expected.

```
cse@slate4:~$ rocminfo

ROCK module version 6.12.12 is loaded
=====
HSA System Attributes
=====
Runtime Version:          1.15
Runtime Ext Version:      1.7
System Timestamp Freq.:   1000.000000MHz
Sig. Max Wait Duration:   18446744073709551615 (0xFFFFFFFFFFFFFFFF)
Machine Model:            LARGE
System Endianness:        LITTLE
Mwaitx:                   DISABLED
DMAbuf Support:           YES

=====
HSA Agents
=====
*****
Agent 1
*****
  Name:                    AMD EPYC 9654 96-Core Processor
  Uuid:                    CPU-XX
  Marketing Name:         AMD EPYC 9654 96-Core Processor
--MORE--

cse@slate4:~$ clinfo
Number of platforms:      1
Platform Profile:         FULL_PROFILE
Platform Version:         OpenCL 2.1 AMD-APP (3635.0)
Platform Name:            AMD Accelerated Parallel Processing
Platform Vendor:          Advanced Micro Devices, Inc.
Platform Extensions:      cl_khr_icd cl_amd_event_callback
-
Platform Name:            AMD Accelerated Parallel Processing
Number of devices:        8
Device Type:              CL_DEVICE_TYPE_GPU
Vendor ID:                1002h
```

```
Board name: AMD Instinct MI300X
Device Topology: PCI[ B#5, D#0, F#0 ]
Max compute units: 304
Max work items dimensions: 3
  Max work items[0]: 1024
  Max work items[1]: 1024
  Max work items[2]: 1024
Max work group size: 256
```

2. Verify that all eight GPUs are visible.

```
cse@slate4:~$ rocm-smi --showhw
===== ROCm System Management Interface
=====
===== Concise Hardware Info
=====
GPU  NODE  DID      GUID    GFX VER  GFX RAS  SDMA RAS  UMC RAS  VBIOS          BUS
PARTITION ID
0    2     0x74a1  28851  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:05:00.0 0
1    3     0x74a1  51499  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:27:00.0 0
2    4     0x74a1  57603  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:47:00.0 0
3    5     0x74a1  22683  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:65:00.0 0
4    6     0x74a1  53458  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:85:00.0 0
5    7     0x74a1  26954  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:A7:00.0 0
6    8     0x74a1  16738  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:C7:00.0 0
7    9     0x74a1  63738  gfx942   ENABLED  ENABLED   ENABLED  113-M3000100-103
0000:E5:00.0 0
=====
=====End of ROCm SMI Log=====
```

```
cse@slate2:~$ cat /opt/rocm/.info/version
6.4.3-128
```



Note: If the installation has not completed properly, all GPUs will be in a DISABLED state.

3. After the ROCm installation is completed, install the GPU drivers.

```
wget https://repo.radeon.com/amdgpu-install/6.4.3/ubuntu/noble/amdgpu-install_6.4.60403-1_all.deb
sudo apt install ./amdgpu-install_6.4.60403-1_all.deb
sudo apt update
sudo apt install "linux-headers-$(uname -r)" "linux-modules-extra-$(uname -r)"
sudo apt install amdgpu-dkms
```

6.2.4 AOC-S400G-B1C 400G NIC driver installation

This validated design uses the Supermicro's AOC-S400G-B1C (Broadcom BCM57608 Thor2) 400 Gb NICs as the GPU-attached NICs. Therefore, the following instructions apply only to Broadcom Thor2 NICs. Other GPU NIC models are supported but require different configuration steps. For information about using AMD Pensando Pollara network interface cards in this solution, see *AMD Pensando Pollara Validation* on the Nokia Documentation Center.

Before proceeding, review the Broadcom guide for installation procedure:

<https://docs.broadcom.com/doc/957608-AN2XX>.

Perform the following steps to install the Broadcom Thor2 NICs.

1. Boot the server.
2. Download the Broadcom software package.
3. Follow the steps in Section 2.2.1 of the BCM957608-AN2XX document to install the NIC drivers.
4. Run the software installer to install the Peer Memory Direct versions of the bnxt modules.

```
$ sudo ./install.sh -v -i 23:00.0 -i 06:00.0 -i 43:00.0 -i 66:00.0 -i a3:00.0 -i 86:00.0 -i c3:00.0 -i e6:00.0 -f -g
```

6.2.4.1 Verify driver installation

1. Verify if RoCE QoS configurations are set.

```
cse@slate4:~/benchmarking$ cat /etc/bnxt_re/bnxt_re.conf
ENABLE_FC=1
FC_MODE=3
ROCE_PRI=3
ROCE_DSCP=26
CNP_PRI=7
CNP_DSCP=48
ROCE_BW=50
UTILITY=3
```

2. List all the available Broadcom NICs and the index for each of them.

```
cse@slate4:~/benchmarking$ sudo niccli --list
-----
NIC CLI v233.0.150.0 - Broadcom Inc. (c) 2025 (Bld-106.52.39.138.16.0)
-----

  BoardId(Rev)   MAC Address      FwVersion   PCIAddr        Type   Mode
1) BCM57608(B1) 90:5A:08:26:3A:C0 231.2.63.0  0000:06:00.0  NIC   PCI
2) BCM57608(B1) 90:5A:08:26:42:10 231.2.63.0  0000:23:00.0  NIC   PCI
```

| | | | | | | |
|-----|--------------|-------------------|------------|--------------|-----|-----|
| 3) | BCM57608(B1) | 7C:C2:55:BA:FC:E8 | 230.2.49.0 | 0000:30:00.0 | NIC | PCI |
| 4) | BCM57608(B1) | 7C:C2:55:BA:FC:E9 | 230.2.49.0 | 0000:30:00.1 | NIC | PCI |
| 5) | BCM57608(B1) | 90:5A:08:26:3E:E0 | 231.2.63.0 | 0000:43:00.0 | NIC | PCI |
| 6) | BCM57608(B1) | 90:5A:08:26:3E:F0 | 231.2.63.0 | 0000:66:00.0 | NIC | PCI |
| 7) | BCM57608(B1) | 90:5A:08:26:42:F0 | 231.2.63.0 | 0000:86:00.0 | NIC | PCI |
| 8) | BCM57608(B1) | 90:5A:08:26:3F:30 | 231.2.63.0 | 0000:A3:00.0 | NIC | PCI |
| 9) | BCM57608(B1) | 90:5A:08:26:42:00 | 231.2.63.0 | 0000:C3:00.0 | NIC | PCI |
| 10) | BCM57608(B1) | 90:5A:08:26:42:50 | 231.2.63.0 | 0000:E6:00.0 | NIC | PCI |

3. Verify the firmware version installed on the NIC.

```
cse@slate4:~/benchmarking$ sudo niccli -i 1 pkgver
-----
NIC CLI v233.0.150.0 - Broadcom Inc. (c) 2025 (Bld-106.52.39.138.16.0)
-----

Package Information :
  Active Package Version      : 231.2.63.0
  Package Version on NVM     : 231.2.63.0
  Primary SBI Version        : 231.0.53.0
  Secondary SBI Version      : 231.0.53.0
  Primary SRT Version        : 231.2.63.0
  Secondary SRT Version      : 231.2.63.0
  Primary CRT Version        : 231.2.63.0
  Secondary CRT Version      : 231.2.63.0
```

4. Verify the NIC QoS settings.

```
cse@slate4:~/benchmarking$ sudo niccli -i 1 getqos
-----
NIC CLI v233.0.150.0 - Broadcom Inc. (c) 2025 (Bld-106.52.39.138.16.0)
-----

IEEE 8021QAZ ETS Configuration TLV:
  PRI0_MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:0 7:2
  TC Bandwidth: 50% 50% 0%
  TSA_MAP: 0:ets 1:ets 2:strict
IEEE 8021QAZ PFC TLV:
  PFC enabled: 3
IEEE 8021QAZ APP TLV:
  APP#0:
    Priority: 7
    Sel: 5
    DSCP: 48

  APP#1:
    Priority: 3
    Sel: 5
    DSCP: 26

  APP#2:
    Priority: 3
```

```
Sel: 3
UDP or DCCP: 4791
```

```
TC Rate Limit: 100% 100% 100% 0% 0% 0% 0% 0%
```

6.2.4.2 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, `bnxt_re` 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 `ens42np0`, though the specific interface name may vary depending on the system.

```
cse@slate3:~$ ethtool -i ens42np0 | grep -i bus
bus-info: 0000:23:00.0
```

1. Map the interfaces to the NUMA nodes using either interface names or PCI addresses.

- Based on interface name

```
cse@slate3:~$ cat /sys/class/net/ens42np0/device/numa_node
0
```

- Based on interface PCI address

```
cse@slate3:~$ cat /sys/bus/pci/devices/0000\:23\:00.0/numa_node
0
```

2. Map the interfaces to the RDMA interfaces.

```
cse@slate3:~$ rdma link show | grep ens42np0
link bnxt_re1/1 state ACTIVE physical_state LINK_UP netdev ens42np0
```

3. List all GPU PCI addresses. The same information can be retrieved with `rocm-smi -showhw` command.

```
cse@slate3:~$ sudo lspci -d 1002:74a1
05:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
27:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
47:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
65:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
85:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
a7:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
```

```
c7:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
e5:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
```

- 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.

```
cse@slate3:~$ lstopo
<snipped>
HostBridge
  PCIBridge
    PCIBridge
      PCI 23:00.0 (Ethernet)
        Net "ens42np0"
        OpenFabrics "bnxt_re1"
      PCIBridge
        PCIBridge
          PCI 27:00.0 (ProcessingAccelerator)
<snipped>
```

- Map the GPU PIC address to the OAM-ID.

```
cse@slate3:~$ amd-smi static | grep -e GPU -e BDF -e OAM_ID
GPU: 1
    OAM_ID: 6
    BDF: 0000:27:00.0
```

- 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 | GPU OAM-ID |
|----------|----------|-------------|---------------|---------------|----------------|-----------------|------------|
| 0 | 1 | 23:00.0 | ens42np0 | bnxt_re1 | GPU1 | 27:00.0 | 6 |
| 0 | 2 | 06:00.0 | ens41np0 | bnxt_re0 | GPU0 | 05:00.0 | 7 |
| 0 | 3 | 43:00.0 | ens32np0 | bnxt_re4 | GPU2 | 47:00.0 | 4 |
| 0 | 4 | 66:00.0 | ens31np0 | bnxt_re5 | GPU3 | 65:00.0 | 5 |
| 1 | 5 | a3:00.0 | ens22np0 | bnxt_re7 | GPU5 | a7:00.0 | 2 |
| 1 | 6 | 86:00.0 | ens21np0 | bnxt_re6 | GPU4 | 85:00.0 | 3 |
| 1 | 7 | c3:00.0 | ens12np0 | bnxt_re8 | GPU6 | c7:00.0 | 0 |
| 1 | 8 | e6:00.0 | ens11np0 | bnxt_re9 | GPU7 | e5:00.0 | 1 |

6.3 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>

  ens51f0np0: { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens51f1np1: { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens42np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens41np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens32np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens31np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens22np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens21np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens12np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }
  ens11np0:   { dhcp4: no, dhcp6: no, optional: true, mtu: 9000 }

  bonds:
    bond0:
      interfaces:
        - ens51f0np0
        - ens51f1np1
      mtu: 9000
      parameters:
        mode: 802.3ad
        lacp-rate: fast
        mii-monitor-interval: 100

  vlans:
    ens42np0.1000: # GPU 1
      id: 1000
      link: ens42np0
      mtu: 9000
      accept-ra: false
      dhcp4: no
      dhcp6: no
```

```
optional: true
addresses:
  - "fd00:1:1:1:0:1:0:2/96"
routes:
  - to: "::/0"
    via: "fd00:1:1:1:0:1:0:1"
    on-link: true
    metric: 10
    table: 101
routing-policy:
  - from: "fd00:1:1:1:0:1:0:2/96"
    table: 101
ens41np0.1000: # GPU 2
id: 1000
link: ens41np0
mtu: 9000
accept-ra: false
dhcp4: no
dhcp6: no
optional: true
addresses:
  - "fd00:1:2:1:0:1:0:2/96"
routes:
  - to: "::/0"
    via: "fd00:1:2:1:0:1:0:1"
    on-link: true
    metric: 10
    table: 102
routing-policy:
  - from: "fd00:1:2:1:0:1:0:2/96"
    table: 102
```

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

```
bond0.100: # storage network
id: 100
link: bond0
mtu: 9000
addresses:
  - 192.168.1.174/24
bond0.200: # frontend network
id: 200
link: bond0
mtu: 9000
addresses:
  -192.168.2.174/24
```

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:
  ACPIRATL3CacheAsNUMADomain_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
  '*':
  ACPIRATL3CacheAsNUMADomain: Disabled
  IOMMU: Disabled
  NUMANodesPerSocket: NPS1
  SMTControl: Disabled
  DFCstates: Disabled
  GlobalC_stateControl: Disabled
  DeterminismControl: Manual
  PowerProfileSelection: "Maximum IO Performance Mode"
<snipped>

```

- 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.

```

Weka Configurator (Hosts)
Select DP Networks:
[ ] 100.116.160.0/21 - ETH, 1 Gbps, 8 hosts
[X] 192.168.1.0/24 - ETH, 200 Gbps, 8 hosts

Number of Hosts: 8
Enable Options:
[X] High Availability (HA)
[X] Multicontainer Backends (MCB)

Select Hosts:
[X] weka01.ncse.io
[X] weka02
[X] weka03.ncse.io
[X] weka04.ncse.io
[X] weka05.ncse.io
[X] weka06.ncse.io
[X] weka07.ncse.io
[X] weka08.ncse.io

```

```

Weka Configurator (Cores)
Host Configuration Reference
Cores per host: 48
Drives per host: 4
Number of hosts: 8
Data Drives: 5
Parity Drives: 2
Hot Spares: 1
Bias:
[X] Enable Protocols
[ ] Protocols are Primary
[X] DRIVES over COMPUTE
Reserved RAM per Host: 20
Cores for OS: 2
Cores for Protocols: 4
Usable Weka Cores: 42
Used Weka Cores: 18
FE Cores: 2
DRIVES Cores: 4
COMPUTE Cores: 12
Cluster Name: weka

```

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
```

- 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
```

- Start the WEKA storage cluster.

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

- 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

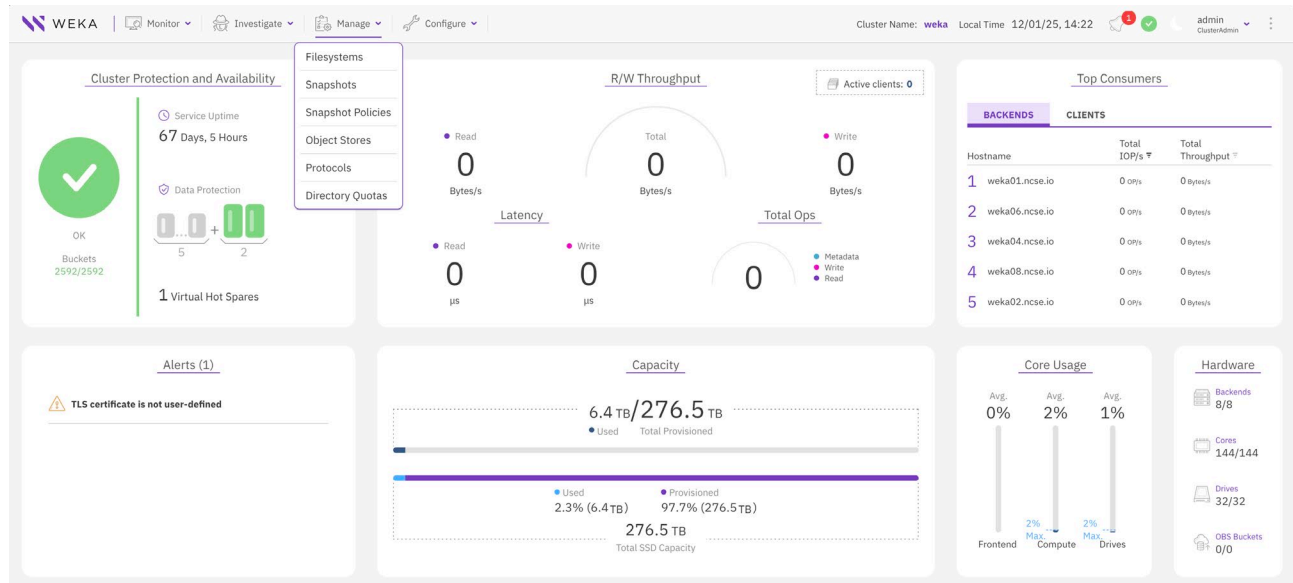
io status: STARTED 7 seconds ago (144 io-nodes UP, 2592 Buckets UP)
link layer: Ethernet
clients: 0 connected
reads: 0 B/s (0 IO/s)
writes: 0 B/s (0 IO/s)
```

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

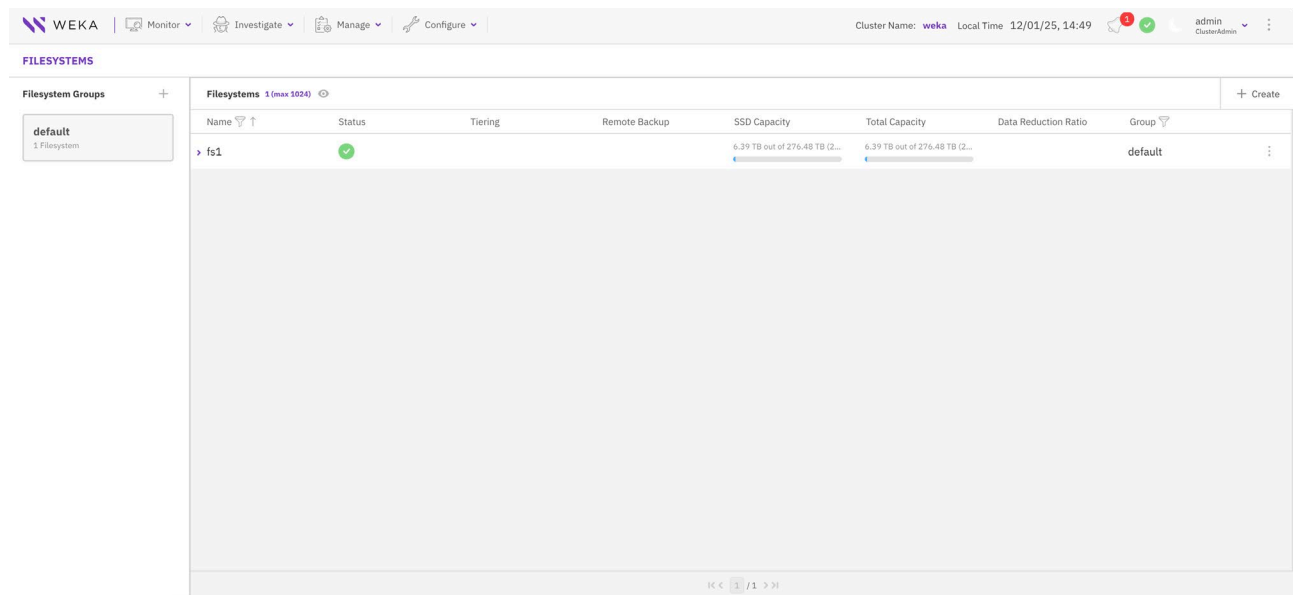
7.5 Creating the storage file system

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**.

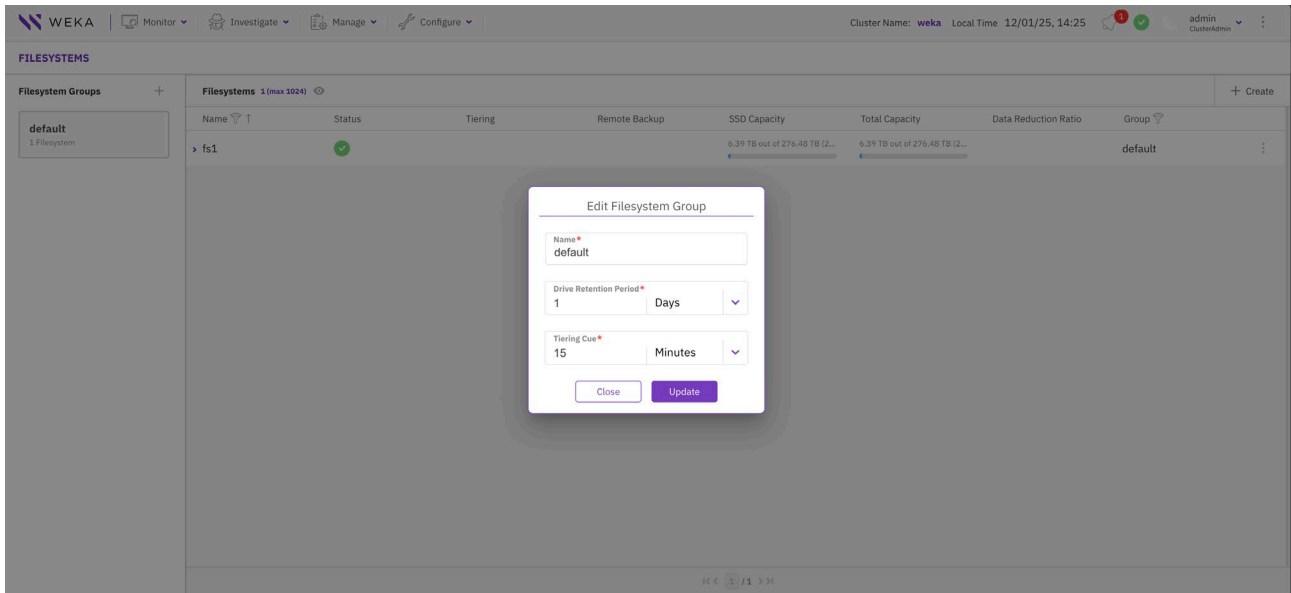


2. Create a **Filesystem Group**. Every file system must belong to a group. A group is a collection of file systems that share the same tiering rules.

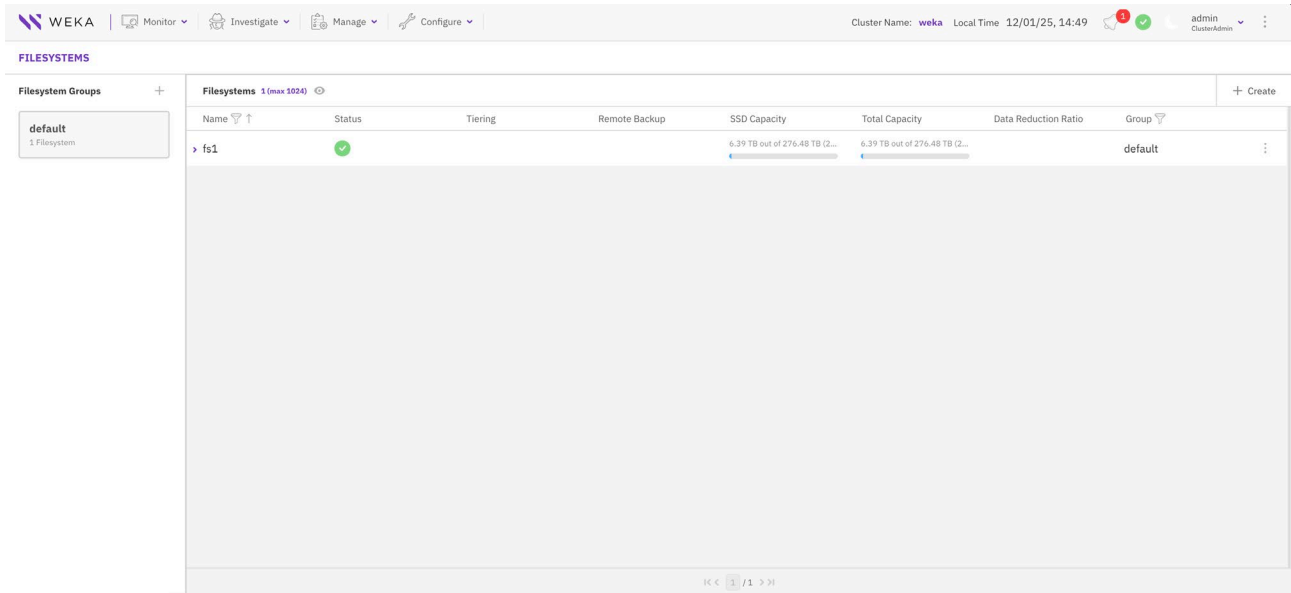


3. Name the Filesystem Group. You can leave the **Drive Retention Period** and **Tiering Cue** at their default values.

- 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

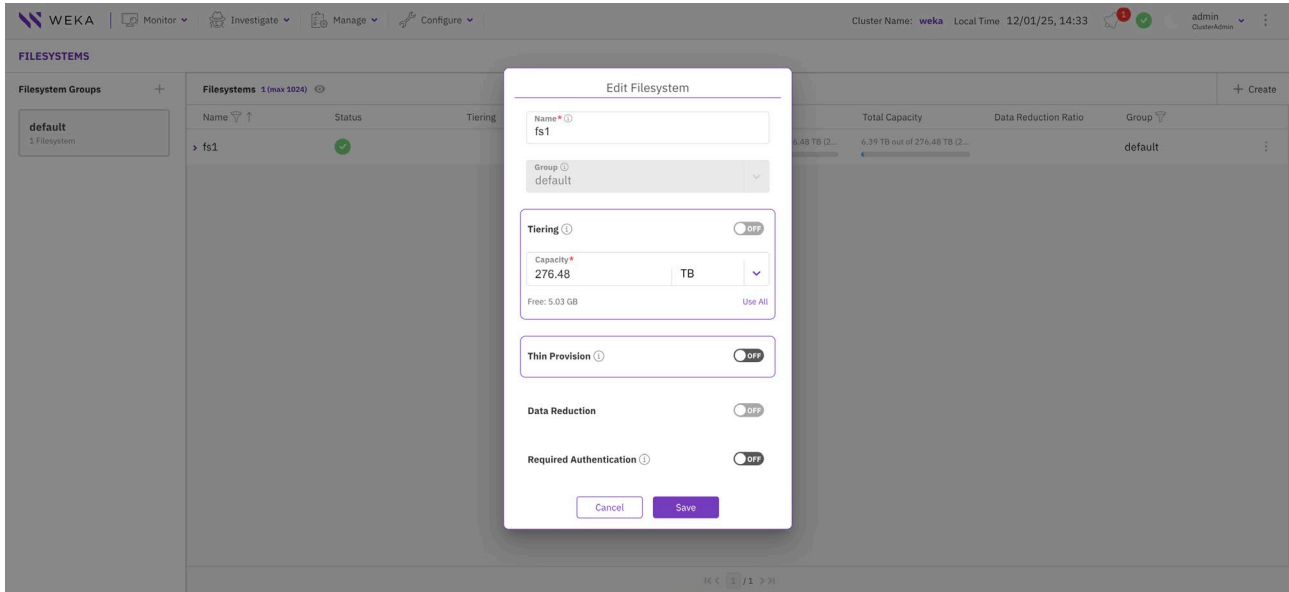


4. Click the Create button at the top right to create a file system.



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

- 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.

```
cse@slate4:~$ weka user login
Username: admin
Password:
+-----+
| Login completed successfully |
+-----+
Default profile updated successfully
cse@slate4:~$
cse@slate4:~$ 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 {
```



```
        maximum-paths 1
    }
}
ipv4-unicast {
    advertise-ipv6-next-hops true
    receive-ipv6-next-hops true
}
}
afi-safi ipv6-unicast {
    admin-state disable
    multipath {
        allow-multiple-as true
        ebgp {
            maximum-paths 2
        }
    }
}
}
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 {
  }
}
```

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 AMD MI300X 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-amd-gpus-backend-fabric
    }
  }
  action {
    policy-result accept
  }
}
```

```
A:admin@backend-stripe1-leaf1# info routing-policy prefix-set prefixset-routeleak-import-
amd-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 amd-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-1/1.1000 {
  }
  interface ethernet-1/2.1000 {
  }
  inter-instance-policies {
    apply-policy {
      import-policy import-routeleak-amd-gpus-backend-fabric
      export-policy export-routeleak-amd-gpus-backend-fabric
    }
  }
}

A:admin@backend-stripe1-leaf1# info routing-policy policy export-routeleak-amd-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-amd-gpus-backend-fabric
      }
    }
  }
}
```

```
    action {
      policy-result accept
    }
  }
```

```
A:admin@backend-stripe1-leaf1# info routing-policy policy import-routeleak-amd-gpus-backend-fabric
```

```
  default-action {
    policy-result reject
  }
  statement 10 {
    match {
      protocol bgp
      prefix {
        prefix-set prefixset-routeleak-import-amd-gpus-backend-fabric
      }
    }
    action {
      policy-result accept
    }
  }
}
```

```
A:admin@backend-stripe1-leaf1# info routing-policy prefix-set prefixset-routeleak-export-amd-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-
amd-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
    }
  }
}
```



```
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-front-end-server
  admin-state enable
  ethernet {
    aggregate-id lag13
    lacp-port-priority 32768
  }

A:admin@frontend-leaf1# info interface lag13
  description lag-front-end-server
  admin-state enable
  vlan-tagging true
  subinterface 200 {
    type bridged
    description front-end-server
    admin-state enable
    vlan {
      encaps {
        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
        }
    }
}

```



```
}
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
```

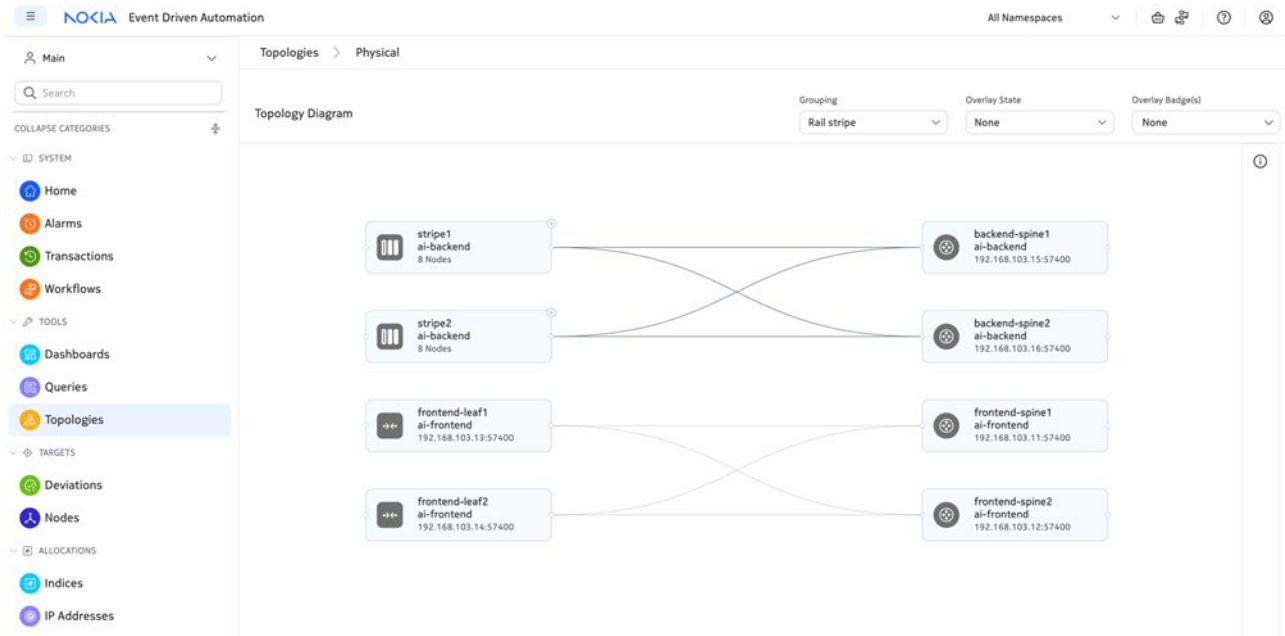
| NAMESPACE | NAME | READY | STATUS | RESTARTS | AGE |
|--------------|--|-------|---------|----------|-------|
| cert-manager | cert-manager-777c6f8ff4-njdtm | 1/1 | Running | 0 | 6d4h |
| cert-manager | cert-manager-cainjector-6558fc6578-rbbft | 1/1 | Running | 0 | 6d4h |
| cert-manager | cert-manager-webhook-6964489477-mn4x7 | 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-q19ng | 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-559f8894d6-197hw | 1/1 | Running | 0 | 6d4h |
| eda-system | eda-toolbox-76886bc564-ftkvq | 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-6fbf94cbbb-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-m717g | 1/1 | Running | 0 | 6d4h |
| kube-system | etcd-eda-demo-control-plane | 1/1 | Running | 0 | 6d4h |
| kube-system | kindnet-tnhnm | 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-17q4b | 1/1 | Running | 0 | 6d4h |

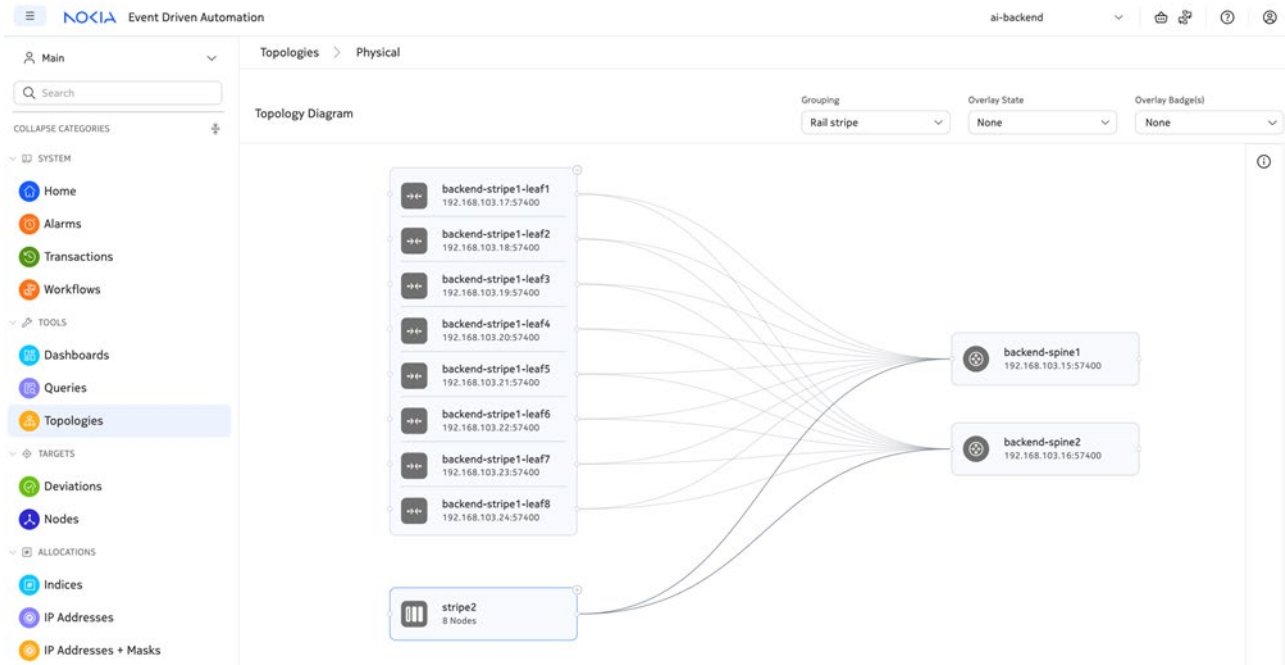
Some commonly used pods and their functionalities are:

- **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 forth. 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 leaves 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 the converged frontend and storage EDA integration section).

```

// 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 leaves 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 leaves 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: amd-gpus
  rocev2QoS:
    ecnMaxDropProbabilityPercent: 100
    ecnSlopeMaxThresholdPercent: 80
    ecnSlopeMinThresholdPercent: 5
    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 AMD MI300X 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 three configlets –

- configlet to enable DLB
- configlet to modify QoS
- configlet to enable IPv6 multipath

```

// IPv6 multipath configlet

apiVersion: config.eda.nokia.com/v1alpha1
kind: Configlet
metadata:
  name: ipv6-multipath
  namespace: ai-backend
spec:
  endpointSelector:
    - eda.nokia.com/stripe=stripe1
    - eda.nokia.com/stripe=stripe2
  operatingSystem: srl
  priority: 100
  configs:
    - path: .network-instance{.name=="default"}.protocols.bgp.afs-safi{.afi-safi-
name=="ipv6-unicast"}
      operation: Update
      config: |-
        {
          "multipath": {
            "allow-multiple-as": true,
            "ebgp": {
              "maximum-paths": 2
            }
          }
        }

// 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"
                }
              ]
            }
          }
        }
}

// 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
            }
          ]
        }
    ---
  apiVersion: config.eda.nokia.com/v1alpha1
  kind: Configlet
  metadata:
    name: qos-pfc-headroom
    namespace: ai-backend
  spec:
    endpointSelector:
      - eda.nokia.com/role=leaf
      - eda.nokia.com/role=spine
    operatingSystem: srl
    priority: 100
    configs:
      - path: .qos.linecard{.slot==1}.forwarding-complex{.name=="0"}.input
        operation: Update
        config: |-
          {
            "pfc-buffer-reservation": 10
          }
      ---
  apiVersion: config.eda.nokia.com/v1alpha1
  kind: Configlet
  metadata:
    name: qos-enable-ecn-slope
    namespace: ai-backend
  spec:
    endpointSelector:
      - eda.nokia.com/role=leaf
      - eda.nokia.com/role=spine
    operatingSystem: srl
    priority: 100
    configs:
      - path: .qos.buffer-management.queue-management-profile{.name=="egress-backend-backend-fabric-2"}.wred
        operation: Update

```

```
config: |-
  {
    "wred-slope": [
      {
        "traffic-type": "all",
        "drop-probability": "all",
        "enable-ecn": true,
        "min-threshold-percent": 30,
        "max-threshold-percent": 85,
        "slope-enabled": true,
        "max-drop-probability-percent": 100
      }
    ]
  }
---
apiVersion: config.eda.nokia.com/v1alpha1
kind: Configlet
metadata:
  name: qos-unicast-3-queue
  namespace: ai-backend
spec:
  endpointSelector:
    - eda.nokia.com/role=leaf
    - eda.nokia.com/role=spine
  operatingSystem: srl
  priority: 100
  configs:
    - path: .qos.scheduler-policies.scheduler-policy{.name=="egress-backend-backend-
fabric"}.scheduler{.sequence==1}
      operation: Update
      config: |-
        {
          "input": [
            {
              "id": "unicast-0",
              "queue-name": "unicast-0",
              "peak-rate-percent": 10,
              "weight": 10
            }
          ]
        }
---
apiVersion: config.eda.nokia.com/v1alpha1
kind: Configlet
metadata:
  name: qos-unicast-6-queue
  namespace: ai-backend
spec:
  endpointSelector:
    - eda.nokia.com/role=leaf
    - eda.nokia.com/role=spine
  operatingSystem: srl
  priority: 100
  configs:
```

```

- path: .qos.scheduler-policies.scheduler-policy{.name=="egress-backend-backend-
fabric"}.scheduler{.sequence==0}
  operation: Update
  config: |-
    {
      "input": [
        {
          "id": "unicast-6",
          "queue-name": "unicast-6",
          "peak-rate-percent": 80
        }
      ]
    }
  ---

```

10 Test summary

| # | Feature | SRL v25.10.1 | EDA 25.8.3 | |
|----|--|--------------|------------|-----------|
| | | 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 | Yes |
| 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 | |
| RCCL 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 | RCCL performance per algorithm | Yes |
| MLCommons benchmarking – training | | |
| 40 | Llama2 70B 8/16/24/32 accelerators | Yes |
| MLCommons benchmarking – inference | | |
| 41 | Llama 2 70B – 8/16/24/32 accelerators – 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
}
```


| | | | | | | | |
|----------------|---------|-------------------------------------|-------------------------|----|-----|-------------|--|
| 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 leaves, 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: 420000016, local AS: 420000014
Type      : dynamic
Description : None
Group     : bgpgroup-ebgp-stripe-connector
-----

```

```
Origin codes: i=IGP, e=EGP, ?=incomplete
```

```

-----
+-----+
| LocPref      Network      AsPath      Path-id      Origin      | Next Hop      MED      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| fd00:1:1:1::/94      ?      | 0      |      |      | fe80::4e62:cdff:feb3:6      -      |
| [420000014]      |      |      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```
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: 420000016, local AS: 420000014
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      AsPath      Path-id      Origin      | Next Hop      MED      |
+-----+-----+-----+-----+-----+-----+-----+

```

```

| 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 the advertised via BGP to its peers.

```

A:admin@backend-stripe1-leaf1# show network-instance amd-gpus route-table ipv6-unicast summary
-----
IPv6 unicast route table of network instance amd-gpus
-----

```

| Prefix | Next-hop Interface | Backup Next-hop | ID (Type) | Route Type | Route Owner | Active | Origin | Metric | Pref | Next-hop (Type) |
|------------------------|------------------------|-----------------|-----------|---------------------------|--------------|--------|----------|--------|------|-----------------------------|
| | | | | Backup Next-hop Interface | Next-hop | | Network | | | |
| | | | | | | | Instance | | | |
| fd00:1:1:1:0:1::/96 | ethernet-1/1.1000 | | 7 | local | net_inst_mgr | True | amd-gpus | 0 | 0 | fd00:1:1:1:0:1:0:1 (direct) |
| fd00:1:1:1:0:1:0:1/128 | None | | 7 | host | net_inst_mgr | True | amd-gpus | 0 | 0 | None |
| fd00:1:1:1:0:1:0:2/128 | ethernet-1/1.1000 | | 7 | arp-nd | arp_nd_mgr | True | amd-gpus | 0 | 1 | fd00:1:1:1:0:1:0:2 (direct) |
| fd00:1:1:1:0:2::/96 | ethernet-1/2.1000 | | 8 | local | net_inst_mgr | True | amd-gpus | 0 | 0 | fd00:1:1:1:0:2:0:1 (direct) |
| fd00:1:1:1:0:2:0:1/128 | None | | 8 | host | net_inst_mgr | True | amd-gpus | 0 | 0 | None |
| fd00:1:1:1:0:2:0:2/128 | ethernet-1/2.1000 | | 8 | arp-nd | arp_nd_mgr | True | amd-gpus | 0 | 1 | fd00:1:1:1:0:2:0:2 (direct) |
| fd00:1:2:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:1:3:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:1:4:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:1:5:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:1:6:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:1:7:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:1:8:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | 6b1 (direct) |
| fd00:2:9:1:1::/94 | fe80::3200:fcff:fe2e:1 | ethernet-1/31.0 | 0 | bgp | bgp_mgr | True | default | 0 | 170 | c82 (direct) |
| ethernet-1/32.0 | | | | | | | | | | |


```

| 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 | Last Update | Destination | Dest Index | Type | Active |
+-----+-----+-----+-----+-----+
| 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 | Last Update | Destination | Dest Index | Type | Active |
+-----+-----+-----+-----+-----+
| 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 AMD MI300X 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   : 00:00:00:00:11
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 | 00:00:00:00:11 | 3 |
| 82:0C:97:65:F8:6E | 31 | | 2 | | 2 | ue | |
+-----+-----+-----+-----+-----+-----+-----+
-----
LACP State for lag5
-----
Lag Id       : lag5
Interval    : FAST
Mode        : ACTIVE
System Id   : 00:00:00:00:15
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 | 00:00:00:00:15 | 6 |
| E0:9D:73:BD:89:F8 | 31 | | 6 | | 2 | ue | |
+-----+-----+-----+-----+-----+-----+-----+
-----
LACP State for lag13
-----
Lag Id       : lag13
Interval    : FAST
Mode        : ACTIVE
System Id   : 00:00:00:00:23
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/ | 00:00:00:00:23 | 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 indicate if any pause frames are being generated or received.

```
A:admin@backend-stripe1-leaf1# info from state qos interfaces interface ethernet-1/1
output queues queue unicast-3 queue-statistics
  aggregate-statistics {
    last-clear "2025-11-26T13:22:44.287Z (4 days ago)"
    transmitted-packets 1959178975
    transmitted-octets 6166352627430
    dropped-packets 0
    dropped-octets 0
  }

A:admin@backend-stripe1-leaf1# info from state qos interfaces interface ethernet-1/1 pfc
statistics pfc-priority 3
  pfc-pause-frames-received 0
  pfc-pause-frames-generated 764
  pfc-transitions 0
  deadlock-recovery-occurrencesInterface statistics for discards and drops
```

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-1/1 statistics
  in-packets 1963372892
  in-octets 6166741248552
  in-unicast-packets 1963372776
  in-broadcast-packets 0
  in-multicast-packets 116
  in-discarded-packets 116
  in-error-packets 0
  in-fcs-error-packets 0
  out-packets 1959977745
  out-octets 6166435578046
  out-unicast-packets 1959961829
  out-broadcast-packets 0
  out-multicast-packets 15916
  out-discarded-packets 15369
  out-error-packets 0
  carrier-transitions 0
  last-clear "2025-11-26T13:22:44.263Z (4 days ago)"
```

11.2 EDA validation

11.2.1 EDA pods validation

Use `kubect1 get pods -A` to ensure all pods are in a running state.

```
cse@d2vm-4~ kubect1 get pods -A
NAMESPACE          NAME                                READY   STATUS    RESTARTS   AGE
cert-manager       cert-manager-777c6f8ff4-njdtm     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-tnhnm | 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-17q4b | 1/1 | Running | 0 | 8d |

11.2.2 TopoNode validation

Use `kubectl get toponodes -A` to ensure that all the nodes 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-H5-64D | 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-H5-640 | 25.10.1 | srl | true | normal | Connected | Synced | 7d8h |
| ai-frontend | frontend-spine2 | 7220 | IXR-H5-640 | 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.

```
cse@d2vm-4 ~ 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:  2025-11-26T00:46:48Z
  Generation:         1
  Resource Version:    3105117
  UID:                32d9c936-fe71-4644-8179-91325c35f44f
Spec:
  Asn Pool:  asn-pool-4byte
  Gpu Isolation Groups:
    Interface Selector:
      eda.nokia.com/role=edge
    Name:  amd-gpus
  rocev2QoS:
    Ecn Max Drop Probability Percent:  100
    Ecn Slope Max Threshold Percent:   80
    Ecn Slope Min Threshold Percent:   5
    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/stripes=stripe1
    Stripe ID:  1
    Gpu Vlan:  1000
    Name:      stripe2
    Node Selector:
      eda.nokia.com/stripes=stripe2
    Stripe ID:  2
  systemPoolIPV4:  system0
Status:
  Last Change:     2025-12-02T17:41:49.000Z
```

```
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 status. 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 interface enable, link status, speed.

```

kubect1 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 | degraded | 2025-12-04T19:08:32.000Z |
| ai-frontend | storage | degraded | 2025-12-04T19:08:32.000Z |

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 storage node with its detailed usage. To view this information, navigate to: WEKA Home UI > Configure > Cluster Servers.

| UID | Hostname | IP Address | Version | Status | Cores | Drives | Load | Memory | Architecture | Uptime |
|-----------|----------------|---------------|----------|--------|-------|--------|------|-----------|--------------|--------|
| 46880... | weka08.ncse.io | 192.168.1.196 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| a4b2e8... | weka05.ncse.io | 192.168.1.193 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| 64f2e0... | weka06.ncse.io | 192.168.1.194 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| d3d63... | weka07.ncse.io | 192.168.1.195 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| 53e9c2... | weka04.ncse.io | 192.168.1.192 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| 69400... | weka01.ncse.io | 192.168.1.189 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| ebf38c... | weka02.ncse.io | 192.168.1.190 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |
| 21b97... | weka03.ncse.io | 192.168.1.191 | 5.0.3.19 | UP | 18/18 | 4/4 | 1% | 267.03 GB | x86_64 | 6d |

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
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 ROCm version.

```
cse@slate1:~$ cat /opt/rocm/.info/version
6.4.3-128
```

2. Confirm all MI300X GPUs accelerators are present and available on the PCIe bus.

```
sudo lspci -d 1002:74a1
05:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
27:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
47:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
65:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
85:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
a7:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
c7:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
e5:00.0 Processing accelerators: Advanced Micro Devices, Inc. [AMD/ATI] Aqua Vanjaram
[Instinct MI300X]
```

3. Verify that the GPUs are detected by ROCm.

```
cse@slate1:~$ rocm-smi --showhw
===== ROCm System Management Interface =====
===== Concise Hardware Info =====
GPU  NODE  DID      GUID      GFX_VER  GFX_RAS  SDMA_RAS  UMC_RAS  VBIOS      BUS      PARTITION ID
0     2     0x74a1  28851    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:05:00.0  0
1     3     0x74a1  51499    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:27:00.0  0
2     4     0x74a1  57603    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:47:00.0  0
3     5     0x74a1  22683    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:65:00.0  0
4     6     0x74a1  53458    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:85:00.0  0
5     7     0x74a1  26954    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:A7:00.0  0
6     8     0x74a1  16738    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:C7:00.0  0
7     9     0x74a1  63738    gfx942   ENABLED  ENABLED   ENABLED   113-M3000100-103  0000:E5:00.0  0
=====
===== End of ROCm SMI Log =====
```

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

```
cse@slate1:~$ amd-smi monitor -putm
GPU POWER GPU_T MEM_T GFX_CLK GFX% MEM% MEM_CLOCK
0 138 W 36 °C 31 °C 131 MHz 0 % 0 % 900 MHz
1 138 W 34 °C 30 °C 132 MHz 0 % 0 % 900 MHz
2 138 W 39 °C 33 °C 131 MHz 0 % 0 % 900 MHz
3 143 W 39 °C 31 °C 131 MHz 0 % 0 % 900 MHz
4 138 W 40 °C 35 °C 132 MHz 0 % 0 % 900 MHz
5 139 W 37 °C 32 °C 131 MHz 0 % 0 % 900 MHz
6 142 W 39 °C 33 °C 131 MHz 0 % 0 % 900 MHz
7 145 W 40 °C 35 °C 132 MHz 0 % 0 % 900 MHz
```

11.4.1 GPU NICs

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

1. Verify that the Broadcom NICs are listed.

```
cse@slate1:~$ sudo niccli --list
```

```
-----  
NIC CLI v233.0.150.0 - Broadcom Inc. (c) 2025 (Bld-106.52.39.138.16.0)  
-----
```

| | BoardId(Rev) | MAC Address | FwVersion | PCIAddr | Type | Mode |
|-----|--------------|-------------------|-------------|--------------|------|------|
| 1) | BCM57608(B1) | 90:5A:08:26:3F:50 | 231.2.63.0 | 0000:06:00.0 | NIC | PCI |
| 2) | BCM57608(B1) | 90:5A:08:26:42:70 | 231.2.63.0 | 0000:23:00.0 | NIC | PCI |
| 3) | BCM57608(B1) | 82:0C:97:65:F8:6E | 230.2.49.0 | 0000:30:00.0 | NIC | PCI |
| 4) | BCM57608(B1) | 82:0C:97:65:F8:6E | 230.2.49.0 | 0000:30:00.1 | NIC | PCI |
| 5) | BCM57608(B1) | 90:5A:08:26:3F:20 | 231.2.63.0 | 0000:43:00.0 | NIC | PCI |
| 6) | BCM57608(B1) | 7C:C2:55:BD:FC:F6 | 231.2.63.0 | 0000:66:00.0 | NIC | PCI |
| 7) | BCM57608(B1) | 90:5A:08:26:3E:D0 | 231.2.63.0 | 0000:86:00.0 | NIC | PCI |
| 8) | BCM57608(B1) | 90:5A:08:26:3E:00 | 231.2.63.0 | 0000:A3:00.0 | NIC | PCI |
| 9) | BCM57608(B1) | 90:5A:08:26:43:A0 | 231.2.63.0 | 0000:C3:00.0 | NIC | PCI |
| 10) | BCM57608(B1) | 90:5A:08:B4:99:F0 | 234.0.145.0 | 0000:E6:00.0 | NIC | PCI |

2. Verify the NIC QoS settings.

```
cse@slate1:~/benchmarking$ sudo niccli -i 1 getqos
```

```
-----  
NIC CLI v233.0.150.0 - Broadcom Inc. (c) 2025 (Bld-106.52.39.138.16.0)  
-----
```

```
IEEE 8021QAZ ETS Configuration TLV:
```

```
  PRIO_MAP: 0:0 1:0 2:0 3:1 4:0 5:0 6:0 7:2
```

```
  TC Bandwidth: 50% 50% 0%
```

```
  TSA_MAP: 0:ets 1:ets 2:strict
```

```
IEEE 8021QAZ PFC TLV:
```

```
  PFC enabled: 3
```

```
IEEE 8021QAZ APP TLV:
```

```
  APP#0:
```

```
    Priority: 7
```

```
    Sel: 5
```

```
    DSCP: 48
```

```
  APP#1:
```

```
    Priority: 3
```

```
    Sel: 5
```

```
    DSCP: 26
```

```
  APP#2:
```

```
    Priority: 3
```

```
    Sel: 3
```

```
    UDP or DCCP: 4791
```

 TC Rate Limit: 100% 100% 100% 0% 0% 0% 0% 0%

11.4.2 RoCEv2 statistics

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

```

cse@slate1:~$ sudo cat /sys/kernel/debug/bnxt_re/bnxt_re0/info
bnxt_re debug info:
=====[ IBDEV bnxt_re0 ]=====
link state: UP
Max QP:          131073
Max SRQ:         8192
Max CQ:          65536
Max MR:          262144
Max MW:          262144
Max AH:          131072
Max PD:          131072
Active QP:       1
Active RC QP:    0
Active UD QP:    0
Active SRQ:      0
Active CQ:       1
Active MR:       0
Active DMABUF MR: 0
Active MW:       0
Active AH:       0
Active HW AH:    0
Active PD:       1
QP Watermark:   305
RC QP Watermark: 296
UD QP Watermark: 8
SRQ Watermark:  8
CQ Watermark:   176
MR Watermark:   548
DMABUF MR Watermark: 0
MW Watermark:   0
AH Watermark:   25
AH HW Watermark:      25
PD Watermark:   10
Resize CQ count: 0
Recoverable Errors: 0
Rx Pkts: 63932369096
Rx Bytes: 169469560700554
Tx Pkts: 27270235729
Tx Bytes: 27378624199610
CNP Tx Pkts: 445800924
CNP Rx Pkts: 855739
RoCE Only Rx Pkts: 63931513357
RoCE Only Rx Bytes: 169469560700554
RoCE Only Tx Pkts: 26824434805
RoCE Only Tx Bytes: 27378624199610
rx_roce_error_pkts: 0
  
```

```
rx_roce_discard_pkts: 1396189
tx_roce_error_pkts: 0
tx_roce_discards_pkts: 0
res_oob_drop_count: 1053
tx_atomic_req: 0
rx_atomic_req: 0
tx_read_req: 64143744
tx_read_resp: 105106744
rx_read_req: 64306744
rx_read_resp: 64143744
tx_write_req: 5739443365
rx_write_req: 38361739553
tx_send_req: 1392653402
rx_send_req: 23472009359
rx_good_pkts: 63930117168
rx_good_bytes: 169463731589236
rx_dcn_payload_cut: 0
te_bypassed: 0
rx_ecn_marked_pkts: 445800924
max_retry_exceeded: 0
to_retransmits: 0
seq_err_naks_rcvd: 0
rnr_naks_rcvd: 0
missing_resp: 0
dup_req: 0
unrecoverable_err: 0
bad_resp_err: 0
local_qp_op_err: 0
local_protection_err: 0
mem_mgmt_op_err: 0
remote_invalid_req_err: 0
remote_access_err: 0
remote_op_err: 0
res_exceed_max: 0
res_length_mismatch: 0
res_exceeds_wqe: 0
res_opcode_err: 0
res_rx_invalid_rkey: 0
res_rx_domain_err: 0
res_rx_no_perm: 0
```

snip

```
cse@slate1:~$ rdma statistic
link bnxt_re0/1 active_pds 1 active_ahs 0 active_qps 1 active_rc_qps 0 active_ud_qps 0
active_srqs 0 active_cqs 1 active_mrs 0 active_mws 0 watermark_pds 10 watermark_ahs 25
watermark_qps 305 watermark_rc_qps 296 watermark_ud_qps 8 watermark_srqs 8 watermark_cqs
176 watermark_mrs 548 watermark_mws 0 resize_cq_cnt 0 rx_pkts 63932369096 rx_bytes
169469560700554 tx_pkts 27270235729 tx_bytes 27378624199610 recoverable_errors 0
tx_roce_errors 0 tx_roce_discards 0 rx_roce_errors 0 rx_roce_discards 1396189
to_retransmits 0 seq_err_naks_rcvd 0 max_retry_exceeded 0 rnr_naks_rcvd 0 missing_resp 0
unrecoverable_err 0 bad_resp_err 0 local_qp_op_err 0 local_protection_err 0
mem_mgmt_op_err 0 remote_invalid_req_err 0 remote_access_err 0 remote_op_err 0 dup_req 0
res_exceed_max 0 res_length_mismatch 0 res_exceeds_wqe 0 res_opcode_err 0
```

```

res_rx_invalid_rkey 0 res_rx_domain_err 0 res_rx_no_perm 0 res_rx_range_err 0
res_tx_invalid_rkey 0 res_tx_domain_err 0 res_tx_no_perm 0 res_tx_range_err 0
res_irq_oflow 0 res_unsup_opcode 0 res_unaligned_atomic 0 res_rem_inv_err 0 res_mem_err 0
res_srq_err 0 res_cmp_err 0 res_invalid_dup_rkey 0 res_wqe_format_err 0 res_cq_load_err 0
res_srq_load_err 0 res_tx_pci_err 0 res_rx_pci_err 0 tx_atomic_req 0 tx_read_req 64143744
tx_read_resp 105106744 tx_write_req 5739443365 tx_send_req 1392653402 tx_roce_only_pkts
26824434805 tx_roce_only_bytes 27378624199610 rx_atomic_req 0 rx_read_req 64306744
rx_read_resp 64143744 rx_write_req 38361739553 rx_send_req 23472009359 rx_roce_only_pkts
63931513357 rx_roce_only_bytes 169469560700554 rx_good_pkts 63930117168 rx_good_bytes
169463731589236 rx_out_of_buffer 1053 tx_cnp_pkts 445800924 rx_cnp_pkts 855739
rx_ecn_marked_pkts 445800924 oos_drop_count 22420 pacing_reschedule 0 pacing_complete 0
pacing_alerts 0 db_fifo_register 2147450881

```

snip

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.

```

cse@slate1:~$ ethtool -S ens42np0 | grep pfc
rx_pfc_frames: 0
rx_pfc_ena_frames_pri0: 0
rx_pfc_ena_frames_pri1: 0
rx_pfc_ena_frames_pri2: 0
rx_pfc_ena_frames_pri3: 0
rx_pfc_ena_frames_pri4: 0
rx_pfc_ena_frames_pri5: 0
rx_pfc_ena_frames_pri6: 0
rx_pfc_ena_frames_pri7: 0
tx_pfc_frames: 0
tx_pfc_ena_frames_pri0: 0
tx_pfc_ena_frames_pri1: 0
tx_pfc_ena_frames_pri2: 0
tx_pfc_ena_frames_pri3: 0
tx_pfc_ena_frames_pri4: 0
tx_pfc_ena_frames_pri5: 0
tx_pfc_ena_frames_pri6: 0
tx_pfc_ena_frames_pri7: 0

```

snip

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).

```

cse@slate3:~$ ip addr show ens42np0.1000
17: ens42np0.1000@ens42np0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue state
UP group default qlen 1000

```

```

link/ether 90:5a:08:26:41:c0 brd ff:ff:ff:ff:ff:ff
inet6 fd00:1:1:1:0:1:0:2/96 scope global
    valid_lft forever preferred_lft forever
inet6 fe80::925a:8ff:fe26:41c0/64 scope link
    valid_lft forever preferred_lft forever
cse@slate3:~$
cse@slate3:~$
cse@slate3:~$ ip -6 rule show
0:      from all lookup local
32758:  from fd00:1:8:1:0:1:0:2/96 lookup 108 proto static
32759:  from fd00:1:7:1:0:1:0:2/96 lookup 107 proto static
32760:  from fd00:1:5:1:0:1:0:2/96 lookup 105 proto static
32761:  from fd00:1:6:1:0:1:0:2/96 lookup 106 proto static
32762:  from fd00:1:4:1:0:1:0:2/96 lookup 104 proto static
32763:  from fd00:1:3:1:0:1:0:2/96 lookup 103 proto static
32764:  from fd00:1:1:1:0:1:0:2/96 lookup 101 proto static
32765:  from fd00:1:2:1:0:1:0:2/96 lookup 102 proto static
32766:  from all lookup main
cse@slate3:~$
cse@slate3:~$ ip -6 rule show table 101
32764:  from fd00:1:1:1:0:1:0:2/96 lookup 101 proto static
cse@slate3:~$

```

12 Performance tests

| RCCL and ROCEv2 perf test | |
|---|--|
| ib_send_bw for different packet sizes 256 to 4096 bytes | 256 - 234.40 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 | 390.4 Gb/s |
| ib_write_bw average for all interfaces – RoCEv2 4096 byte packets | 390.4 Gb/s |
| RCCL performance per algorithm | 4-node AllReduce – 361.96 GB/s 4-node All-to-All – 60.66 GB/s 4-node ReduceScatter – 360.17 GB/s 4-node Broadcast – 341.02 GB/s |

12.1 RCCL operations test

RCCL tests, using mpirun, are used to measure collective communication performance within a single node (a single AMD MI300X server) and across multiple nodes (up to four AMD MI300X 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 $N \times (N-1)$ 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.

12.1.1 Test results for AllReduce collective

```
cse@slate1:~/benchmarking/mpi-tests$ cat hostfile.txt
slate1.ncse.io slots=8
slate2.ncse.io slots=8
slate3.ncse.io slots=8
slate4.ncse.io slots=8

cse@slate1:~/benchmarking/mpi-tests$ cat run-it0-new.sh
#!/bin/bash -x

export INSTALL_DIR=$HOME/ompi_for_gpu
export UCX_DIR=$INSTALL_DIR/ucx
export OMPI_DIR=$INSTALL_DIR/ompi
export RCCL_HOME=/home/cse/benchmarking/rccl-tests
export BUILD_DIR=/tmp/ompi_for_gpu_build
export LD_LIBRARY_PATH=$OMPI_DIR/lib:$UCX_DIR/lib:/opt/rocm/lib
export PATH=$OMPI_DIR/bin:$UCX_DIR/bin:$PATH

echo "LD Path -> $LD_LIBRARY_PATH"

export NCCL_IB_GID_INDEX=5

    /home/cse/ompi_for_gpu/ompi/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt \
    -x PATH=$PATH \
    -x LD_LIBRARY_PATH=$LD_LIBRARY_PATH \
    -x NCCL_SOCKET_IFNAME=ens50f1 \
    -x NCCL_IB_GID_INDEX=5 \
    -x UCX_IB_GID_INDEX=5 \
    -x
NCCL_IB_HCA=bnxt_re0,bnxt_re1,bnxt_re4,bnxt_re5,bnxt_re6,bnxt_re7,bnxt_re8,bnxt_re9 \
-x
UCX_NET_DEVICES=bnxt_re0:1,bnxt_re1:1,bnxt_re4:1,bnxt_re5:1,bnxt_re6:1,bnxt_re7:1,bnxt_re8
:1,bnxt_re9:1 \
    -x HIP_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 \
    -x NCCL_IB_PCI_RELAXED_ORDERING=1 \
    -x HSA_DISABLE_CACHE=1 \
    -x HSA_FORCE_FINE_GRAIN_PCIE=1 \
    -x NCCL_IB_TIMEOUT=22 \
    -x NCCL_IB_DISABLE=0 \
    --bind-to numa \
```

```

--mca pml ucx \
--mca osc ucx \
--mca spml ucx \
--mca btl ^vader,openib \
--mca btl_tcp_if_include ens50f1 \
/home/cse/benchmarking/rccl-tests/build/all_reduce_perf -b 8 -e 16G -f 2 -i 0 -g 1
#-x NCCL_ALGO=Ring \
#-x NCCL_PROTO=Simple \
#-x NCCL_DEBUG=INFO \
#-x NCCL_SOCKET_IFNAME=ens50f1\
#-x
UCX_NET_DEVICES=bnxt_re0:1,bnxt_re1:1,bnxt_re4:1,bnxt_re5:1,bnxt_re6:1,bnxt_re7:1,bnxt_re8
:1,bnxt_re9:1 \

cse@slate1:~/benchmarking/mpi-tests$ ./run-it0-new.sh
+ export INSTALL_DIR=/home/cse/mpi_for_gpu
+ INSTALL_DIR=/home/cse/mpi_for_gpu
+ export UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ export OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ export RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ export BUILD_DIR=/tmp/mpi_for_gpu_build
+ BUILD_DIR=/tmp/mpi_for_gpu_build
+ export
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/l
ib
+
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/l
ib
+ export
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/l
ocal/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-
6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/l
ocal/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-
6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+ echo 'LD Path ->
/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib'
LD Path -> /home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export NCCL_IB_GID_INDEX=5
+ NCCL_IB_GID_INDEX=5
+ /home/cse/mpi_for_gpu/mpi/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/l
ocal/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-
6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin -x
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/l
ib -x NCCL_SOCKET_IFNAME=ens50f1 -x NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x
NCCL_IB_HCA=bnxt_re0,bnxt_re1,bnxt_re4,bnxt_re5,bnxt_re6,bnxt_re7,bnxt_re8,bnxt_re9 -x
UCX_NET_DEVICES=bnxt_re0:1,bnxt_re1:1,bnxt_re4:1,bnxt_re5:1,bnxt_re6:1,bnxt_re7:1,bnxt_re8
:1,bnxt_re9:1 -x HIP_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 -x NCCL_IB_PCI_RELAXED_ORDERING=1 -x
HSA_DISABLE_CACHE=1 -x HSA_FORCE_FINE_GRAIN_PCIE=1 -x NCCL_IB_TIMEOUT=22 -x

```

```

NCCL_IB_DISABLE=0 --bind-to numa --mca pml ucx --mca osc ucx --mca spml ucx --mca btl
'^vader,openib' --mca btl_tcp_if_include ens50f1 /home/cse/benchmarking/rccl-
tests/build/all_reduce_perf -b 8 -e 16G -f 2 -i 0 -g 1
[1762249869.299643] [slate1:68620:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.299643] [slate1:68620:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.324455] [slate1:68622:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.324455] [slate1:68622:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.333485] [slate1:68623:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.333485] [slate1:68623:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.344911] [slate1:68626:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.344911] [slate1:68626:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.356840] [slate1:68621:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.356840] [slate1:68621:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.357686] [slate1:68627:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.357686] [slate1:68627:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.357746] [slate1:68624:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.357746] [slate1:68624:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1762249869.357774] [slate1:68625:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1762249869.357774] [slate1:68625:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
# Collective test starting: all_reduce_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 5 iters:
20 agg iters: 1 validation: 1 graph: 0
#
rccl-tests: Version develop:e1b8a3a
# Using devices
# Rank 0 Group 0 Pid 68620 on slate1 device 0 [0000:05:00] AMD Instinct MI300X
# Rank 1 Group 0 Pid 68621 on slate1 device 1 [0000:27:00] AMD Instinct MI300X
# Rank 2 Group 0 Pid 68622 on slate1 device 2 [0000:47:00] AMD Instinct MI300X
# Rank 3 Group 0 Pid 68623 on slate1 device 3 [0000:65:00] AMD Instinct MI300X
# Rank 4 Group 0 Pid 68624 on slate1 device 4 [0000:85:00] AMD Instinct MI300X
# Rank 5 Group 0 Pid 68625 on slate1 device 5 [0000:a7:00] AMD Instinct MI300X
# Rank 6 Group 0 Pid 68626 on slate1 device 6 [0000:c7:00] AMD Instinct MI300X
# Rank 7 Group 0 Pid 68627 on slate1 device 7 [0000:e5:00] AMD Instinct MI300X
# Rank 8 Group 0 Pid 41627 on slate2 device 0 [0000:05:00] AMD Instinct MI300X
# Rank 9 Group 0 Pid 41628 on slate2 device 1 [0000:27:00] AMD Instinct MI300X
# Rank 10 Group 0 Pid 41629 on slate2 device 2 [0000:47:00] AMD Instinct MI300X
# Rank 11 Group 0 Pid 41632 on slate2 device 3 [0000:65:00] AMD Instinct MI300X
# Rank 12 Group 0 Pid 41630 on slate2 device 4 [0000:85:00] AMD Instinct MI300X

```

| | | | | | | | | | | | | | |
|--------------|---------|------------|------------|-------|--------|--------|-------|--------------|--------|----------|--------|--------|--------|
| # | Rank 13 | Group 0 | Pid 41631 | on | slate2 | device | 5 | [0000:a7:00] | AMD | Instinct | MI300X | | |
| # | Rank 14 | Group 0 | Pid 41633 | on | slate2 | device | 6 | [0000:c7:00] | AMD | Instinct | MI300X | | |
| # | Rank 15 | Group 0 | Pid 41634 | on | slate2 | device | 7 | [0000:e5:00] | AMD | Instinct | MI300X | | |
| # | Rank 16 | Group 0 | Pid 230049 | on | slate3 | device | 0 | [0000:05:00] | AMD | Instinct | MI300X | | |
| # | Rank 17 | Group 0 | Pid 230048 | on | slate3 | device | 1 | [0000:27:00] | AMD | Instinct | MI300X | | |
| # | Rank 18 | Group 0 | Pid 230050 | on | slate3 | device | 2 | [0000:47:00] | AMD | Instinct | MI300X | | |
| # | Rank 19 | Group 0 | Pid 230051 | on | slate3 | device | 3 | [0000:65:00] | AMD | Instinct | MI300X | | |
| # | Rank 20 | Group 0 | Pid 230054 | on | slate3 | device | 4 | [0000:85:00] | AMD | Instinct | MI300X | | |
| # | Rank 21 | Group 0 | Pid 230052 | on | slate3 | device | 5 | [0000:a7:00] | AMD | Instinct | MI300X | | |
| # | Rank 22 | Group 0 | Pid 230053 | on | slate3 | device | 6 | [0000:c7:00] | AMD | Instinct | MI300X | | |
| # | Rank 23 | Group 0 | Pid 230055 | on | slate3 | device | 7 | [0000:e5:00] | AMD | Instinct | MI300X | | |
| # | Rank 24 | Group 0 | Pid 174264 | on | slate4 | device | 0 | [0000:05:00] | AMD | Instinct | MI300X | | |
| # | Rank 25 | Group 0 | Pid 174265 | on | slate4 | device | 1 | [0000:27:00] | AMD | Instinct | MI300X | | |
| # | Rank 26 | Group 0 | Pid 174266 | on | slate4 | device | 2 | [0000:47:00] | AMD | Instinct | MI300X | | |
| # | Rank 27 | Group 0 | Pid 174269 | on | slate4 | device | 3 | [0000:65:00] | AMD | Instinct | MI300X | | |
| # | Rank 28 | Group 0 | Pid 174267 | on | slate4 | device | 4 | [0000:85:00] | AMD | Instinct | MI300X | | |
| # | Rank 29 | Group 0 | Pid 174268 | on | slate4 | device | 5 | [0000:a7:00] | AMD | Instinct | MI300X | | |
| # | Rank 30 | Group 0 | Pid 174270 | on | slate4 | device | 6 | [0000:c7:00] | AMD | Instinct | MI300X | | |
| # | Rank 31 | Group 0 | Pid 174271 | on | slate4 | device | 7 | [0000:e5:00] | AMD | Instinct | MI300X | | |
| # | | | | | | | | | | | | | |
| # | | | | | | | | | | | | | |
| 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 | 63.15 | 0.00 | 0.00 | 0 | | | | |
| 63.47 | 0.00 | 0.00 | 0 | | | | | | | | | | |
| | 16 | 4 | float | sum | -1 | 62.41 | 0.00 | 0.00 | 0 | | | | |
| 62.79 | 0.00 | 0.00 | 0 | | | | | | | | | | |
| | 32 | 8 | float | sum | -1 | 61.85 | 0.00 | 0.00 | 0 | | | | |
| 62.44 | 0.00 | 0.00 | 0 | | | | | | | | | | |
| | 64 | 16 | float | sum | -1 | 61.43 | 0.00 | 0.00 | 0 | | | | |
| 65.50 | 0.00 | 0.00 | 0 | | | | | | | | | | |
| | 128 | 32 | float | sum | -1 | 61.98 | 0.00 | 0.00 | 0 | | | | |
| 61.69 | 0.00 | 0.00 | 0 | | | | | | | | | | |
| | 256 | 64 | float | sum | -1 | 62.36 | 0.00 | 0.01 | 0 | | | | |
| 62.56 | 0.00 | 0.01 | 0 | | | | | | | | | | |
| | 512 | 128 | float | sum | -1 | 63.11 | 0.01 | 0.02 | 0 | | | | |
| 63.76 | 0.01 | 0.02 | 0 | | | | | | | | | | |
| | 1024 | 256 | float | sum | -1 | 63.67 | 0.02 | 0.03 | 0 | | | | |
| 64.06 | 0.02 | 0.03 | 0 | | | | | | | | | | |
| | 2048 | 512 | float | sum | -1 | 68.32 | 0.03 | 0.06 | 0 | | | | |
| 68.48 | 0.03 | 0.06 | 0 | | | | | | | | | | |
| | 4096 | 1024 | float | sum | -1 | 71.47 | 0.06 | 0.11 | 0 | | | | |
| 69.93 | 0.06 | 0.11 | 0 | | | | | | | | | | |
| | 8192 | 2048 | float | sum | -1 | 71.88 | 0.11 | 0.22 | 0 | | | | |
| 70.38 | 0.12 | 0.23 | 0 | | | | | | | | | | |
| | 16384 | 4096 | float | sum | -1 | 75.87 | 0.22 | 0.42 | 0 | | | | |
| 73.01 | 0.22 | 0.43 | 0 | | | | | | | | | | |
| | 32768 | 8192 | float | sum | -1 | 78.15 | 0.42 | 0.81 | 0 | | | | |
| 75.43 | 0.43 | 0.84 | 0 | | | | | | | | | | |
| | 65536 | 16384 | float | sum | -1 | 84.93 | 0.77 | 1.50 | 0 | | | | |
| 81.20 | 0.81 | 1.56 | 0 | | | | | | | | | | |

| | | | | | | | | | |
|---|------------|--------|-----|----|--------|--------|--------|---|--|
| 131072 | 32768 | float | sum | -1 | 89.07 | 1.47 | 2.85 | 0 | |
| 85.72 | 1.53 | 2.96 | 0 | | | | | | |
| 262144 | 65536 | float | sum | -1 | 96.08 | 2.73 | 5.29 | 0 | |
| 98.12 | 2.67 | 5.18 | 0 | | | | | | |
| 524288 | 131072 | float | sum | -1 | 134.6 | 3.89 | 7.55 | 0 | |
| 134.9 | 3.89 | 7.53 | 0 | | | | | | |
| 1048576 | 262144 | float | sum | -1 | 206.4 | 5.08 | 9.84 | 0 | |
| 215.4 | 4.87 | 9.43 | 0 | | | | | | |
| 2097152 | 524288 | float | sum | -1 | 202.1 | 10.38 | 20.11 | 0 | |
| 202.2 | 10.37 | 20.10 | 0 | | | | | | |
| 4194304 | 1048576 | float | sum | -1 | 239.8 | 17.49 | 33.89 | 0 | |
| 241.6 | 17.36 | 33.64 | 0 | | | | | | |
| 8388608 | 2097152 | float | sum | -1 | 333.4 | 25.16 | 48.76 | 0 | |
| 329.4 | 25.47 | 49.35 | 0 | | | | | | |
| 16777216 | 4194304 | float | sum | -1 | 451.5 | 37.16 | 71.99 | 0 | |
| 451.3 | 37.17 | 72.03 | 0 | | | | | | |
| 33554432 | 8388608 | float | sum | -1 | 762.1 | 44.03 | 85.31 | 0 | |
| 760.0 | 44.15 | 85.55 | 0 | | | | | | |
| 67108864 | 16777216 | float | sum | -1 | 910.6 | 73.70 | 142.79 | 0 | |
| 908.3 | 73.89 | 143.15 | 0 | | | | | | |
| 134217728 | 33554432 | float | sum | -1 | 1544.1 | 86.93 | 168.42 | 0 | |
| 1544.6 | 86.89 | 168.36 | 0 | | | | | | |
| 268435456 | 67108864 | float | sum | -1 | 2855.5 | 94.01 | 182.14 | 0 | |
| 2816.5 | 95.31 | 184.66 | 0 | | | | | | |
| 536870912 | 134217728 | float | sum | -1 | 3124.4 | 171.83 | 332.93 | 0 | |
| 3106.9 | 172.80 | 334.80 | 0 | | | | | | |
| 1073741824 | 268435456 | float | sum | -1 | 6011.3 | 178.62 | 346.08 | 0 | |
| 6008.3 | 178.71 | 346.25 | 0 | | | | | | |
| 2147483648 | 536870912 | float | sum | -1 | 11692 | 183.66 | 355.85 | 0 | |
| 11687 | 183.74 | 356.01 | 0 | | | | | | |
| 4294967296 | 1073741824 | float | sum | -1 | 23359 | 183.87 | 356.25 | 0 | |
| 23342 | 184.00 | 356.51 | 0 | | | | | | |
| 8589934592 | 2147483648 | float | sum | -1 | 46086 | 186.39 | 361.13 | 0 | |
| 46308 | 185.50 | 359.40 | 0 | | | | | | |
| 17179869184 | 4294967296 | float | sum | -1 | 91960 | 186.82 | 361.96 | 0 | |
| 91929 | 186.88 | 362.08 | 0 | | | | | | |
| # Errors with asterisks indicate errors that have exceeded the maximum threshold. | | | | | | | | | |
| # Out of bounds values : 0 OK | | | | | | | | | |
| # Avg bus bandwidth : 90.5714 | | | | | | | | | |
| # | | | | | | | | | |
| # Collective test concluded: all_reduce_perf | | | | | | | | | |

12.1.2 Test results for ReduceScatter collective

```
cse@slate1:~/benchmarking/mpi-tests$ ./run-it0-new.sh
+ export INSTALL_DIR=/home/cse/mpi_for_gpu
+ INSTALL_DIR=/home/cse/mpi_for_gpu
+ export UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ export OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ export RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ RCCL_HOME=/home/cse/benchmarking/rccl-tests
```

```

+ export BUILD_DIR=/tmp/mpi_for_gpu_build
+ BUILD_DIR=/tmp/mpi_for_gpu_build
+ export
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+ echo 'LD Path ->
/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib'
LD Path -> /home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export NCCL_IB_GID_INDEX=5
+ NCCL_IB_GID_INDEX=5
+ /home/cse/mpi_for_gpu/mpi/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin -x
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib -x
NCCL_SOCKET_IFNAME=ens50f1 -x NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x
NCCL_IB_HCA=bnxt_re0,bnxt_re1,bnxt_re4,bnxt_re5,bnxt_re6,bnxt_re7,bnxt_re8,bnxt_re9 -x
UCX_NET_DEVICES=bnxt_re0:1,bnxt_re1:1,bnxt_re4:1,bnxt_re5:1,bnxt_re6:1,bnxt_re7:1,bnxt_re8:1,bnxt_re9:1 -x
HIP_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 -x NCCL_IB_PCI_RELAXED_ORDERING=1 -x
HSA_DISABLE_CACHE=1 -x HSA_FORCE_FINE_GRAIN_PCIE=1 -x NCCL_IB_TIMEOUT=22 -x
NCCL_IB_DISABLE=0 --bind-to numa --mca pml ucx --mca osc ucx --mca spml ucx --mca btl '^vader,openib' --mca btl_tcp_if_include ens50f1 /home/cse/benchmarking/rccl-tests/build/reduce_scatter_perf -b 8 -e 16G -f 2 -i 0 -g 1
[1763097786.146898] [slate1:353945:0] parser.c:2036 UCX WARN unused environment variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.146898] [slate1:353945:0] parser.c:2036 UCX WARN (set UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097786.147497] [slate1:353947:0] parser.c:2036 UCX WARN unused environment variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.147497] [slate1:353947:0] parser.c:2036 UCX WARN (set UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097786.148133] [slate1:353943:0] parser.c:2036 UCX WARN unused environment variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.148133] [slate1:353943:0] parser.c:2036 UCX WARN (set UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097786.157079] [slate1:353944:0] parser.c:2036 UCX WARN unused environment variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.157079] [slate1:353944:0] parser.c:2036 UCX WARN (set UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097786.157188] [slate1:353946:0] parser.c:2036 UCX WARN unused environment variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.157188] [slate1:353946:0] parser.c:2036 UCX WARN (set UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)

```

```

[1763097786.162201] [slate1:353940:0]          parser.c:2036 UCX  WARN  unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.162201] [slate1:353940:0]          parser.c:2036 UCX  WARN  (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097786.171561] [slate1:353941:0]          parser.c:2036 UCX  WARN  unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.171561] [slate1:353941:0]          parser.c:2036 UCX  WARN  (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097786.173934] [slate1:353942:0]          parser.c:2036 UCX  WARN  unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097786.173934] [slate1:353942:0]          parser.c:2036 UCX  WARN  (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
# Collective test starting: reduce_scatter_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 5 iters:
20 agg iters: 1 validation: 1 graph: 0
#
rccl-tests: Version develop:e1b8a3a
# Using devices
# Rank  0 Group  0 Pid 353940 on    slate1 device  0 [0000:05:00] AMD Instinct MI300X
# Rank  1 Group  0 Pid 353941 on    slate1 device  1 [0000:27:00] AMD Instinct MI300X
# Rank  2 Group  0 Pid 353942 on    slate1 device  2 [0000:47:00] AMD Instinct MI300X
# Rank  3 Group  0 Pid 353943 on    slate1 device  3 [0000:65:00] AMD Instinct MI300X
# Rank  4 Group  0 Pid 353944 on    slate1 device  4 [0000:85:00] AMD Instinct MI300X
# Rank  5 Group  0 Pid 353945 on    slate1 device  5 [0000:a7:00] AMD Instinct MI300X
# Rank  6 Group  0 Pid 353946 on    slate1 device  6 [0000:c7:00] AMD Instinct MI300X
# Rank  7 Group  0 Pid 353947 on    slate1 device  7 [0000:e5:00] AMD Instinct MI300X
# Rank  8 Group  0 Pid 287617 on    slate2 device  0 [0000:05:00] AMD Instinct MI300X
# Rank  9 Group  0 Pid 287616 on    slate2 device  1 [0000:27:00] AMD Instinct MI300X
# Rank 10 Group  0 Pid 287619 on    slate2 device  2 [0000:47:00] AMD Instinct MI300X
# Rank 11 Group  0 Pid 287620 on    slate2 device  3 [0000:65:00] AMD Instinct MI300X
# Rank 12 Group  0 Pid 287621 on    slate2 device  4 [0000:85:00] AMD Instinct MI300X
# Rank 13 Group  0 Pid 287618 on    slate2 device  5 [0000:a7:00] AMD Instinct MI300X
# Rank 14 Group  0 Pid 287622 on    slate2 device  6 [0000:c7:00] AMD Instinct MI300X
# Rank 15 Group  0 Pid 287623 on    slate2 device  7 [0000:e5:00] AMD Instinct MI300X
# Rank 16 Group  0 Pid 456461 on    slate3 device  0 [0000:05:00] AMD Instinct MI300X
# Rank 17 Group  0 Pid 456462 on    slate3 device  1 [0000:27:00] AMD Instinct MI300X
# Rank 18 Group  0 Pid 456464 on    slate3 device  2 [0000:47:00] AMD Instinct MI300X
# Rank 19 Group  0 Pid 456465 on    slate3 device  3 [0000:65:00] AMD Instinct MI300X
# Rank 20 Group  0 Pid 456463 on    slate3 device  4 [0000:85:00] AMD Instinct MI300X
# Rank 21 Group  0 Pid 456466 on    slate3 device  5 [0000:a7:00] AMD Instinct MI300X
# Rank 22 Group  0 Pid 456467 on    slate3 device  6 [0000:c7:00] AMD Instinct MI300X
# Rank 23 Group  0 Pid 456468 on    slate3 device  7 [0000:e5:00] AMD Instinct MI300X
# Rank 24 Group  0 Pid 410491 on    slate4 device  0 [0000:05:00] AMD Instinct MI300X
# Rank 25 Group  0 Pid 410492 on    slate4 device  1 [0000:27:00] AMD Instinct MI300X
# Rank 26 Group  0 Pid 410493 on    slate4 device  2 [0000:47:00] AMD Instinct MI300X
# Rank 27 Group  0 Pid 410495 on    slate4 device  3 [0000:65:00] AMD Instinct MI300X
# Rank 28 Group  0 Pid 410494 on    slate4 device  4 [0000:85:00] AMD Instinct MI300X
# Rank 29 Group  0 Pid 410497 on    slate4 device  5 [0000:a7:00] AMD Instinct MI300X
# Rank 30 Group  0 Pid 410496 on    slate4 device  6 [0000:c7:00] AMD Instinct MI300X
# Rank 31 Group  0 Pid 410498 on    slate4 device  7 [0000:e5:00] AMD Instinct MI300X
#
#
# out-of-place
in-place

```

| # | size | count | type | redop | root | time | algbw | busbw | #wrong | |
|--------|--------|------------|--------|-------|------|------|--------|--------|--------|---|
| time | algbw | busbw | #wrong | | | (us) | (GB/s) | (GB/s) | | |
| # | (B) | (elements) | | | | | | | | |
| (us) | (GB/s) | (GB/s) | | | | | | | | |
| 0.09 | 0.00 | 0.00 | 0 | float | sum | -1 | 0.19 | 0.00 | 0.00 | 0 |
| 0.08 | 0.00 | 0.00 | 0 | float | sum | -1 | 0.08 | 0.00 | 0.00 | 0 |
| 0.09 | 0.00 | 0.00 | 0 | float | sum | -1 | 0.09 | 0.00 | 0.00 | 0 |
| 0.08 | 0.00 | 0.00 | 0 | float | sum | -1 | 0.09 | 0.00 | 0.00 | 0 |
| 0.08 | 0.00 | 0.00 | 0 | float | sum | -1 | 0.08 | 0.00 | 0.00 | 0 |
| 0.08 | 0.00 | 0.00 | 0 | float | sum | -1 | 0.08 | 0.00 | 0.00 | 0 |
| 97.39 | 0.01 | 0.01 | 0 | float | sum | -1 | 98.60 | 0.01 | 0.01 | 0 |
| 97.61 | 0.01 | 0.01 | 0 | float | sum | -1 | 97.33 | 0.01 | 0.01 | 0 |
| 98.08 | 0.02 | 0.02 | 0 | float | sum | -1 | 97.47 | 0.02 | 0.02 | 0 |
| 97.18 | 0.04 | 0.04 | 0 | float | sum | -1 | 97.76 | 0.04 | 0.04 | 0 |
| 97.92 | 0.08 | 0.08 | 0 | float | sum | -1 | 97.28 | 0.08 | 0.08 | 0 |
| 98.05 | 0.17 | 0.16 | 0 | float | sum | -1 | 98.63 | 0.17 | 0.16 | 0 |
| 100.0 | 0.33 | 0.32 | 0 | float | sum | -1 | 100.3 | 0.33 | 0.32 | 0 |
| 102.1 | 0.64 | 0.62 | 0 | float | sum | -1 | 101.5 | 0.65 | 0.63 | 0 |
| 102.6 | 1.28 | 1.24 | 0 | float | sum | -1 | 102.0 | 1.29 | 1.25 | 0 |
| 102.9 | 2.55 | 2.47 | 0 | float | sum | -1 | 104.5 | 2.51 | 2.43 | 0 |
| 152.5 | 3.44 | 3.33 | 0 | float | sum | -1 | 114.2 | 4.59 | 4.45 | 0 |
| 180.5 | 5.81 | 5.63 | 0 | float | sum | -1 | 231.9 | 4.52 | 4.38 | 0 |
| 300.2 | 6.99 | 6.77 | 0 | float | sum | -1 | 302.6 | 6.93 | 6.71 | 0 |
| 561.7 | 7.47 | 7.23 | 0 | float | sum | -1 | 564.2 | 7.43 | 7.20 | 0 |
| 398.3 | 21.06 | 20.40 | 0 | float | sum | -1 | 403.0 | 20.82 | 20.17 | 0 |
| 409.8 | 40.94 | 39.66 | 0 | float | sum | -1 | 409.5 | 40.97 | 39.69 | 0 |
| 426.2 | 78.72 | 76.26 | 0 | float | sum | -1 | 427.3 | 78.52 | 76.06 | 0 |
| 528.4 | 127.01 | 123.05 | 0 | float | sum | -1 | 529.2 | 126.80 | 122.84 | 0 |
| 1152.7 | 116.44 | 112.80 | 0 | float | sum | -1 | 1154.7 | 116.24 | 112.61 | 0 |

```

268435456      2097152      float      sum      -1      1229.6      218.31      211.49      0
1229.5  218.34  211.51      0
536870912      4194304      float      sum      -1      1529.0      351.13      340.16      0
1517.2  353.86  342.81      0
1073741824     8388608      float      sum      -1      2937.8      365.49      354.07      0
2837.8  378.37  366.54      0
2147483648     16777216     float      sum      -1      5846.2      367.33      355.85      0
5590.6  384.12  372.12      0
4294967296     33554432     float      sum      -1      11653      368.56      357.04      0
11168  384.59  372.58      0
8589934592     67108864     float      sum      -1      23196      370.32      358.75      0
22552  380.90  369.00      0
17179869184    134217728    float      sum      -1      46209      371.79      360.17      0
46214  371.75  360.13      0
# Errors with asterisks indicate errors that have exceeded the maximum threshold.
# Out of bounds values : 0 OK
# Avg bus bandwidth   : 86.4272
#
# Collective test concluded: reduce_scatter_perf

```

12.1.3 Test results for Broadcast collective

```

cse@slate1:~/benchmarking/mpi-tests$ ./run-it0-new.sh
+ export INSTALL_DIR=/home/cse/mpi_for_gpu
+ INSTALL_DIR=/home/cse/mpi_for_gpu
+ export UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ export OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ export RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ export BUILD_DIR=/tmp/mpi_for_gpu_build
+ BUILD_DIR=/tmp/mpi_for_gpu_build
+ export
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+ echo 'LD Path ->
/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib'
LD Path -> /home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export NCCL_IB_GID_INDEX=5
+ NCCL_IB_GID_INDEX=5

```

```

+ /home/cse/mpi_for_gpu/mpi/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/l
ocal/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-
6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin -x
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/l
ib -x NCCL_SOCKET_IFNAME=ens50f1 -x NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x
NCCL_IB_HCA=bnxt_re0,bnxt_re1,bnxt_re4,bnxt_re5,bnxt_re6,bnxt_re7,bnxt_re8,bnxt_re9 -x
UCX_NET_DEVICES=bnxt_re0:1,bnxt_re1:1,bnxt_re4:1,bnxt_re5:1,bnxt_re6:1,bnxt_re7:1,bnxt_re8
:1,bnxt_re9:1 -x HIP_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 -x NCCL_IB_PCI_RELAXED_ORDERING=1 -x
HSA_DISABLE_CACHE=1 -x HSA_FORCE_FINE_GRAIN_PCIE=1 -x NCCL_IB_TIMEOUT=22 -x
NCCL_IB_DISABLE=0 --bind-to numa --mca pml ucx --mca osc ucx --mca spml ucx --mca btl
'^vader,openib' --mca btl_tcp_if_include ens50f1 /home/cse/benchmarking/rccl-
tests/build/broadcast_perf -b 8 -e 16G -f 2 -i 0 -g 1
[1763097654.334241] [slate1:353581:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.334241] [slate1:353581:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.358916] [slate1:353580:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.358916] [slate1:353580:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.376851] [slate1:353583:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.376851] [slate1:353583:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.376851] [slate1:353584:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.376851] [slate1:353584:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.379198] [slate1:353582:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.379198] [slate1:353582:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.379311] [slate1:353578:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.379311] [slate1:353578:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.379405] [slate1:353579:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.379405] [slate1:353579:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1763097654.379434] [slate1:353585:0] parser.c:2036 UCX WARN unused environment
variable: UCX_DIR (maybe: UCX_TLS?)
[1763097654.379434] [slate1:353585:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
# Collective test starting: broadcast_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 5 iters:
20 agg iters: 1 validation: 1 graph: 0
#
rccl-tests: Version develop:e1b8a3a
# Using devices
# Rank 0 Group 0 Pid 353578 on slate1 device 0 [0000:05:00] AMD Instinct MI300X
# Rank 1 Group 0 Pid 353579 on slate1 device 1 [0000:27:00] AMD Instinct MI300X
# Rank 2 Group 0 Pid 353580 on slate1 device 2 [0000:47:00] AMD Instinct MI300X

```


| | | | | | | | | | | |
|---|-------------|------------|-------|------|---|--------|--------|--------|---|--|
| 16.29 | 4096 | 1024 | float | none | 0 | 16.62 | 0.25 | 0.25 | 0 | |
| | 0.25 | 0.25 | 0 | | | | | | | |
| 17.91 | 8192 | 2048 | float | none | 0 | 18.67 | 0.44 | 0.44 | 0 | |
| | 0.46 | 0.46 | 0 | | | | | | | |
| 21.28 | 16384 | 4096 | float | none | 0 | 21.62 | 0.76 | 0.76 | 0 | |
| | 0.77 | 0.77 | 0 | | | | | | | |
| 30.38 | 32768 | 8192 | float | none | 0 | 30.79 | 1.06 | 1.06 | 0 | |
| | 1.08 | 1.08 | 0 | | | | | | | |
| 42.48 | 65536 | 16384 | float | none | 0 | 43.36 | 1.51 | 1.51 | 0 | |
| | 1.54 | 1.54 | 0 | | | | | | | |
| 82.01 | 131072 | 32768 | float | none | 0 | 82.64 | 1.59 | 1.59 | 0 | |
| | 1.60 | 1.60 | 0 | | | | | | | |
| 124.1 | 262144 | 65536 | float | none | 0 | 125.6 | 2.09 | 2.09 | 0 | |
| | 2.11 | 2.11 | 0 | | | | | | | |
| 128.4 | 524288 | 131072 | float | none | 0 | 131.1 | 4.00 | 4.00 | 0 | |
| | 4.08 | 4.08 | 0 | | | | | | | |
| 152.8 | 1048576 | 262144 | float | none | 0 | 153.0 | 6.85 | 6.85 | 0 | |
| | 6.86 | 6.86 | 0 | | | | | | | |
| 238.5 | 2097152 | 524288 | float | none | 0 | 230.7 | 9.09 | 9.09 | 0 | |
| | 8.79 | 8.79 | 0 | | | | | | | |
| 285.3 | 4194304 | 1048576 | float | none | 0 | 289.6 | 14.48 | 14.48 | 0 | |
| | 14.70 | 14.70 | 0 | | | | | | | |
| 315.7 | 8388608 | 2097152 | float | none | 0 | 320.5 | 26.17 | 26.17 | 0 | |
| | 26.57 | 26.57 | 0 | | | | | | | |
| 406.8 | 16777216 | 4194304 | float | none | 0 | 407.4 | 41.18 | 41.18 | 0 | |
| | 41.24 | 41.24 | 0 | | | | | | | |
| 534.4 | 33554432 | 8388608 | float | none | 0 | 534.2 | 62.82 | 62.82 | 0 | |
| | 62.79 | 62.79 | 0 | | | | | | | |
| 693.0 | 67108864 | 16777216 | float | none | 0 | 707.2 | 94.89 | 94.89 | 0 | |
| | 96.84 | 96.84 | 0 | | | | | | | |
| 1111.6 | 134217728 | 33554432 | float | none | 0 | 1115.1 | 120.37 | 120.37 | 0 | |
| | 120.74 | 120.74 | 0 | | | | | | | |
| 1517.1 | 268435456 | 67108864 | float | none | 0 | 1515.3 | 177.15 | 177.15 | 0 | |
| | 176.94 | 176.94 | 0 | | | | | | | |
| 2844.6 | 536870912 | 134217728 | float | none | 0 | 2848.4 | 188.48 | 188.48 | 0 | |
| | 188.73 | 188.73 | 0 | | | | | | | |
| 4372.4 | 1073741824 | 268435456 | float | none | 0 | 4375.1 | 245.42 | 245.42 | 0 | |
| | 245.57 | 245.57 | 0 | | | | | | | |
| 8571.1 | 2147483648 | 536870912 | float | none | 0 | 8563.6 | 250.77 | 250.77 | 0 | |
| | 250.55 | 250.55 | 0 | | | | | | | |
| 14563 | 4294967296 | 1073741824 | float | none | 0 | 14588 | 294.41 | 294.41 | 0 | |
| | 294.91 | 294.91 | 0 | | | | | | | |
| 26543 | 8589934592 | 2147483648 | float | none | 0 | 26533 | 323.74 | 323.74 | 0 | |
| | 323.63 | 323.63 | 0 | | | | | | | |
| 50387 | 17179869184 | 4294967296 | float | none | 0 | 50377 | 341.02 | 341.02 | 0 | |
| | 340.96 | 340.96 | 0 | | | | | | | |
| # Errors with asterisks indicate errors that have exceeded the maximum threshold. | | | | | | | | | | |
| # Out of bounds values : 0 OK | | | | | | | | | | |
| # Avg bus bandwidth : 69.0747 | | | | | | | | | | |
| # | | | | | | | | | | |
| # Collective test concluded: broadcast_perf | | | | | | | | | | |

12.1.4 Test results for All-to-All collective

```

cse@slate1:~/benchmarking/mpi-tests$ ./run-it0-new.sh
+ export INSTALL_DIR=/home/cse/mpi_for_gpu
+ INSTALL_DIR=/home/cse/mpi_for_gpu
+ export UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ UCX_DIR=/home/cse/mpi_for_gpu/ucx
+ export OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ OMPI_DIR=/home/cse/mpi_for_gpu/mpi
+ export RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ RCCL_HOME=/home/cse/benchmarking/rccl-tests
+ export BUILD_DIR=/tmp/mpi_for_gpu_build
+ BUILD_DIR=/tmp/mpi_for_gpu_build
+ export
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin
+ echo 'LD Path ->
/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib'
LD Path -> /home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib
+ export NCCL_IB_GID_INDEX=5
+ NCCL_IB_GID_INDEX=5
+ /home/cse/mpi_for_gpu/mpi/bin/mpirun -np 32 -N 8 --hostfile hostfile.txt -x
PATH=/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/opt/rocm-6.4.3/bin:/home/cse/mpi_for_gpu/mpi/bin:/home/cse/mpi_for_gpu/ucx/bin -x
LD_LIBRARY_PATH=/home/cse/mpi_for_gpu/mpi/lib:/home/cse/mpi_for_gpu/ucx/lib:/opt/rocm/lib -x NCCL_SOCKET_IFNAME=ens50f1 -x NCCL_IB_GID_INDEX=5 -x UCX_IB_GID_INDEX=5 -x
NCCL_IB_HCA=bnxt_re0,bnxt_re1,bnxt_re4,bnxt_re5,bnxt_re6,bnxt_re7,bnxt_re8,bnxt_re9 -x
UCX_NET_DEVICES=bnxt_re0:1,bnxt_re1:1,bnxt_re4:1,bnxt_re5:1,bnxt_re6:1,bnxt_re7:1,bnxt_re8:1,bnxt_re9:1 -x HIP_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 -x NCCL_IB_PCI_RELAXED_ORDERING=1 -x
HSA_DISABLE_CACHE=1 -x HSA_FORCE_FINE_GRAIN_PCIE=1 -x NCCL_IB_TIMEOUT=22 -x
NCCL_IB_DISABLE=0 -x NCCL_PXN_DISABLE=0 --bind-to numa --mca pml ucx --mca osc ucx --mca spml ucx --mca btl '^vader,openib' --mca btl_tcp_if_include ens50f1
/home/cse/benchmarking/rccl-tests/build/alltoall_perf -b 8 -e 16G -f 2 -i 0 -g 1
[1764162170.162162] [slate1:1415321:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.162162] [slate1:1415321:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1764162170.162256] [slate1:1415320:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.162256] [slate1:1415320:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)

```

```

[1764162170.198662] [slate1:1415323:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.198662] [slate1:1415323:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1764162170.208744] [slate1:1415322:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.208744] [slate1:1415322:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1764162170.225705] [slate1:1415325:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.225705] [slate1:1415325:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1764162170.226568] [slate1:1415324:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.226568] [slate1:1415324:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1764162170.227741] [slate1:1415326:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.227741] [slate1:1415326:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
[1764162170.227824] [slate1:1415319:0] parser.c:2036 UCX WARN unused
environment variable: UCX_DIR (maybe: UCX_TLS?)
[1764162170.227824] [slate1:1415319:0] parser.c:2036 UCX WARN (set
UCX_WARN_UNUSED_ENV_VARS=n to suppress this warning)
# Collective test starting: alltoall_perf
# nThread 1 nGpus 1 minBytes 8 maxBytes 17179869184 step: 2(factor) warmup iters: 5 iters:
20 agg iters: 1 validation: 1 graph: 0
#
rccl-tests: Version develop:e1b8a3a
# Using devices
# Rank 0 Group 0 Pid 1415319 on slate1 device 0 [0000:05:00] AMD Instinct MI300X
# Rank 1 Group 0 Pid 1415320 on slate1 device 1 [0000:27:00] AMD Instinct MI300X
# Rank 2 Group 0 Pid 1415321 on slate1 device 2 [0000:47:00] AMD Instinct MI300X
# Rank 3 Group 0 Pid 1415322 on slate1 device 3 [0000:65:00] AMD Instinct MI300X
# Rank 4 Group 0 Pid 1415323 on slate1 device 4 [0000:85:00] AMD Instinct MI300X
# Rank 5 Group 0 Pid 1415324 on slate1 device 5 [0000:a7:00] AMD Instinct MI300X
# Rank 6 Group 0 Pid 1415325 on slate1 device 6 [0000:c7:00] AMD Instinct MI300X
# Rank 7 Group 0 Pid 1415326 on slate1 device 7 [0000:e5:00] AMD Instinct MI300X
# Rank 8 Group 0 Pid 2320617 on slate2 device 0 [0000:05:00] AMD Instinct MI300X
# Rank 9 Group 0 Pid 2320619 on slate2 device 1 [0000:27:00] AMD Instinct MI300X
# Rank 10 Group 0 Pid 2320618 on slate2 device 2 [0000:47:00] AMD Instinct MI300X
# Rank 11 Group 0 Pid 2320620 on slate2 device 3 [0000:65:00] AMD Instinct MI300X
# Rank 12 Group 0 Pid 2320621 on slate2 device 4 [0000:85:00] AMD Instinct MI300X
# Rank 13 Group 0 Pid 2320622 on slate2 device 5 [0000:a7:00] AMD Instinct MI300X
# Rank 14 Group 0 Pid 2320616 on slate2 device 6 [0000:c7:00] AMD Instinct MI300X
# Rank 15 Group 0 Pid 2320623 on slate2 device 7 [0000:e5:00] AMD Instinct MI300X
# Rank 16 Group 0 Pid 671619 on slate4 device 0 [0000:05:00] AMD Instinct MI300X
# Rank 17 Group 0 Pid 671620 on slate4 device 1 [0000:27:00] AMD Instinct MI300X
# Rank 18 Group 0 Pid 671621 on slate4 device 2 [0000:47:00] AMD Instinct MI300X
# Rank 19 Group 0 Pid 671622 on slate4 device 3 [0000:65:00] AMD Instinct MI300X
# Rank 20 Group 0 Pid 671623 on slate4 device 4 [0000:85:00] AMD Instinct MI300X
# Rank 21 Group 0 Pid 671624 on slate4 device 5 [0000:a7:00] AMD Instinct MI300X
# Rank 22 Group 0 Pid 671625 on slate4 device 6 [0000:c7:00] AMD Instinct MI300X
# Rank 23 Group 0 Pid 671626 on slate4 device 7 [0000:e5:00] AMD Instinct MI300X

```

| # | Rank | Group | 0 | Pid | 1212198 | on | slate3 | device | 0 | [0000:05:00] | AMD | Instinct | MI300X |
|--------|---------|------------|--------|-------|---------|-------|--------|--------|--------|--------------|--------------|----------|--------|
| # | Rank | Group | 0 | Pid | 1212199 | on | slate3 | device | 1 | [0000:27:00] | AMD | Instinct | MI300X |
| # | Rank | Group | 0 | Pid | 1212200 | on | slate3 | device | 2 | [0000:47:00] | AMD | Instinct | MI300X |
| # | Rank | Group | 0 | Pid | 1212201 | on | slate3 | device | 3 | [0000:65:00] | AMD | Instinct | MI300X |
| # | Rank | Group | 0 | Pid | 1212202 | on | slate3 | device | 4 | [0000:85:00] | AMD | Instinct | MI300X |
| # | Rank | Group | 0 | Pid | 1212203 | on | slate3 | device | 5 | [0000:a7:00] | AMD | Instinct | MI300X |
| # | Rank | Group | 0 | Pid | 1212204 | on | slate3 | device | 6 | [0000:c7:00] | AMD | Instinct | MI300X |
| # | Rank | Group | 0 | Pid | 1212205 | on | slate3 | device | 7 | [0000:e5:00] | AMD | Instinct | MI300X |
| # | | | | | | | | | | | out-of-place | in- | |
| place | | | | | | | | | | | | | |
| # | size | count | type | redop | root | time | algbw | busbw | #wrong | | | | |
| time | algbw | busbw | #wrong | | | (us) | (GB/s) | (GB/s) | (us) | | | | |
| # | (B) | (elements) | | | | | | | | | | | |
| (GB/s) | (GB/s) | | | | | | | | | | | | |
| | 0 | 0 | float | none | -1 | 0.15 | 0.00 | 0.00 | 0 | | | | |
| 0.06 | 0.00 | 0.00 | N/A | | | | | | | | | | |
| | 0 | 0 | float | none | -1 | 0.05 | 0.00 | 0.00 | 0 | | | | |
| 0.05 | 0.00 | 0.00 | N/A | | | | | | | | | | |
| | 0 | 0 | float | none | -1 | 0.05 | 0.00 | 0.00 | 0 | | | | |
| 0.05 | 0.00 | 0.00 | N/A | | | | | | | | | | |
| | 0 | 0 | float | none | -1 | 0.05 | 0.00 | 0.00 | 0 | | | | |
| 0.05 | 0.00 | 0.00 | N/A | | | | | | | | | | |
| | 0 | 0 | float | none | -1 | 0.05 | 0.00 | 0.00 | 0 | | | | |
| 0.05 | 0.00 | 0.00 | N/A | | | | | | | | | | |
| | 512 | 4 | float | none | -1 | 82.21 | 0.01 | 0.01 | 0 | | | | |
| 79.64 | 0.01 | 0.01 | N/A | | | | | | | | | | |
| | 1024 | 8 | float | none | -1 | 85.68 | 0.01 | 0.01 | 0 | | | | |
| 90.68 | 0.01 | 0.01 | N/A | | | | | | | | | | |
| | 2048 | 16 | float | none | -1 | 81.42 | 0.03 | 0.02 | 0 | | | | |
| 78.09 | 0.03 | 0.03 | N/A | | | | | | | | | | |
| | 4096 | 32 | float | none | -1 | 78.30 | 0.05 | 0.05 | 0 | | | | |
| 79.79 | 0.05 | 0.05 | N/A | | | | | | | | | | |
| | 8192 | 64 | float | none | -1 | 79.58 | 0.10 | 0.10 | 0 | | | | |
| 79.88 | 0.10 | 0.10 | N/A | | | | | | | | | | |
| | 16384 | 128 | float | none | -1 | 80.65 | 0.20 | 0.20 | 0 | | | | |
| 84.69 | 0.19 | 0.19 | N/A | | | | | | | | | | |
| | 32768 | 256 | float | none | -1 | 78.80 | 0.42 | 0.40 | 0 | | | | |
| 79.50 | 0.41 | 0.40 | N/A | | | | | | | | | | |
| | 65536 | 512 | float | none | -1 | 79.95 | 0.82 | 0.79 | 0 | | | | |
| 80.45 | 0.81 | 0.79 | N/A | | | | | | | | | | |
| | 131072 | 1024 | float | none | -1 | 79.95 | 1.64 | 1.59 | 0 | | | | |
| 80.10 | 1.64 | 1.59 | N/A | | | | | | | | | | |
| | 262144 | 2048 | float | none | -1 | 79.89 | 3.28 | 3.18 | 0 | | | | |
| 80.67 | 3.25 | 3.15 | N/A | | | | | | | | | | |
| | 524288 | 4096 | float | none | -1 | 79.66 | 6.58 | 6.38 | 0 | | | | |
| 81.40 | 6.44 | 6.24 | N/A | | | | | | | | | | |
| | 1048576 | 8192 | float | none | -1 | 81.65 | 12.84 | 12.44 | 0 | | | | |
| 85.67 | 12.24 | 11.86 | N/A | | | | | | | | | | |
| | 2097152 | 16384 | float | none | -1 | 135.1 | 15.53 | 15.04 | 0 | | | | |
| 129.5 | 16.19 | 15.69 | N/A | | | | | | | | | | |

| | | | | | | | | |
|---|-----------|-------|------|----|--------|-------|-------|---|
| 4194304 | 32768 | float | none | -1 | 177.4 | 23.64 | 22.90 | 0 |
| 166.2 | 25.24 | 24.45 | N/A | | | | | |
| 8388608 | 65536 | float | none | -1 | 257.7 | 32.56 | 31.54 | 0 |
| 261.4 | 32.10 | 31.09 | N/A | | | | | |
| 16777216 | 131072 | float | none | -1 | 463.6 | 36.19 | 35.06 | 0 |
| 467.4 | 35.89 | 34.77 | N/A | | | | | |
| 33554432 | 262144 | float | none | -1 | 634.6 | 52.88 | 51.22 | 0 |
| 636.1 | 52.75 | 51.11 | N/A | | | | | |
| 67108864 | 524288 | float | none | -1 | 1246.4 | 53.84 | 52.16 | 0 |
| 1212.0 | 55.37 | 53.64 | N/A | | | | | |
| 134217728 | 1048576 | float | none | -1 | 2460.6 | 54.55 | 52.84 | 0 |
| 2421.2 | 55.43 | 53.70 | N/A | | | | | |
| 268435456 | 2097152 | float | none | -1 | 7401.5 | 36.27 | 35.13 | 0 |
| 4330.9 | 61.98 | 60.04 | N/A | | | | | |
| 536870912 | 4194304 | float | none | -1 | 8531.8 | 62.93 | 60.96 | 0 |
| 10960 | 48.98 | 47.45 | N/A | | | | | |
| 1073741824 | 8388608 | float | none | -1 | 18866 | 56.91 | 55.14 | 0 |
| 19292 | 55.66 | 53.92 | N/A | | | | | |
| 2147483648 | 16777216 | float | none | -1 | 35974 | 59.70 | 57.83 | 0 |
| 37040 | 57.98 | 56.17 | N/A | | | | | |
| 4294967296 | 33554432 | float | none | -1 | 70186 | 61.19 | 59.28 | 0 |
| 70166 | 61.21 | 59.30 | N/A | | | | | |
| 8589934592 | 67108864 | float | none | -1 | 137632 | 62.41 | 60.46 | 0 |
| 137566 | 62.44 | 60.49 | N/A | | | | | |
| 17179869184 | 134217728 | float | none | -1 | 274346 | 62.62 | 60.66 | 0 |
| 275372 | 62.39 | 60.44 | N/A | | | | | |
| # Errors with asterisks indicate errors that have exceeded the maximum threshold. | | | | | | | | |
| # Out of bounds values : 0 OK | | | | | | | | |
| # Avg bus bandwidth : 21.2823 | | | | | | | | |
| # | | | | | | | | |
| # Collective test concluded: alltoall_perf | | | | | | | | |

12.2 Broadcom NIC throughput

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.

The tests are performed by first running the following command from the server perspective (on slate1, as an example):

```
ib_send_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576 -n 5000 -q 32 -F --ipv6 --ipv6-addr -m 256
```

After, the following command is run on another terminal window of the same server (thus, slate1 again), where the IPv6 address specified is the IPv6 address assigned to the interface mapped to bnxt_re0.

```
ib_send_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576 -n 5000 -q 32 -F --ipv6 --ipv6-addr fd00:2:10:1:1:1:0:2 -m 256
```

12.2.1.1 Test results for RoCE MTU size 256

```
cse@slate1:~/benchmarking/mpi-tests$ ib_send_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576
-n 5000 -q 32 -F --ipv6 --ipv6-addr -m 256
WARNING: BW peak won't be measured in this run.

*****
* Waiting for client to connect... *
*****

-----
                Send BW Test
Dual-port       : OFF           Device           : bnxt_re0
Number of qps   : 32           Transport type   : IB
Connection type : RC           Using SRQ        : OFF
PCIe relax order: ON
ibv_wr* API     : OFF
RX depth        : 512
CQ Moderation   : 1
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 0x2c29 PSN 0xb088f4
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c5a PSN 0x636505
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c6b PSN 0xab8f35
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c76 PSN 0x41c3a9
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c63 PSN 0x7e4e3f
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c66 PSN 0x49ccff
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c72 PSN 0x20b0bc
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c92 PSN 0x52f0cf
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c1a PSN 0xd6e891
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
local address: LID 0000 QPN 0x2c35 PSN 0x6f5608
GID: 253:00:00:02:00:16:00:01:00:01:00:00:00:00:02
```

```
local address: LID 0000 QPN 0x2c37 PSN 0x9a7a42
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2ccf PSN 0x268de4
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cb5 PSN 0xa7bcca
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d09 PSN 0x165478
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c53 PSN 0xfbbf0c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1e PSN 0xa56316
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2f PSN 0x53e2aa
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d27 PSN 0x162e5
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc5 PSN 0xe68b93
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d0a PSN 0xdf754d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cfb PSN 0xc4ec60
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cd6 PSN 0x1653e6
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc8 PSN 0x29611c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d11 PSN 0x12c2be
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c67 PSN 0x4c5cb1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7b PSN 0x3f3df2
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c78 PSN 0x9650f5
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c65 PSN 0x2ff69d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c8d PSN 0xae15c1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c4c PSN 0x9d2366
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c81 PSN 0x5e514d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c97 PSN 0xf165ef
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8f PSN 0x19bae9
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c5d PSN 0x75551f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c2c PSN 0x74997d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c7e PSN 0xee4788
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c21 PSN 0x8a5afb
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
```

```
remote address: LID 0000 QPN 0x2c3f PSN 0xa29441
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d2e PSN 0x41470c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2ce4 PSN 0x754415
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cd2 PSN 0x43b495
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d04 PSN 0x343ef3
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c2a PSN 0xe2551b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c13 PSN 0xf33f12
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d2c PSN 0xdcf496
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1c PSN 0x66d18b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cbd PSN 0xad9eed
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d03 PSN 0x7b68ad
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cfa PSN 0x377abe
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1f PSN 0x718ea0
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cee PSN 0xe0b8fb
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cc7 PSN 0x86adcc
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c6a PSN 0x57a13d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8a PSN 0x5170c8
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c51 PSN 0x162c8c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c80 PSN 0x6274a4
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c84 PSN 0x8e32d5
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c5b PSN 0x52367c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c06 PSN 0xa492ed
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c88 PSN 0x9d506b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c64 PSN 0x4ad9ae
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c15 PSN 0x690a18
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c50 PSN 0x1ea4e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c02 PSN 0x85fe25
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
```

| #bytes | #iterations | BW peak[Gb/sec] | BW average[Gb/sec] | MsgRate[Mpps] |
|---------|-------------|-----------------|--------------------|---------------|
| 1048576 | 160000 | 0.00 | 234.40 | 0.027943 |

12.2.1.2 Test results for RoCE MTU size 1024

```

cse@slate1:~/benchmarking/mpi-tests$ ib_send_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576
-n 5000 -q 32 -F --ipv6 --ipv6-addr -m 1024
WARNING: BW peak won't be measured in this run.

*****
* Waiting for client to connect... *
*****

-----
                Send BW Test
Dual-port      : OFF          Device       : bnxt_re0
Number of qps  : 32          Transport type : IB
Connection type : RC          Using SRQ     : OFF
PCIe relax order: ON
ibv_wr* API    : OFF
RX depth       : 512
CQ Moderation  : 1
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 0x2c8f PSN 0xf6a65d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c29 PSN 0xffcd83
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5d PSN 0xc3b411
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5a PSN 0xed908c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c6b PSN 0x8609af
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c76 PSN 0xb1aee5
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c63 PSN 0xcef6e0
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c66 PSN 0x319959
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c72 PSN 0xbf1a89
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c92 PSN 0x47dfd7
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1a PSN 0xfaae2f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c35 PSN 0xc92496

```

GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c37 PSN 0xd445ca
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2ccf PSN 0x991caf
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cb5 PSN 0xc9f541
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d09 PSN 0x8ba271
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c53 PSN 0xb62b32
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1e PSN 0xdae804
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2f PSN 0x3ba08f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d27 PSN 0xba3fd0
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc5 PSN 0x2c64f1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d0a PSN 0xac7c6c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cfb PSN 0x7f7960
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cd6 PSN 0xaea2e8
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc8 PSN 0x3efdc9
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d11 PSN 0xdfd860
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c67 PSN 0x955901
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7b PSN 0x879eef
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c78 PSN 0x48dfe2
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c65 PSN 0x14663c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c8d PSN 0xf07da2
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c4c PSN 0xe830e9
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c2c PSN 0x7fb6
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c7e PSN 0x834e68
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c21 PSN 0xab5392
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c3f PSN 0xc66079
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d2e PSN 0x217f98
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2ce4 PSN 0x9e391a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cd2 PSN 0x8d6d71

```

GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d04 PSN 0x7c5b16
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c2a PSN 0xdf6602
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c13 PSN 0x62b85c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d2c PSN 0x8c5cd0
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1c PSN 0x212523
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cbd PSN 0xc72fd3
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d03 PSN 0x96d884
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cfa PSN 0xf04cf2
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1f PSN 0xd3fece
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cee PSN 0x3d0ccb
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cc7 PSN 0x46c29
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c6a PSN 0xbe2250
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8a PSN 0xcae4fd
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c51 PSN 0x25271a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c80 PSN 0xd3fde1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c84 PSN 0x7cb631
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c5b PSN 0xd14de5
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c06 PSN 0xdd1982
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c88 PSN 0xa5dc25
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c64 PSN 0xe3f1e2
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c15 PSN 0xc4dcbc
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c50 PSN 0xde5e2b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c81 PSN 0x7cc151
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c02 PSN 0x902393
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c97 PSN 0x245e86
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02

```

```

-----
#bytes      #iterations    BW peak[Gb/sec]    BW average[Gb/sec]    MsgRate[Mpps]
1048576     160000         0.00               359.18                0.042818

```

12.2.1.3 Test results for RoCE MTU size 2048

```
cse@slate1:~/benchmarking/mpi-tests$ ib_send_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576
-n 5000 -q 32 -F --ipv6 --ipv6-addr -m 2048
WARNING: BW peak won't be measured in this run.
```

```
*****
* Waiting for client to connect... *
*****
```

```
-----
                Send BW Test
Dual-port       : OFF           Device       : bnxt_re0
Number of qps   : 32           Transport type : IB
Connection type : RC           Using SRQ     : OFF
PCIe relax order: ON
ibv_wr* API     : OFF
RX depth        : 512
CQ Moderation   : 1
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 0x2c8f PSN 0x48b9ab
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2c PSN 0xa88cf0
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c29 PSN 0x9eb044
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7e PSN 0x4e510c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5d PSN 0x217366
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5a PSN 0x15a89a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c6b PSN 0x1e73bb
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c76 PSN 0xa41162
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c63 PSN 0x29d928
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c66 PSN 0x1b7053
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c72 PSN 0xa50e31
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c92 PSN 0xe444a7
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1a PSN 0x65bfd1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
```

```
local address: LID 0000 QPN 0x2c35 PSN 0x29e873
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c37 PSN 0xa942eb
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2ccf PSN 0x94e309
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cb5 PSN 0x45af21
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d09 PSN 0x5c5b90
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c53 PSN 0xb00e62
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1e PSN 0x62170
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2f PSN 0x27a947
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d27 PSN 0xcc4c41
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc5 PSN 0xa9e1db
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d0a PSN 0x892e11
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cfb PSN 0xa8a108
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cd6 PSN 0x2280fd
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc8 PSN 0xb9bea4
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d11 PSN 0x24e420
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c67 PSN 0x1ee888
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7b PSN 0x79ac21
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c78 PSN 0x808aec
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c65 PSN 0xdcc8a2
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c21 PSN 0x4b06bf
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c3f PSN 0xdce33d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d2e PSN 0xe31043
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2ce4 PSN 0x34bdd6
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cd2 PSN 0xbcce31
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d04 PSN 0x68263f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c2a PSN 0xb2a432
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c13 PSN 0x721143
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
```

```

remote address: LID 0000 QPN 0x2d2c PSN 0xa6652b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1c PSN 0x2832d1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cbd PSN 0x829ea1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d03 PSN 0xc8f120
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cfa PSN 0xba588c
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1f PSN 0x416549
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cee PSN 0xa33ad3
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cc7 PSN 0x276d9b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c6a PSN 0x4b6814
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8a PSN 0x70e03e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c51 PSN 0x926d41
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c80 PSN 0xaf539a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c84 PSN 0x854df3
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c5b PSN 0xb77e46
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c06 PSN 0xd11f32
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c88 PSN 0x4e952
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c64 PSN 0xf634eb
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c15 PSN 0xc2dda
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c50 PSN 0xa249f3
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c81 PSN 0x9ba1f9
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8d PSN 0xc12724
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c02 PSN 0x280956
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c4c PSN 0xce4bb4
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c97 PSN 0x151a93
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02

```

```

-----
#bytes      #iterations    BW peak[Gb/sec]    BW average[Gb/sec]    MsgRate[Mpps]
1048576     160000          0.00                380.12                 0.045314
-----

```

12.2.1.4 Test results for RoCE MTU size 4096

```
cse@slate1:~/benchmarking/mpi-tests$ ib_send_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576
-n 5000 -q 32 -F --ipv6 --ipv6-addr -m 4096
WARNING: BW peak won't be measured in this run.
```

```
*****
* Waiting for client to connect... *
*****
```

```
-----
                        Send BW Test
Dual-port       : OFF          Device       : bnxt_re0
Number of qps   : 32          Transport type : IB
Connection type : RC          Using SRQ     : OFF
PCIe relax order: ON
ibv_wr* API     : OFF
RX depth       : 512
CQ Moderation   : 1
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 0x2c21 PSN 0xf7f063
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c8f PSN 0x95b611
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c3f PSN 0x6f4c87
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2c PSN 0xbf7eca
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d2e PSN 0x1c2e15
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c29 PSN 0x5ad253
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7e PSN 0xfe0b6
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5d PSN 0x336d77
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5a PSN 0xc15f4f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c6b PSN 0x910c25
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c76 PSN 0xd89f65
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c63 PSN 0xc75c94
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c66 PSN 0x8850f0
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c72 PSN 0x4dffdd
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c92 PSN 0x5203d7
```

GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1a PSN 0xdc9c4f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c35 PSN 0x5602b8
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c37 PSN 0x141012
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2ccf PSN 0x764285
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cb5 PSN 0xfc398e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d09 PSN 0xcc6ed7
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c53 PSN 0xbd575a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1e PSN 0xc684b6
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2f PSN 0x65ba86
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d27 PSN 0x14000f
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc5 PSN 0x24b42e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d0a PSN 0x4a03b7
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cfb PSN 0x9bd26d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cd6 PSN 0x4e0088
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc8 PSN 0xa070ea
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d11 PSN 0x915db8
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c67 PSN 0x35e47
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2ce4 PSN 0xdf979e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cd2 PSN 0xdaec30
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d04 PSN 0x1b31ba
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c2a PSN 0xdf781
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c13 PSN 0xb12e00
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d2c PSN 0x5ceb62
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1c PSN 0x4c0619
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cbd PSN 0xaa4a9e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d03 PSN 0x292ea
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cfa PSN 0x954724

```

GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2d1f PSN 0xbad7f8
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cee PSN 0x71852b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2cc7 PSN 0xa7433b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c6a PSN 0x4e8bcc
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8a PSN 0x9a529a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c51 PSN 0x2d5756
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c80 PSN 0x31eeb3
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c84 PSN 0x1f0bf1
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c5b PSN 0x1da78
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c06 PSN 0xf43e05
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c88 PSN 0x1d3f82
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c64 PSN 0x43d229
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c15 PSN 0x32c8d9
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c50 PSN 0xfa2f6d
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c81 PSN 0x45506a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c8d PSN 0x39aced
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c02 PSN 0xc7870a
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c7b PSN 0x8d4ec4
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c4c PSN 0xa1b93
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c78 PSN 0x27d699
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c97 PSN 0xe6e33b
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c65 PSN 0x24e90e
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02

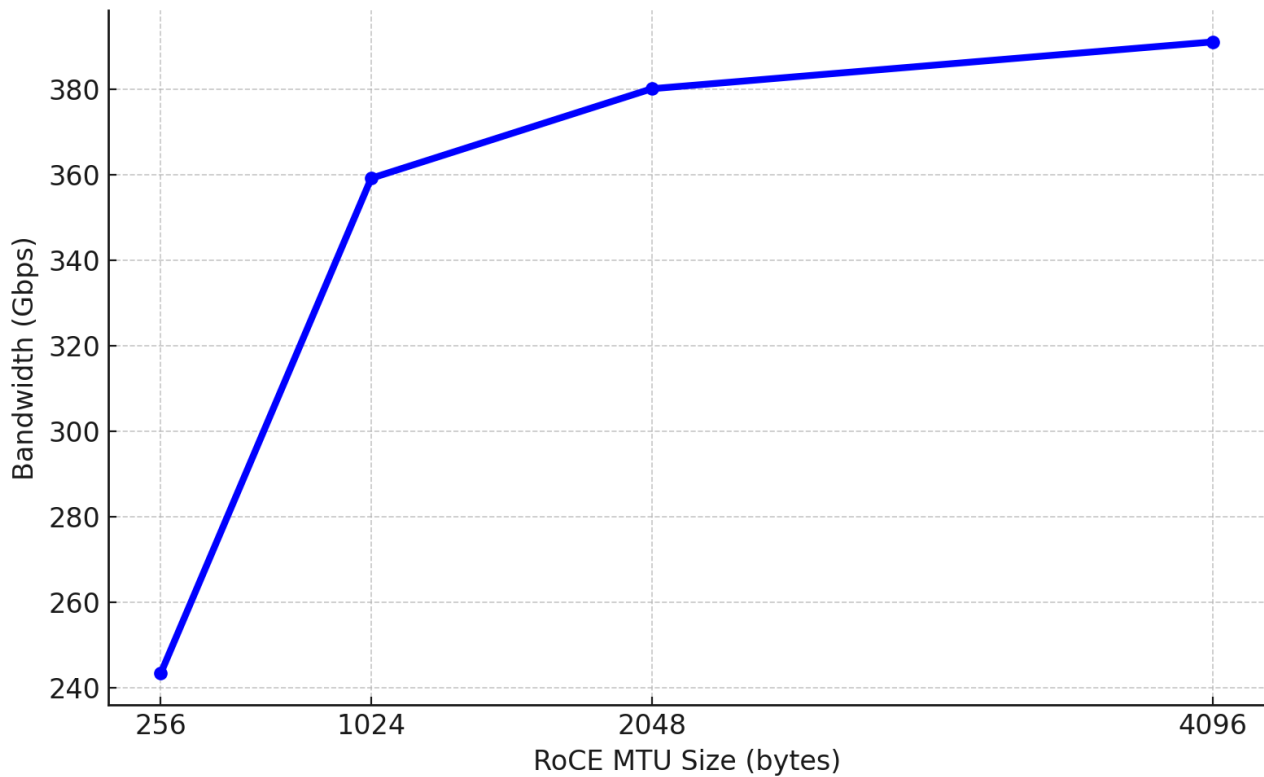
```

```

-----
#bytes      #iterations    BW peak[Gb/sec]    BW average[Gb/sec]    MsgRate[Mpps]
1048576     160000         0.00               391.08                0.046621
-----

```

The following graphic 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 server NIC and with `ib_write_bw`, the client issues RDMA WRITE operations to memory on server NIC.

12.2.2.1 Test results for RDMA `ib_write_bw`

```
cse@slate1:~/benchmarking/mpi-tests$ ib_write_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576 -n 5000 -q 32 -F --ipv6 --ipv6-addr -m 4096
```

```
*****
* Waiting for client to connect... *
*****
```

```
-----
RDMA_Write BW Test
Dual-port      : OFF      Device       : bnxt_re0
Number of qps : 32      Transport type : IB
Connection type : RC      Using SRQ    : OFF
PCIe relax order: ON
ib_wr* API    : OFF
CQ Moderation : 1
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 0x2c21 PSN 0xd47a5b RKey 0x200e304 VAddr 0x0072573e3d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2ce4 PSN 0x1144a9 RKey 0x200e304 VAddr 0x0072573e4d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c8f PSN 0x47933f RKey 0x200e304 VAddr 0x0072573e5d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cd2 PSN 0xfeeb22 RKey 0x200e304 VAddr 0x0072573e6d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c3f PSN 0x945f8d RKey 0x200e304 VAddr 0x0072573e7d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d04 PSN 0x99b26b RKey 0x200e304 VAddr 0x0072573e8d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2c PSN 0x4adaee RKey 0x200e304 VAddr 0x0072573e9d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2a PSN 0xbe474f RKey 0x200e304 VAddr 0x0072573ead0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d2e PSN 0xb83047 RKey 0x200e304 VAddr 0x0072573ebd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c13 PSN 0x6d5bd RKey 0x200e304 VAddr 0x0072573ecd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c29 PSN 0x3e051d RKey 0x200e304 VAddr 0x0072573edd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d2c PSN 0xf6fbec RKey 0x200e304 VAddr 0x0072573eed0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7e PSN 0x183968 RKey 0x200e304 VAddr 0x0072573efd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d1c PSN 0xfcaf5 RKey 0x200e304 VAddr 0x0072573f0d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5d PSN 0x6d0d0f RKey 0x200e304 VAddr 0x0072573f1d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cbd PSN 0x47d927 RKey 0x200e304 VAddr 0x0072573f2d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5a PSN 0xc8fab0 RKey 0x200e304 VAddr 0x0072573f3d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d03 PSN 0x7074aa RKey 0x200e304 VAddr 0x0072573f4d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c6b PSN 0x4da73d RKey 0x200e304 VAddr 0x0072573f5d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cfa PSN 0xcf6be6 RKey 0x200e304 VAddr 0x0072573f6d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c76 PSN 0x13ee4f RKey 0x200e304 VAddr 0x0072573f7d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d1f PSN 0x546d72 RKey 0x200e304 VAddr 0x0072573f8d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c63 PSN 0x147cee RKey 0x200e304 VAddr 0x0072573f9d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cee PSN 0x3aba5e RKey 0x200e304 VAddr 0x0072573fad0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02

```
local address: LID 0000 QPN 0x2c66 PSN 0xc0ff07 RKey 0x200e304 VAddr 0x0072573fbd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc7 PSN 0x8013c6 RKey 0x200e304 VAddr 0x0072573fcd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c72 PSN 0x34476f RKey 0x200e304 VAddr 0x0072573fdd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c6a PSN 0xd1f7c5 RKey 0x200e304 VAddr 0x0072573fed0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c92 PSN 0x8f700 RKey 0x200e304 VAddr 0x0072573ffd0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c8a PSN 0x4b3202 RKey 0x200e304 VAddr 0x007257400d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1a PSN 0xe124f0 RKey 0x200e304 VAddr 0x007257401d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c51 PSN 0x97811f RKey 0x200e304 VAddr 0x007257402d0000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2e6a PSN 0x87e6 RKey 0x2023c65 VAddr 0x007ffff579b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c4c PSN 0x21ead8 RKey 0x2023c65 VAddr 0x007ffff589b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cad PSN 0x8f42 RKey 0x2023c65 VAddr 0x007ffff599b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c01 PSN 0xf0ea69 RKey 0x2023c65 VAddr 0x007ffff5a9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cc8 PSN 0x6bdac8 RKey 0x2023c65 VAddr 0x007ffff5b9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e3a PSN 0x742c8a RKey 0x2023c65 VAddr 0x007ffff5c9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2ca7 PSN 0x9cb421 RKey 0x2023c65 VAddr 0x007ffff5d9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c85 PSN 0xb9406 RKey 0x2023c65 VAddr 0x007ffff5e9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c31 PSN 0xe64432 RKey 0x2023c65 VAddr 0x007ffff5f9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c7a PSN 0x2e72cc RKey 0x2023c65 VAddr 0x007ffff609b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c68 PSN 0xb5e80 RKey 0x2023c65 VAddr 0x007ffff619b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cae PSN 0xf8fd13 RKey 0x2023c65 VAddr 0x007ffff629b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c13 PSN 0xefc103 RKey 0x2023c65 VAddr 0x007ffff639b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2d23 PSN 0x8dc9f4 RKey 0x2023c65 VAddr 0x007ffff649b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e54 PSN 0x3fb9a2 RKey 0x2023c65 VAddr 0x007ffff659b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cac PSN 0x8b65be RKey 0x2023c65 VAddr 0x007ffff669b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c7d PSN 0xf80fb RKey 0x2023c65 VAddr 0x007ffff679b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cc3 PSN 0x430499 RKey 0x2023c65 VAddr 0x007ffff689b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2d07 PSN 0x99aa00 RKey 0x2023c65 VAddr 0x007ffff699b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
```

```

remote address: LID 0000 QPN 0x2d3f PSN 0x6bcaed RKey 0x2023c65 VAddr 0x007ffff6a9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2d1c PSN 0x7cae4a RKey 0x2023c65 VAddr 0x007ffff6b9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e6c PSN 0xcdad51 RKey 0x2023c65 VAddr 0x007ffff6c9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e64 PSN 0xab08e1 RKey 0x2023c65 VAddr 0x007ffff6d9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e62 PSN 0x14a2d5 RKey 0x2023c65 VAddr 0x007ffff6e9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c92 PSN 0x17e3b2 RKey 0x2023c65 VAddr 0x007ffff6f9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2d1a PSN 0x951295 RKey 0x2023c65 VAddr 0x007ffff709b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e46 PSN 0xe6bf92 RKey 0x2023c65 VAddr 0x007ffff719b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c1f PSN 0x290ac RKey 0x2023c65 VAddr 0x007ffff729b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e5e PSN 0xd9b5b RKey 0x2023c65 VAddr 0x007ffff739b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cc0 PSN 0xd2eec1 RKey 0x2023c65 VAddr 0x007ffff749b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cd9 PSN 0x541c43 RKey 0x2023c65 VAddr 0x007ffff759b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cd2 PSN 0x436176 RKey 0x2023c65 VAddr 0x007ffff769b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
-----
#bytes      #iterations    BW peak[Gb/sec]    BW average[Gb/sec]    MsgRate[Mpps]
1048576     160000         390.49             390.48                 0.046549
-----

```

12.2.2.2 Test results for RDMA ib_read_bw

```

cse@slate1:~/benchmarking/mpi-tests$ ib_read_bw -d bnxt_re0 -x 5 --report_gbits -s 1048576
-n 5000 -q 32 -F --ipv6 --ipv6-addr -m 4096
-----
Device not recognized to implement inline feature. Disabling it

*****
* Waiting for client to connect... *
*****
-----
RDMA_Read BW Test
Dual-port      : OFF          Device           : bnxt_re0
Number of qps  : 32           Transport type   : IB
Connection type : RC          Using SRQ        : OFF
PCIe relax order: ON
ibv_wr* API    : OFF
CQ Moderation  : 1
Mtu            : 4096[B]
Link type      : Ethernet
GID index      : 5
Outstand reads : 126

```

rdma_cm QPs : OFF
Data ex. method : Ethernet

local address: LID 0000 QPN 0x2c35 PSN 0xda65ab OUT 0x7e RKey 0x2004d04 VAddr
0x007bf34fb90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c80 PSN 0xc055b9 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf34fc90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c37 PSN 0xe6370f OUT 0x7e RKey 0x2004d04 VAddr
0x007bf34fd90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c84 PSN 0x778ab2 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf34fe90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2ccf PSN 0xbafdd OUT 0x7e RKey 0x2004d04 VAddr
0x007bf34ff90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c5b PSN 0x4ec47b OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350090000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cb5 PSN 0x5debbe OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350190000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c06 PSN 0xb74fdf OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350290000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d09 PSN 0x323597 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350390000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c88 PSN 0x80f8cd OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350490000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c53 PSN 0x7f52ed OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350590000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c64 PSN 0x6dfd7c OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350690000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c1e PSN 0xeb43b8 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350790000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c15 PSN 0xef0f05 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350890000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c2f PSN 0x6367df OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350990000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c50 PSN 0x8363b7 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350a90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d27 PSN 0x605a00 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350b90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02

local address: LID 0000 QPN 0x2c81 PSN 0x45e9ba OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350c90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc5 PSN 0x1cdf0d OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350d90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c8d PSN 0xef0f76 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350e90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d0a PSN 0x3ff29f OUT 0x7e RKey 0x2004d04 VAddr
0x007bf350f90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c02 PSN 0xb22382 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351090000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cfb PSN 0x4d61be OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351190000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c7b PSN 0xc706ee OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351290000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cd6 PSN 0x6f857 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351390000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c4c PSN 0x91ad6 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351490000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2cc8 PSN 0xa4a93f OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351590000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c78 PSN 0x4c7d55 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351690000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2d11 PSN 0xf33550 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351790000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c97 PSN 0xc39a12 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351890000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c67 PSN 0x63d3c0 OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351990000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
local address: LID 0000 QPN 0x2c65 PSN 0xacfaf OUT 0x7e RKey 0x2004d04 VAddr
0x007bf351a90000
GID: 253:00:00:02:00:16:00:01:00:01:00:01:00:00:00:02
remote address: LID 0000 QPN 0x2c17 PSN 0x6fd048 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff579b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c23 PSN 0xe3c892 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff589b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e55 PSN 0x391374 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff599b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02

```
remote address: LID 0000 QPN 0x2e68 PSN 0xde1fb3 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff5a9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e66 PSN 0x84074a OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff5b9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2d0c PSN 0x93ebe4 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff5c9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cc6 PSN 0x40973 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff5d9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e52 PSN 0xfe93f0 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff5e9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c93 PSN 0x5d76d4 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff5f9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c5b PSN 0xc48dc6 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff609b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cdb PSN 0xa976f2 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff619b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c1e PSN 0x57d19d OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff629b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2ce3 PSN 0x7c3bc5 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff639b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e3f PSN 0x805a8e OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff649b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c40 PSN 0x69a734 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff659b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e50 PSN 0x6eb8e8 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff669b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e41 PSN 0x49a5dd OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff679b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c30 PSN 0x7fc4d3 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff689b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2ccc PSN 0xa39eb2 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff699b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c55 PSN 0x98e6b7 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff6a9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cc7 PSN 0xe0ff4c OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff6b9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
```

```

remote address: LID 0000 QPN 0x2cb9 PSN 0x23f72b OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff6c9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e44 PSN 0xba56b3 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff6d9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e56 PSN 0x1b713f OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff6e9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cc9 PSN 0x2902d4 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff6f9b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2cdf PSN 0x2fe00f OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff709b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e5b PSN 0xe3d884 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff719b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c79 PSN 0x479bb6 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff729b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2c90 PSN 0x964a9d OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff739b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e69 PSN 0x52d9db OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff749b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2e72 PSN 0xfc9255 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff759b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02
remote address: LID 0000 QPN 0x2d27 PSN 0xa9d320 OUT 0x7e RKey 0x2009e5f VAddr
0x007ffff769b000
GID: 253:00:00:02:00:16:00:01:00:01:00:02:00:00:00:02

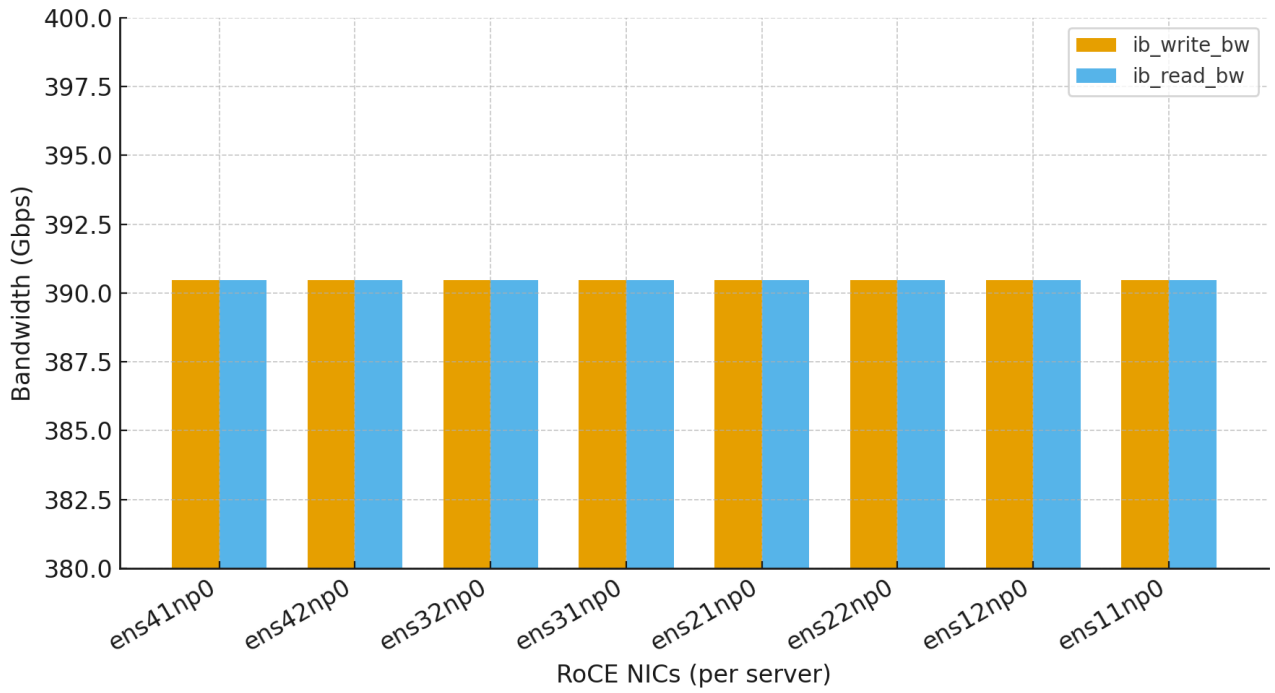
```

```

-----
#bytes      #iterations      BW peak[Gb/sec]      BW average[Gb/sec]      MsgRate[Mpps]
1048576      160000              390.49                390.48                  0.046549
-----

```

The following graphic shows the `ib_read_bw` and `ib_write_bw` throughput values across all eight NICs of an AMD MI300X server.



13 MLCommons benchmarks

| MLCommons benchmarking – Training | |
|--|---|
| Llama 2 70B – 8/16/24/32 accelerators | 1-node training – 28.8 minutes 2-node training – 15.4 minutes 3-node training – 10.6 minutes 4-node training – 10.3 minutes |
| MLCommons benchmarking – Inference | |
| Llama 2 70B – 8/16/24/32 accelerators – offline mode | 1-node inference - 26372.1 tokens/second 2-node inference - 53812.7 tokens/second 3-node inference - 78864.1 tokens/second 4-node inference – 101314 tokens/second |

13.1 Training

13.1.1 Llama 2 70B using local storage for dataset

Llama 2 70B can be benchmarked using MangoBoost for AMD MI300X GPUs for single-node and multi-node MLPerf results. First, the Llama 2 70B-LoRA training dataset, GovReport, must be downloaded. (GovReport is a dataset for long document summarization that consists of reports written by government research agencies. The dataset is already tokenized and packed so that each sequence has a length of 8192.)

Perform the following steps to download GovReport using a docker container provided by AMD.

1. Pull the docker image and then instantiate the container.

```
cse@slate1:~$ sudo docker pull rocm/amd-mlperf:llama2_70b_training_5.0

cse@slate1:~$ sudo docker run -it \
-e HTTP_PROXY=http://squid.ncse.io:3128/ \
-e http_proxy=http://squid.ncse.io:3128/ \
-e HTTPS_PROXY=http://squid.ncse.io:3128/ \
-e https_proxy=http://squid.ncse.io:3128/ \
-e NO_PROXY=localhost,127.0.0.0/8 \
-e no_proxy=localhost,127.0.0.0/8 \
--dns=128.251.10.125 \
--dns=128.251.10.145 \
-v /models/amd2025_april/mlperf_llama2:/data \
--ipc=host --uts=host --device /dev/dri --device /dev/kfd \
--security-opt=seccomp=unconfined \
rocm/amd-mlperf:llama2_70b_training_5.0
```

2. Confirm that you can connect to huggingface.co using CURL:

```
root@slate1:/workspace/code# curl -I https://huggingface.co
HTTP/1.1 200 Connection established

HTTP/2 200
content-type: text/html; charset=utf-8
content-length: 133992
date: Wed, 05 Nov 2025 10:00:51 GMT
x-powered-by: huggingface-moon
x-request-id: Root=1-690b2053-5cc9435469c418673d4e99a7
ratelimit: "pages";r=96;t=97
ratelimit-policy: "fixed window";"pages";q=100;w=300
cross-origin-opener-policy: same-origin
referrer-policy: strict-origin-when-cross-origin
x-frame-options: DENY
etag: W/"20b68-T8MbQji4htarKfcNf2V2owetz+Y"
x-cache: Hit from cloudfront
via: 1.1 c913b0cdc6eda18dcda37ba786f8a1b6.cloudfront.net (CloudFront)
x-amz-cf-pop: DEN53-P4
x-amz-cf-id: eecLXP8yLwNCgyZaxtG9GjnZi_rQkn6rrkfkJDy2v0rMpzaQELWRww==
age: 26
```

- Download and prepare the dataset by executing the following script in the docker container. This script downloads the required data into your locally mapped directory (the example here uses /models/amd2025_april/mlperf_llama2/data/).

```

root@slatel1:/workspace/code# bash ./scripts/prepare_data_and_model.sh
/opt/conda/envs/py_3.10/lib/python3.10/site-
packages/huggingface_hub/file_download.py:1194: UserWarning: `local_dir_use_symlinks`
parameter is deprecated and will be ignored. The process to download files to a local
folder has been updated and do not rely on symlinks anymore. You only need to pass a
destination folder as `local_dir`.
For more details, check out
https://huggingface.co/docs/huggingface_hub/main/en/guides/download#download-files-to-
local-folder.
  warnings.warn(
validation-00000-of-00001.parquet:
100%|+++++|
++++| 2.69M/2.69M [00:00<00:00, 12.2MB/s]
train-00000-of-00001.parquet:
100%|+++++|
++++| 109M/109M [00:02<00:00, 51.0MB/s]
Fetching 2 files:
100%|+++++|
++++| 2/2 [00:03<00:00, 1.53s/it]
Successfully downloaded and verified dataset with hash 682a5f40b790a56751bf8303554efc08
/opt/conda/envs/py_3.10/lib/python3.10/site-
packages/huggingface_hub/file_download.py:1194: UserWarning: `local_dir_use_symlinks`
parameter is deprecated and will be ignored. The process to download files to a local
folder has been updated and do not rely on symlinks anymore. You only need to pass a
destination folder as `local_dir`.
For more details, check out
https://huggingface.co/docs/huggingface_hub/main/en/guides/download#download-files-to-
local-folder.
  warnings.warn(
convert.py: 1.21kB [00:00, 7.24MB/s]
| 0/34 [00:00<?, ?it/s]
generation_config.json:
100%|+++++|
++++| 240/240 [00:00<00:00, 2.68MB/s]
config.json:
100%|+++++|
++++| 846/846 [00:00<00:00, 9.22MB/s]
Fetching 34 files:  3%|+++++
| 1/34 [00:00<00:17, 1.89it/s]
model-00005-of-00029.safetensors: 26%|+++++
| 1.23G/4.66G [01:27<04:49, 11.9MB/s]
model-00001-of-00029.safetensors: 23%|+++++
| 1.08G/4.72G [01:27<04:27, 13.6MB/s]

```

```

model-00003-of-00029.safetensors: 24%|+++++
| 1.18G/4.97G [01:27<04:16, 14.7MB/s]
model-00002-of-00029.safetensors: 25%|+++++
| 1.18G/4.66G [01:27<04:27, 13.0MB/s]
model-00004-of-00029.safetensors: 24%|+++++
| 1.18G/5.00G [01:27<04:56, 12.9MB/s]
model-00006-of-00029.safetensors: 26%|+++++
| 1.21G/4.66G [01:27<03:48, 15.1MB/s]
model-00007-of-00029.safetensors: 25%|+++++
| 1.17G/4.66G [01:26<04:11, 13.9MB/s]
model-00008-of-00029.safetensors: 23%|+++++
| 1.16G/4.97G [01:27<04:20, 14.6MB/s]

```

snip

```

cse@slate1:~$ ls -l /models/amd2025_april/mlperf_llama2/
total 8
drwxr-xr-x 3 root root 4096 Nov  5 05:01 data
drwx----- 3 root root 4096 Nov  5 03:25 model

```

4. Instantiate the docker container for running the training job. This includes the DRI (direct rendering infrastructure) devices from the host and giving access to the GPU compute and display resources to the container. In addition, this also exposes the Kernel Fusion Driver (KFD) which is used by AMD GPUs for ROCm.

```

sudo docker run --rm -it \
  --network host \
  --ipc host \
  --uts host \
  --cap-add=SYS_PTRACE \
  --security-opt seccomp=unconfined \
  --group-add video \
  --device /dev/dri:/dev/dri \
  --device /dev/kfd:/dev/kfd \
  --ulimit memlock=-1:-1 \
  --ulimit stack=67108864 \
  -v /models/amd2025_april/mlperf_llama2/data:/data \
  -v /models/amd2025_april/mlperf_llama2/model:/ckpt \
  -w /workspace/mlperf_training \
  llmboost/mb-llmboost-training:mlperf-5.0-prod

```

In this container, AMD provides scripts for single-node (8 GPUs in a single AMD MI300X server), 2-node (16 GPUs across two AMD MI300X servers) and 4-node (32 GPUs across four AMD MI300X servers) benchmarking, with scripts for common parameters included.

For a single-node MLPerf test using this model, the following parameter script is used:

```

root@slate1:/workspace/mlperf_training# cat config_MI300X_1x8x1.sh
#!/bin/bash
export BS=4
export GRAD_ACCUM_STEPS=2

export WARMUP=True

```

```

export DGXNGPU=8
export DGXSYSTEM=$(basename $(readlink -f ${BASH_SOURCE[0]}) | sed 's/^config_//' | sed 's/\.sh$//')

export DGXNNODES=1
export WALLTIME_MINUTES=50
export WALLTIME=$(( ${NEXP:-1} * WALLTIME_MINUTES ))

export LORA_A2A=1
export POSSIBLE_USER_WARNINGS=0
export CUDNN_FRONTEND_ATTN_DP_WORKSPACE_LIMIT=0

export MAX_STEPS=800
export TP=1
export PP=1
export CP=1
export SP=False
export VBOOST_VALUE=1
export MBS=1
export LR=0.00045
export MINIBS=1
export SKIP_EVALS=3
export VAL_CHECK_INTERVAL=384
export HYDRA_FULL_ERROR=1
export CUDA_DEVICE_MAX_CONNECTIONS=1

export FP8_DPA=0
export FP8=True
export FP8_AMAX_ALGO=most_recent
export FP8_REDUCE_AMAX=False
export FP8_AMAX_HISTORY=4
export FP8_ACTIVATION=True

export ACG=full && export ACM=block && export ACL=21
export FUSED_SOFTMAX=0
export RMSNORM_CAST=0

export PT_TENSOR_VALIDATION=0
export PROFILE_RPD=0

export USE_HIPBLASLT=1
export TORCH_BLAS_PREFER_HIPBLASLT=1
export NVTE_USE_RMSNORM_TRITON=1

export MLPERF_SUBMISSION_ORG="MangoBoost"
export MLPERF_SUBMISSION_PLATFORM="MI300X"

```

The single-node training is then started, and the throughput (tokens per second) and job completion time is monitored.

```

root@slate1:/workspace/mlperf_training# llmboost mlperf --config_sh config_MI300X_1x8x1.sh
2>&1 | tee "log_single_node.txt"
STARTING TIMING RUN AT 2025-11-11 02:59:34 AM
W1111 02:59:43.200000 7234 site-packages/torch/distributed/run.py:792]

```

```

W1111 02:59:43.200000 7234 site-packages/torch/distributed/run.py:792]
*****
W1111 02:59:43.200000 7234 site-packages/torch/distributed/run.py:792] Setting
OMP_NUM_THREADS environment variable for each process to be 1 in default, to avoid your
system being overloaded, please further tune the variable for optimal performance in your
application as needed.
W1111 02:59:43.200000 7234 site-packages/torch/distributed/run.py:792]
*****
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
GPU available: True (cuda), used: True
TPU available: False, using: 0 TPU cores
HPU available: False, using: 0 HPUs
`Trainer(limit_train_batches=1.0)` was configured so 100% of the batches per epoch will be
used..
`Trainer(limit_val_batches=1.0)` was configured so 100% of the batches will be used..
25-11-11 03:00:14 - PID:7300 - rank:(0, 0, 0, 0) - microbatch.py:39 - INFO - setting
number of micro-batches to constant 1
You are using a CUDA device ('AMD Instinct MI300X') that has Tensor Cores. To properly
utilize them, you should set `torch.set_float32_matmul_precision('medium' | 'high')` which
will trade-off precision for performance. For more details, read
https://pytorch.org/docs/stable/generated/torch.set\_float32\_matmul\_precision.html#torch.set\_float32\_matmul\_precision
Initializing distributed: GLOBAL_RANK: 0, MEMBER: 1/8
-----
distributed_backend=nccl
All distributed processes registered. Starting with 8 processes
-----
Initializing distributed: GLOBAL_RANK: 5, MEMBER: 6/8
Initializing distributed: GLOBAL_RANK: 3, MEMBER: 4/8
Initializing distributed: GLOBAL_RANK: 2, MEMBER: 3/8
Initializing distributed: GLOBAL_RANK: 6, MEMBER: 7/8
Initializing distributed: GLOBAL_RANK: 7, MEMBER: 8/8
Initializing distributed: GLOBAL_RANK: 4, MEMBER: 5/8
Initializing distributed: GLOBAL_RANK: 1, MEMBER: 2/8
Loading distributed checkpoint with TensorStoreLoadShardedStrategy
make: Entering directory
'/workspace/deps/nemo/nemo/collections/nlp/data/language_modeling/megatron'
make: Nothing to be done for 'default'.
make: Leaving directory
'/workspace/deps/nemo/nemo/collections/nlp/data/language_modeling/megatron'
> building indices for blendable datasets ...
> sample ratios:
  dataset 0, input: 1, achieved: 1
LOCAL_RANK: 5 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]

```

```

LOCAL_RANK: 3 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]
LOCAL_RANK: 6 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]
LOCAL_RANK: 4 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]
LOCAL_RANK: 1 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]
LOCAL_RANK: 2 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]
LOCAL_RANK: 7 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]
LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0,1,2,3,4,5,6,7]

```

| Name | Type | Params | Mode |
|--|-----------------------|---------------|----------------|
| 0 | model | Float16Module | 69.0 B train |
| 44.6 M | Trainable params | | |
| 69.0 B | Non-trainable params | | |
| 69.0 B | Total params | | |
| 276,084.851 Total estimated model params size (MB) | | | |
| 2490 | Modules in train mode | | |
| 0 | Modules in eval mode | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249398, "event_type": "POINT_IN_TIME", "key": "cache_clear", "value": true, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 233}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "INTERVAL_START", "key": "init_start", "value": null, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 234}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "submission_benchmark", "value": "llama2_70b_lora", "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 235}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "submission_org", "value": "MangoBoost", "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 235}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "submission_division", "value": "closed", "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 235}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "submission_status", "value": "onprem", "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 235}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "submission_platform", "value": "1xMI300X", "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 235}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "seed", "value": 1, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 236}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249399, "event_type": "POINT_IN_TIME", "key": "global_batch_size", "value": 8, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 242}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249926, "event_type": "POINT_IN_TIME", "key": "train_samples", "value": 3901, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 247}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249942, "event_type": "POINT_IN_TIME", "key": "eval_samples", "value": 173, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 251}} | | | |
| ::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key": "opt_learning_rate_warmup_factor", "value": 0.0, "metadata": {"file": "/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 255}} | | | |

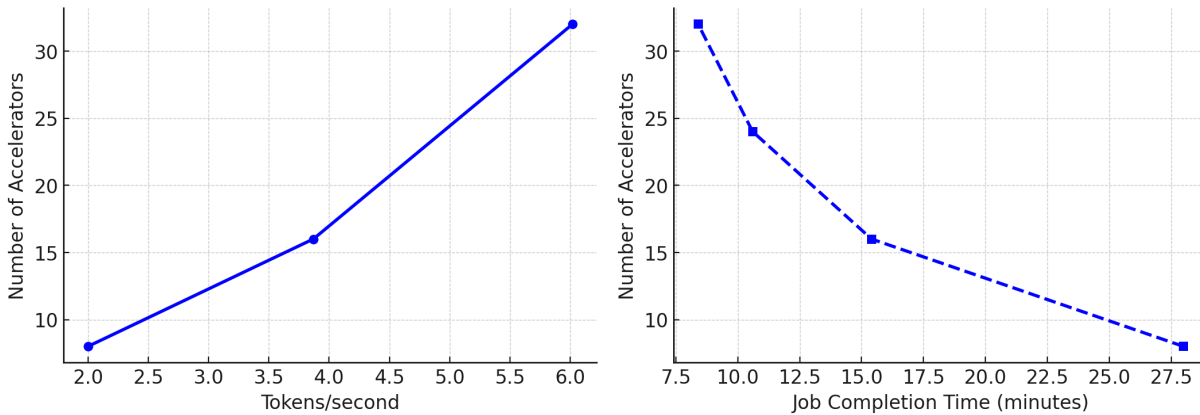
```
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"opt_adamw_weight_decay", "value": 0.0001, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 259}}
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"opt_gradient_clip_norm", "value": 0.3, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 263}}
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"gradient_accumulation_steps", "value": 1, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 268}}
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"opt_learning_rate_training_steps", "value": 800, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 269}}
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"opt_base_learning_rate", "value": 0.00045, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 270}}
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"lora_rank", "value": 16, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 271}}
:::MLLOG {"namespace": "", "time_ms": 1762830249943, "event_type": "POINT_IN_TIME", "key":
"lora_alpha", "value": 32, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 272}}
:::MLLOG {"namespace": "", "time_ms": 1762830356561, "event_type": "INTERVAL_END", "key":
"init_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 179}}
:::MLLOG {"namespace": "", "time_ms": 1762830356561, "event_type": "INTERVAL_START",
"key": "run_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 179}}
:::MLLOG {"namespace": "", "time_ms": 1762830356561, "event_type": "INTERVAL_START",
"key": "block_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 180,
"samples_count": 0}}
:::MLLOG {"namespace": "", "time_ms": 1762831151348, "event_type": "POINT_IN_TIME", "key":
"tracked_stats", "value": {"throughput": 2.0109121617680814}, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 197, "step":
1536}}
:::MLLOG {"namespace": "", "time_ms": 1762831151348, "event_type": "INTERVAL_END", "key":
"block_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 140,
"samples_count": 1536, "curr_global_step": 192, "self.gbs": 8}}
:::MLLOG {"namespace": "", "time_ms": 1762831151348, "event_type": "INTERVAL_START",
"key": "eval_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 149,
"samples_count": 1536}}
:::MLLOG {"namespace": "", "time_ms": 1762831185162, "event_type": "POINT_IN_TIME", "key":
"eval_accuracy", "value": 0.9366033673286438, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 217,
"samples_count": 1536}}
:::MLLOG {"namespace": "", "time_ms": 1762831185162, "event_type": "INTERVAL_END", "key":
"eval_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 222,
"samples_count": 1536}}
:::MLLOG {"namespace": "", "time_ms": 1762831185162, "event_type": "INTERVAL_START",
"key": "block_start", "value": null, "metadata": {"file":
```

```
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 163,
"samples_count": 1536}}
::MLLOG {"namespace": "", "time_ms": 1762831376166, "event_type": "POINT_IN_TIME", "key":
"tracked_stats", "value": {"throughput": 2.010920663666792}, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 197, "step":
1920}}
::MLLOG {"namespace": "", "time_ms": 1762831376166, "event_type": "INTERVAL_END", "key":
"block_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 140,
"samples_count": 1920, "curr_global_step": 240, "self.gbs": 8}}
::MLLOG {"namespace": "", "time_ms": 1762831376166, "event_type": "INTERVAL_START",
"key": "eval_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 149,
"samples_count": 1920}}
::MLLOG {"namespace": "", "time_ms": 1762831409898, "event_type": "POINT_IN_TIME", "key":
"eval_accuracy", "value": 0.9349899888038635, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 217,
"samples_count": 1920}}
::MLLOG {"namespace": "", "time_ms": 1762831409898, "event_type": "INTERVAL_END", "key":
"eval_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 222,
"samples_count": 1920}}
::MLLOG {"namespace": "", "time_ms": 1762831409898, "event_type": "INTERVAL_START",
"key": "block_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 163,
"samples_count": 1920}}
::MLLOG {"namespace": "", "time_ms": 1762831601039, "event_type": "POINT_IN_TIME", "key":
"tracked_stats", "value": {"throughput": 2.009449575877306}, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 197, "step":
2304}}
::MLLOG {"namespace": "", "time_ms": 1762831601039, "event_type": "INTERVAL_END", "key":
"block_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 140,
"samples_count": 2304, "curr_global_step": 288, "self.gbs": 8}}
::MLLOG {"namespace": "", "time_ms": 1762831601039, "event_type": "INTERVAL_START",
"key": "eval_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 149,
"samples_count": 2304}}
::MLLOG {"namespace": "", "time_ms": 1762831634689, "event_type": "POINT_IN_TIME", "key":
"eval_accuracy", "value": 0.9322686791419983, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 217,
"samples_count": 2304}}
::MLLOG {"namespace": "", "time_ms": 1762831634689, "event_type": "INTERVAL_END", "key":
"eval_stop", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 222,
"samples_count": 2304}}
::MLLOG {"namespace": "", "time_ms": 1762831634689, "event_type": "INTERVAL_START",
"key": "block_start", "value": null, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 163,
"samples_count": 2304}}
::MLLOG {"namespace": "", "time_ms": 1762831825894, "event_type": "POINT_IN_TIME", "key":
"tracked_stats", "value": {"throughput": 2.0087606665551485}, "metadata": {"file":
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 197, "step":
2688}}
```

```
:::MLLOG {"namespace": "", "time_ms": 1762831825894, "event_type": "INTERVAL_END", "key":  
"block_stop", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 140,  
"samples_count": 2688, "curr_global_step": 336, "self.gbs": 8}}  
:::MLLOG {"namespace": "", "time_ms": 1762831825894, "event_type": "INTERVAL_START",  
"key": "eval_start", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 149,  
"samples_count": 2688}}  
:::MLLOG {"namespace": "", "time_ms": 1762831859560, "event_type": "POINT_IN_TIME", "key":  
"eval_accuracy", "value": 0.9295409321784973, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 217,  
"samples_count": 2688}}  
:::MLLOG {"namespace": "", "time_ms": 1762831859560, "event_type": "INTERVAL_END", "key":  
"eval_stop", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 222,  
"samples_count": 2688}}  
:::MLLOG {"namespace": "", "time_ms": 1762831859560, "event_type": "INTERVAL_START",  
"key": "block_start", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 163,  
"samples_count": 2688}}  
:::MLLOG {"namespace": "", "time_ms": 1762832050895, "event_type": "POINT_IN_TIME", "key":  
"tracked_stats", "value": {"throughput": 2.007418610699823}, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 197, "step":  
3072}}  
:::MLLOG {"namespace": "", "time_ms": 1762832050895, "event_type": "INTERVAL_END", "key":  
"block_stop", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 140,  
"samples_count": 3072, "curr_global_step": 384, "self.gbs": 8}}  
:::MLLOG {"namespace": "", "time_ms": 1762832050895, "event_type": "INTERVAL_START",  
"key": "eval_start", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 149,  
"samples_count": 3072}}  
:::MLLOG {"namespace": "", "time_ms": 1762832084594, "event_type": "POINT_IN_TIME", "key":  
"eval_accuracy", "value": 0.9208933115005493, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 217,  
"samples_count": 3072}}  
:::MLLOG {"namespace": "", "time_ms": 1762832084594, "event_type": "INTERVAL_END", "key":  
"eval_stop", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 222,  
"samples_count": 3072}}  
:::MLLOG {"namespace": "", "time_ms": 1762832084602, "event_type": "INTERVAL_END", "key":  
"run_stop", "value": null, "metadata": {"file":  
"/workspace/mlperf_training/src/callbacks/custom_callbacks.py", "lineno": 185,  
"samples_count": 3072, "status": "success", "duration": "1728.0411961078644 sec ->  
28.80068660179774 minutes"}}  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed  
FlashAttention Installed
```

```
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
FlashAttention Installed
ENDING TIMING RUN AT 2025-11-11 03:36:26 AM
RESULT,LLM_FINETUNING,,2212,AMD,2025-11-11 02:59:34 AM
Config shell script: config_MI300X_1x8x1.sh
Starting single-node MLPerf training benchmark...
Running in single-node mode...
MLPerf training benchmark completed.
```

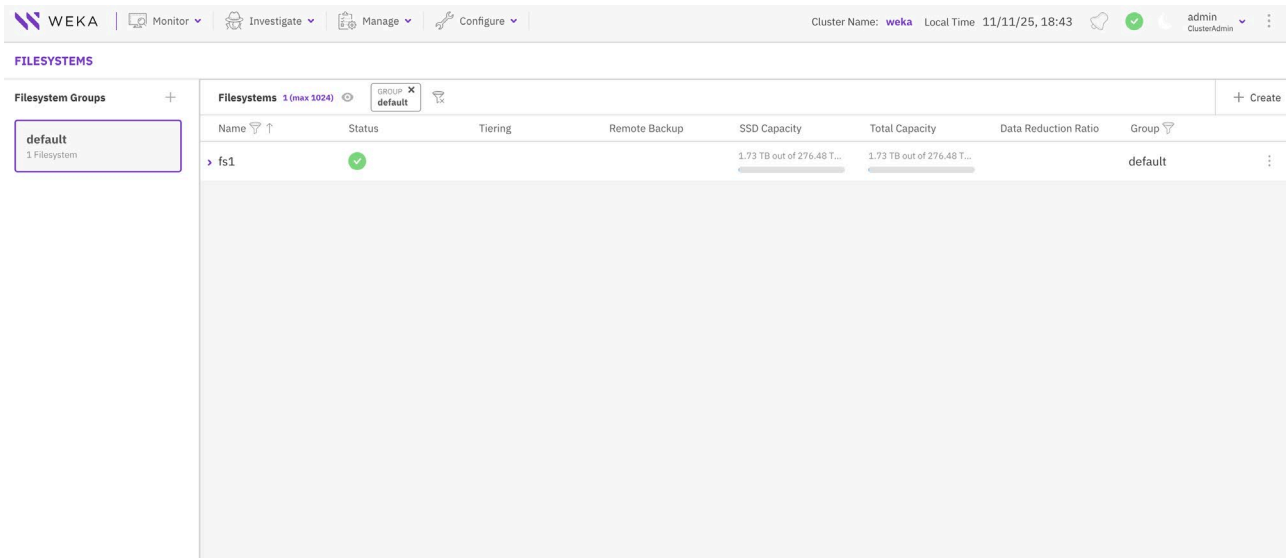
In the same way, the training model is run for 2-node, 3-node and 4-node clusters, with the throughput and job completion time for all tests plotted in the graphs below.



13.1.2 Llama 2 70B using remote storage (WEKA-mounted file system) for dataset

With an 8-node WEKA storage cluster configured in the converged frontend/storage fabric, instead of using local storage, the GPUs can mount a file system in the WEKA cluster.

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 accessible to all GPUs.

```
cse@slate1:~/benchmarking/mpi-tests$ 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 "slate1":"client" allocated 7030 out of 7030 required hugepages after 1
retries
client: Container "slate1":"client" allocated 6327 out of 6327 required hugepages after 1
retries
client: Allocated 26714MB hugepages memory from 2 NUMA nodes for "slate1":"client"
client: Bandwidth of "slate1":"client" set to unlimited
client: WekaFS driver attached by "NodeId<65534>" on "slate1":"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,writocache,relatime,rw,relatime_threshold=0,token=<redacted>,192.168
.1.190/fs1 /mnt/weka`
Cgroups v1 not found, running without cgroups
Mount completed successfully
```

```
cse@slate1:~/benchmarking/mpi-tests$ sudo mkdir -p
/mnt/weka/models/amd2025_april/mlperf_llama2
```

- View the client list on the WEKA UI to confirm that all clients are up and the expected number of cores have been allocated per-client.

The screenshot shows the WEKA UI interface. The top navigation bar includes 'WEKA', 'Monitor', 'Investigate', 'Manage', and 'Configure'. The main content area is titled 'SERVERS' and shows a table of servers. The table has the following columns: UID, Hostname, IP Address, Version, Status, Cores, Drives, Load, Memory, Architecture, Uptime, and Server Removal Countdown. There are four rows of server data, all with a status of 'UP' and 19/19 cores allocated.

| UID | Hostname | IP Address | Version | Status | Cores | Drives | Load | Memory | Architecture | Uptime | Server Removal Countdown |
|-----------|----------|---------------|----------|--------|-------|--------|------|----------|--------------|--------|--------------------------|
| 000000... | slate2 | 192.168.1.173 | 5.0.3.19 | UP | 19/19 | 0 | 1% | 28.01 GB | x86_64 | 7h | 1h |
| 000000... | slate3 | 192.168.1.174 | 5.0.3.19 | UP | 19/19 | 0 | 2% | 28.01 GB | x86_64 | 14d | 1h |
| 000000... | slate1 | 192.168.1.172 | 5.0.3.19 | UP | 19/19 | 0 | 1% | 28.01 GB | x86_64 | 22h | 1h |
| 000000... | slate4 | 192.168.1.175 | 5.0.3.19 | UP | 19/19 | 0 | 1% | 28.01 GB | x86_64 | 7h | 1h |

- Create the container by passing the mounted file system path instead of a local directory. In the same way, for running the training, pass the same mounted file system instead of a local directory. The training is then triggered in the same way as shown earlier for local storage, monitoring the throughput and job completion time.

```
// to download and process dataset

sudo docker run -it \
  -e HTTP_PROXY=http://squid.ncse.io:3128/ \
  -e http_proxy=http://squid.ncse.io:3128/ \
  -e HTTPS_PROXY=http://squid.ncse.io:3128/ \
  -e https_proxy=http://squid.ncse.io:3128/ \
  -e NO_PROXY=localhost,127.0.0.0/8 \
  -e no_proxy=localhost,127.0.0.0/8 \
  --dns=128.251.10.125 \
  --dns=128.251.10.145 \
  -v /mnt/weka/models/amd2025_april/mlperf_llama2:/data:rw \
  --ipc=host --uts=host --device /dev/dri --device /dev/kfd \
  --security-opt=seccomp=unconfined \
  rocm/amd-mlperf:llama2_70b_training_5.0

// to run training models

sudo docker run --rm -it \
  --network host \
  --ipc host \
  --uts host \
  --cap-add=SYS_PTRACE \
```

```

--security-opt seccomp=unconfined \
--group-add video \
--device /dev/dri:/dev/dri \
--device /dev/kfd:/dev/kfd \
--ulimit memlock=-1:-1 \
--ulimit stack=67108864 \
-v /mnt/weka/models/amd2025_april/mlperf_llama2/data:/data \
-v /mnt/weka/models/amd2025_april/mlperf_llama2/model:/ckpt \
-w /workspace/mlperf_training \
llmboost/mb-llmboost-training:mlperf-5.0-prod

```

13.2 Inference

Inference testing is done using Llama, following the steps documented here.

1. Pulling the required docker image on the AMD MI300X servers (as needed). This example assumes that the WEKA mounted file system is used to store the dataset.

```

cse@slate1:~$ sudo docker pull llmboost/mlperf-model-data:latest
latest: Pulling from llmboost/mlperf-model-data
9824c27679d3: Pull complete
e48fefb90869: Pull complete
7c0a09ba434e: Pull complete
7eb858211fa9: Pull complete
787c03c77281: Downloading [=====> ]
43.77GB/56.8GB
787c03c77281: Pull complete
837939fc8896: Pull complete
Digest: sha256:4ca732485f6efe0706a4d9719d2a5305f907de2b23852119545144ef980a3f23
Status: Downloaded newer image for llmboost/mlperf-model-data:latest
docker.io/llmboost/mlperf-model-data:latest

```

2. Create a container using this docker image and copy the required dataset to the WEKA mounted file system.

```

cse@slate1:~$ sudo docker create llmboost/mlperf-model-data --name mlperf-models
6a3ef474ac94f23c238f99537295da163f2788308b42cab77b3ede13c9a4bd3b

```

```

cse@slate1:~$ sudo mkdir /mnt/weka/mlperf-models

cse@slate1:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                COMMAND                                CREATED
STATUS        PORTS          NAMES
6a3ef474ac94   llmboost/mlperf-model-data          "--name mlperf-models"              About a minute ago
Created
              beautiful_bouman

cse@slate1:~$ sudo docker cp 6a3ef474ac94:/models /mnt/weka/mlperf-models/
Successfully copied 74.6GB to /mnt/weka/mlperf-models/

```

3. Start server on all nodes (participating in the inference testing) by creating the docker container and using server-side python script. After the server is started, SSH into one of the nodes and start the client.

```
// create container

sudo docker run -it --rm \
  --network host \
  --group-add video \
  --ipc host \
  --cap-add=SYS_PTRACE \
  --security-opt seccomp=unconfined \
  --device=/dev/dri:/dev/dri \
  --device=/dev/kfd:/dev/kfd \
  -v /mnt/weka/mlperf-models/models:/models \
  mangollm/mb-llmboost-inference:prod-rocm-mlperf-5_1-mi300-mi325-drop1

// run server-side python script

root@slate1:/workspace# cd /workspace/apps/mlperf
root@slate1:/workspace/apps/mlperf# python3 server.py --test_mode Server --model_path
"/models/amd2025_model/model/llama2-70b-chat-hf/fp8_quantized" --accelerator_name mi300x
INFO 11-19 07:06:24 [__init__.py:244] Automatically detected platform rocm.
Initializing LLMBoost...
Preparing model with 2048 context length...
Deploying LLMBoost (this may take a few minutes)
.....|
I1119 07:09:10.320742 140737350291456 mlperf_server_func.py:285] Starting server with
model llama2-70b
I1119 07:09:10.325922 140737350291456 mlperf_server_func.py:286] Available routes are:
I1119 07:09:10.331039 140737350291456 mlperf_server_func.py:291] /openapi.json [GET, HEAD]
I1119 07:09:10.331068 140737350291456 mlperf_server_func.py:291] /docs [GET, HEAD]
I1119 07:09:10.331088 140737350291456 mlperf_server_func.py:291] /docs/oauth2-redirect
[GET, HEAD]
I1119 07:09:10.336178 140737350291456 mlperf_server_func.py:291] /redoc [GET, HEAD]
I1119 07:09:10.341273 140737350291456 mlperf_server_func.py:291] /predict [POST]
I1119 07:09:10.346360 140737350291456 mlperf_server_func.py:291] /getname [POST]
INFO: Started server process [250]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

*snip*

// SSH into one of the nodes and start client script (this example assumes 2-node testing)

cd /workspace/apps/mlperf
python3 client.py \
  --test_mode Offline \
  --user_conf conf/user_llama2-70b_16x_mi300x.conf \
  --sut_server_addr http://slate1.ncse.io:8000,http://slate2.ncse.io:8000
```

13.2.1 Results for single-node inference testing

```
=====
MLPerf Results Summary
=====
SUT name : Network SUT
Scenario : Offline
Mode      : PerformanceOnly
Samples per second: 87.4718
Tokens per second: 26372.1
Result is : VALID
  Min duration satisfied : Yes
  Min queries satisfied  : Yes
  Early stopping satisfied: Yes

=====
Additional Stats
=====
Min latency (ns)           : 56732921325
Max latency (ns)           : 842877536537
Mean latency (ns)          : 464019723369
50.00 percentile latency (ns) : 467274530573
90.00 percentile latency (ns) : 768881031512
95.00 percentile latency (ns) : 788687226379
97.00 percentile latency (ns) : 798244870010
99.00 percentile latency (ns) : 811748143360
99.90 percentile latency (ns) : 829053090419

=====
Test Parameters Used
=====
samples_per_query : 60720
target_qps        : 92
ttft_latency (ns): 2000000000
tpot_latency (ns): 2000000000
max_async_queries : 1
min_duration (ms): 600000
max_duration (ms): 0
min_query_count   : 1
max_query_count   : 0
qsl_rng_seed     : 1780908523862526354
sample_index_rng_seed : 14771362308971278857
schedule_rng_seed : 18209322760996052031
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 : 24576
WARNING: sample_concatenate_permutation was set to true.
```

Generated samples per query might be different as the one in the setting.
Check the generated_samples_per_query line in the detailed log for the real
samples_per_query value

No warnings encountered during test.

No errors encountered during test.

I1205 07:27:17.605689 140737350291456 client.py:123] Run Completed!

I1205 07:27:17.605761 140737350291456 client.py:124] Destroying QSL...

13.2.2 Results for 2-node inference testing

```
=====
MLPerf Results Summary
=====
SUT name : Multi-Node SUT: Network SUT, Network SUT
Scenario : Offline
Mode      : PerformanceOnly
Samples per second: 178.419
Tokens per second: 53812.7
Result is : VALID
  Min duration satisfied : Yes
  Min queries satisfied : Yes
  Early stopping satisfied: Yes
```

Additional Stats

```
=====
Min latency (ns)           : 58632360019
Max latency (ns)           : 4132297949894
Mean latency (ns)         : 2070520957370
50.00 percentile latency (ns) : 2069909192320
90.00 percentile latency (ns) : 3675001375792
95.00 percentile latency (ns) : 3872946269352
97.00 percentile latency (ns) : 3943035212384
99.00 percentile latency (ns) : 4009465175814
99.90 percentile latency (ns) : 4084847907273
```

Test Parameters Used

```
=====
samples_per_query : 728640
target_qps        : 184
ttft_latency (ns): 2000000000
tpot_latency (ns): 2000000000
max_async_queries : 1
min_duration (ms): 3600000
max_duration (ms): 0
min_query_count  : 1
max_query_count  : 0
qsl_rng_seed     : 1780908523862526354
sample_index_rng_seed : 14771362308971278857
```

```

schedule_rng_seed : 18209322760996052031
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 : 24576
WARNING: sample_concatenate_permutation was set to true.
Generated samples per query might be different as the one in the setting.
Check the generated_samples_per_query line in the detailed log for the real
samples_per_query value

No warnings encountered during test.

No errors encountered during test.
I1120 16:39:38.727713 140737350291456 client.py:123] Run Completed!
I1120 16:39:38.727796 140737350291456 client.py:124] Destroying QSL...

```

13.2.3 Results for 3-node inference testing

```

=====
MLPerf Results Summary
=====
SUT name : Multi-Node SUT: Network SUT, Network SUT, Network SUT
Scenario : Offline
Mode      : PerformanceOnly
Samples per second: 261.468
Tokens per second: 78864.1
Result is : VALID
  Min duration satisfied : Yes
  Min queries satisfied : Yes
  Early stopping satisfied: Yes

=====
Additional Stats
=====
Min latency (ns)           : 56667270871
Max latency (ns)           : 1315895563142
Mean latency (ns)          : 693376212678
50.00 percentile latency (ns) : 695543837314
90.00 percentile latency (ns) : 1194851057538
95.00 percentile latency (ns) : 1232212066496
97.00 percentile latency (ns) : 1246556949262
99.00 percentile latency (ns) : 1269681708485
99.90 percentile latency (ns) : 1294195176328

=====
Test Parameters Used
=====
samples_per_query : 332640

```

```

target_qps : 504
ttft_latency (ns): 2000000000
tpot_latency (ns): 2000000000
max_async_queries : 1
min_duration (ms): 600000
max_duration (ms): 0
min_query_count : 1
max_query_count : 0
qsl_rng_seed : 1780908523862526354
sample_index_rng_seed : 14771362308971278857
schedule_rng_seed : 18209322760996052031
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 : 24576
WARNING: sample_concatenate_permutation was set to true.
Generated samples per query might be different as the one in the setting.
Check the generated_samples_per_query line in the detailed log for the real
samples_per_query value

No warnings encountered during test.

No errors encountered during test.
I1205 04:43:05.529103 138827985059840 client.py:123] Run Completed!
I1205 04:43:05.529183 138827985059840 client.py:124] Destroying QSL...

```

13.2.4 Results for 4-node inference testing

```

=====
MLPerf Results Summary
=====
SUT name : Multi-Node SUT: Network SUT, Network SUT, Network SUT, Network SUT
Scenario : Offline
Mode      : PerformanceOnly
Samples per second: 222.944
Tokens per second: 67244.1
Result is : INVALID
  Min duration satisfied : NO
  Min queries satisfied : Yes
  Early stopping satisfied: Yes
Recommendations:
  * Increase expected QPS so the loadgen pre-generates a larger (coalesced) query.

=====
Additional Stats
=====
Min latency (ns)           : 53742850757
Max latency (ns)           : 3307021321652
Mean latency (ns)          : 1483962578642

```

```
50.00 percentile latency (ns) : 1363250214234
90.00 percentile latency (ns) : 2933528927045
95.00 percentile latency (ns) : 3138763470998
97.00 percentile latency (ns) : 3217499155756
99.00 percentile latency (ns) : 3269513400126
99.90 percentile latency (ns) : 3306923248782
```

```
=====
Test Parameters Used
=====
```

```
samples_per_query : 728640
target_qps : 184
ttft_latency (ns): 2000000000
tpot_latency (ns): 2000000000
max_async_queries : 1
min_duration (ms): 3600000
max_duration (ms): 0
min_query_count : 1
max_query_count : 0
qsl_rng_seed : 1780908523862526354
sample_index_rng_seed : 14771362308971278857
schedule_rng_seed : 18209322760996052031
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 : 24576
WARNING: sample_concatenate_permutation was set to true.
Generated samples per query might be different as the one in the setting.
Check the generated_samples_per_query line in the detailed log for the real
samples_per_query value
```

```
No warnings encountered during test.
```

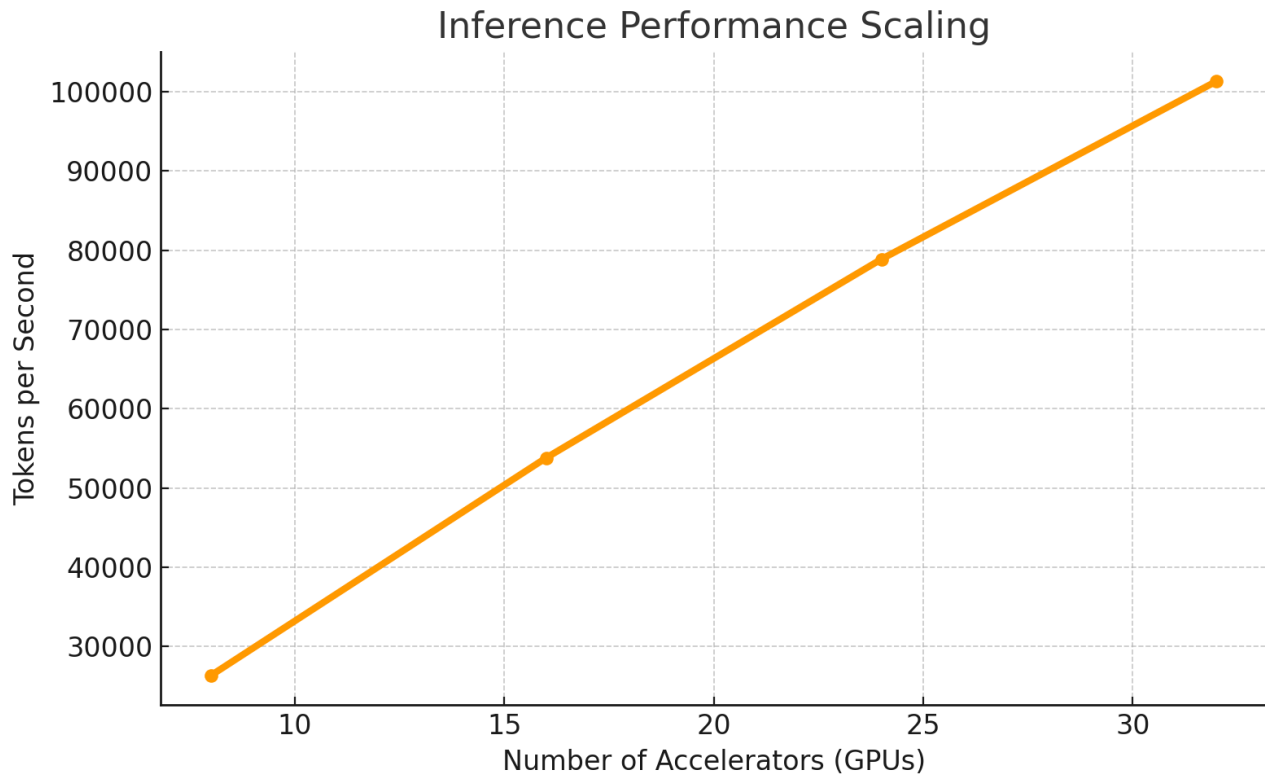
```
No errors encountered during test.
```

```
I1121 06:23:02.571674 140650969542656 client.py:123] Run Completed!
```

```
I1121 06:23:02.571752 140650969542656 client.py:124] Destroying QSL...
```

13.2.5 Inference performance scaling

The following graphic demonstrates how the number of tokens per second increases as the number of accelerators increases.

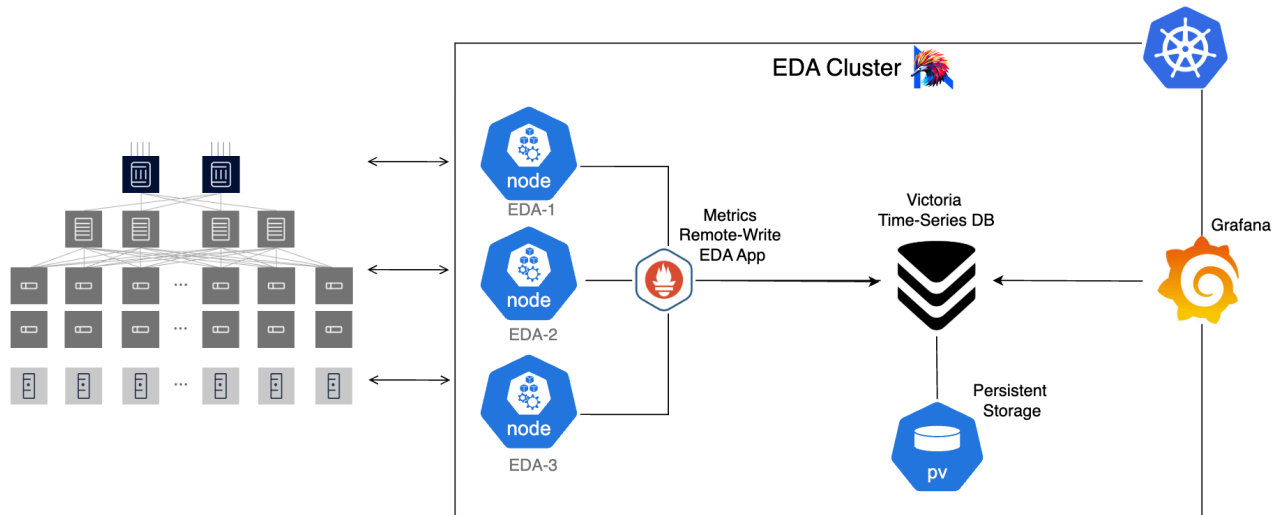


14 Telemetry

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 through 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, these 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, and 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 will 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.

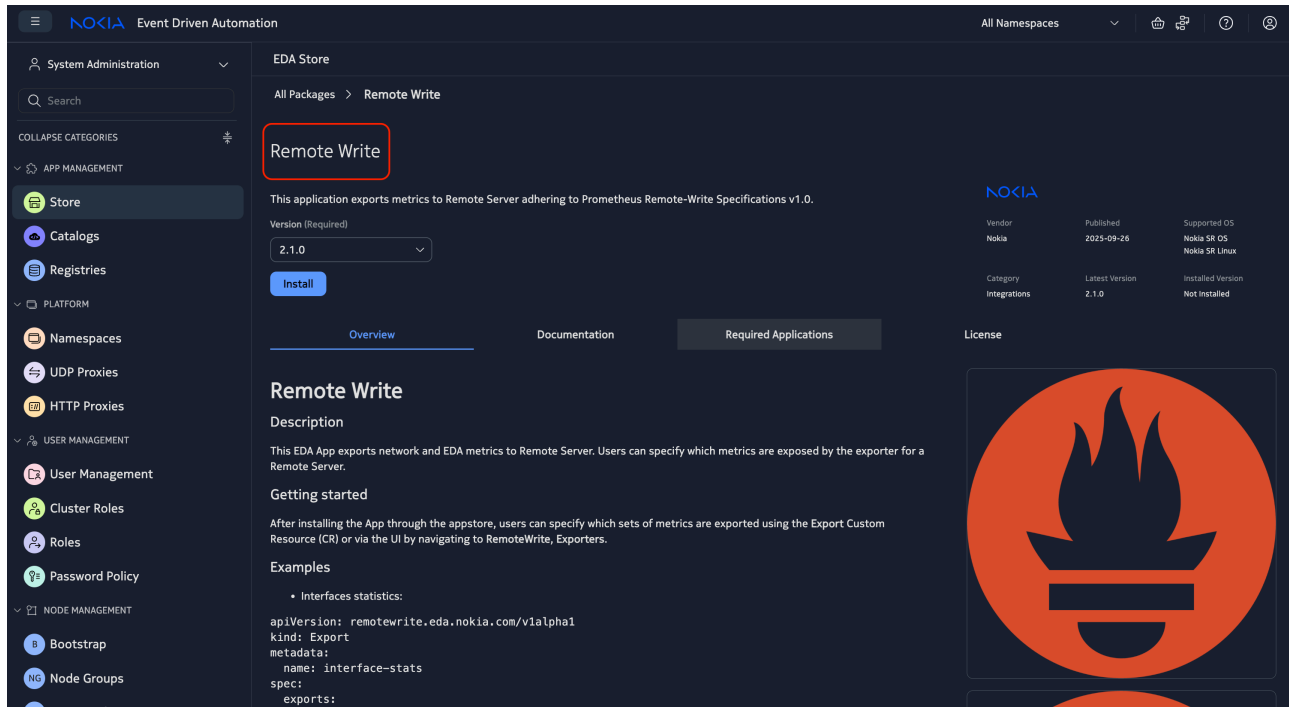
14.3.1 Installing the Remote Write app

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
```

- 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 **kubect1** 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.

The screenshot shows the Nokia Event Driven Automation (EDA) interface. The left sidebar lists various configuration categories, with 'Cluster Destinations' under 'REMOEWRITE' highlighted. The main panel shows the configuration for 'dest1'. The 'Status' section is highlighted, showing the 'Reachable' toggle is turned on. A code editor on the right displays the JSON configuration for the destination, with the 'status' field highlighted in green.

```
1 apiVersion: remotewrite.eda.nokia.com/
2 kind: ClusterDestination
3 metadata:
4   name: dest1
5 spec:
6   authentication: {}
7   authorization:
8     type: Bearer
9   metadata:
10    interval: 0s
11    maxEntriesPerWrite: 500
12    url: http://victoria-metrics.eda-tel
13    writeOptions:
14      bufferSize: 100
15      flushInterval: 1m0s
16      maxRetries: 3
17      maxTimeSeriesPerWrite: 500
18      timeout: 10s
19 status:
20   lastChecked: '2025-11-24T02:09:58Z'
21   reachable: true
22
```

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: First one is the destinations which defines the remote-write target, and 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.

```

1 apiVersion: remotewrite.eda.nokia.com/
2 kind: ClusterExport
3 metadata:
4   name: interface-traffic-rate
5 spec:
6   destinations:
7     - dest1
8   exports:
9     - interval: 10s
10      labels: {}
11      metricName:
12        regex: namespace_(.+)
```

14.4 Set up external time-series database and dashboards

The Helm charts install all the necessary tools and configurations with dashboards. Here in this lab, we are using VictoriaMetrics, which is compatible with v1.0 Prometheus and 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 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
```

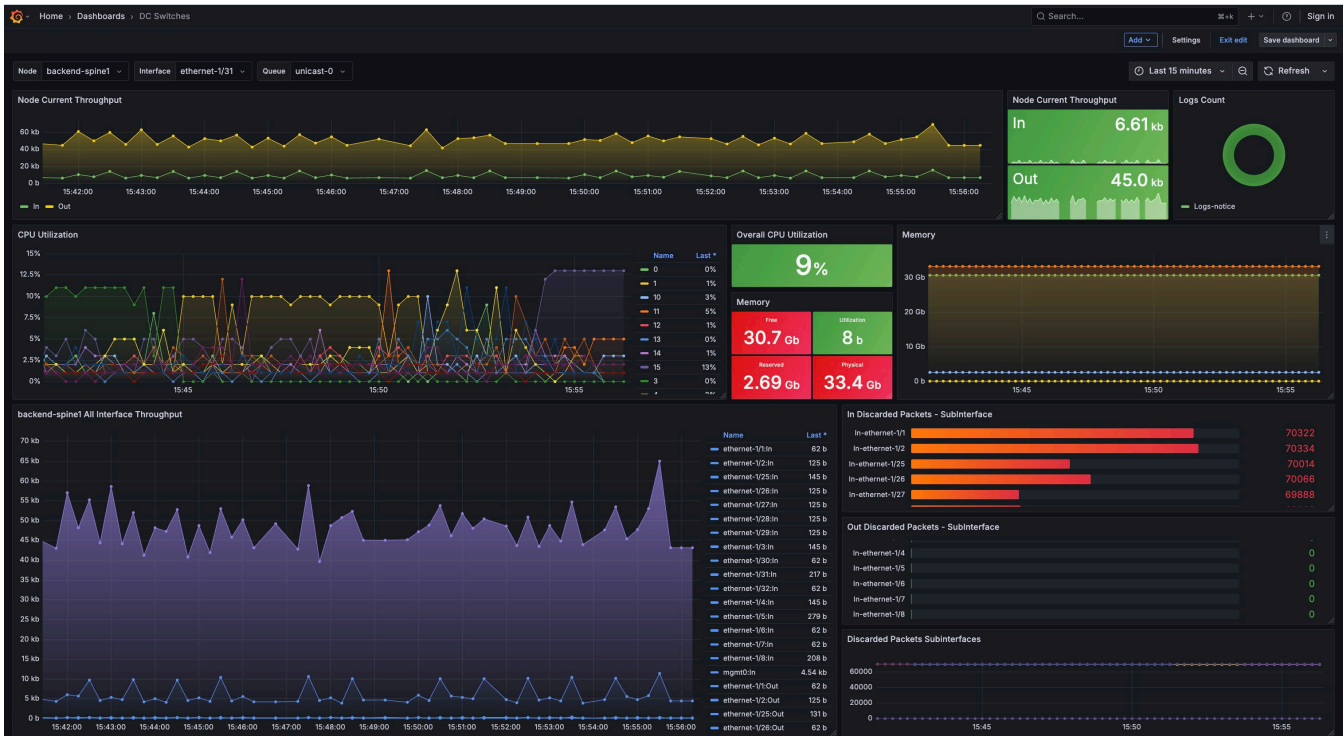
The Grafana and Victoria Metrics UI can be accessed by port-forwarding it.

```
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
```

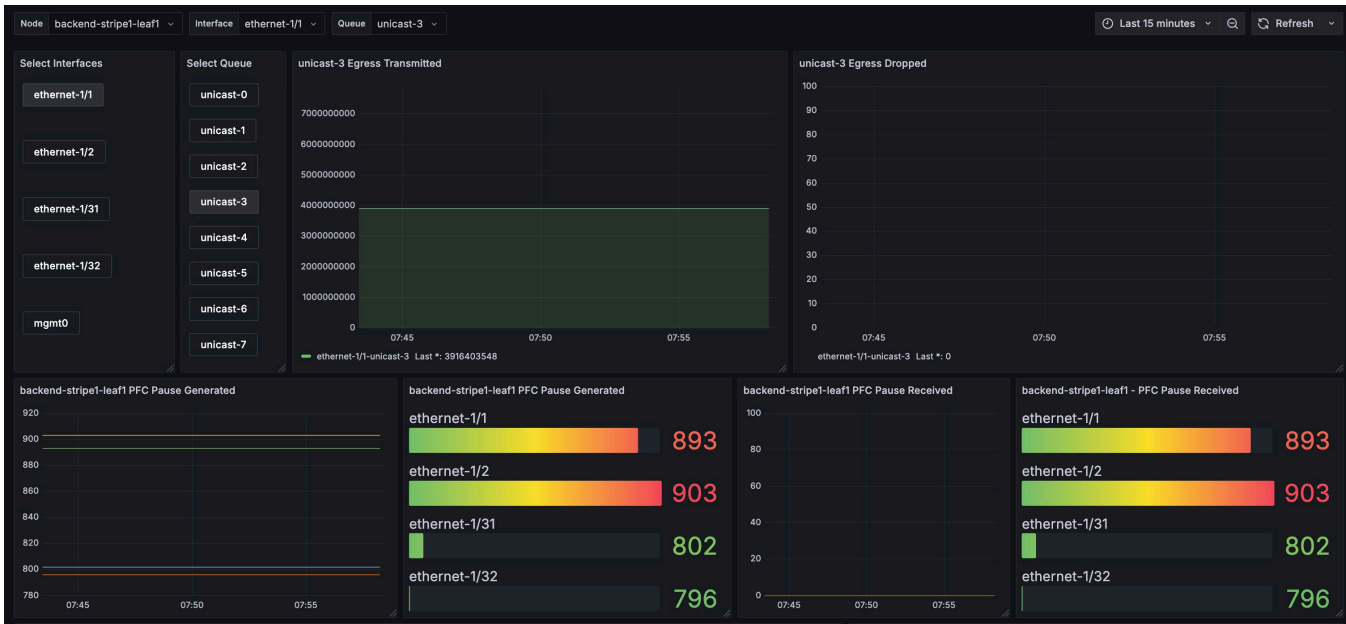
There are pre-built dashboards in Grafana, as shown below:



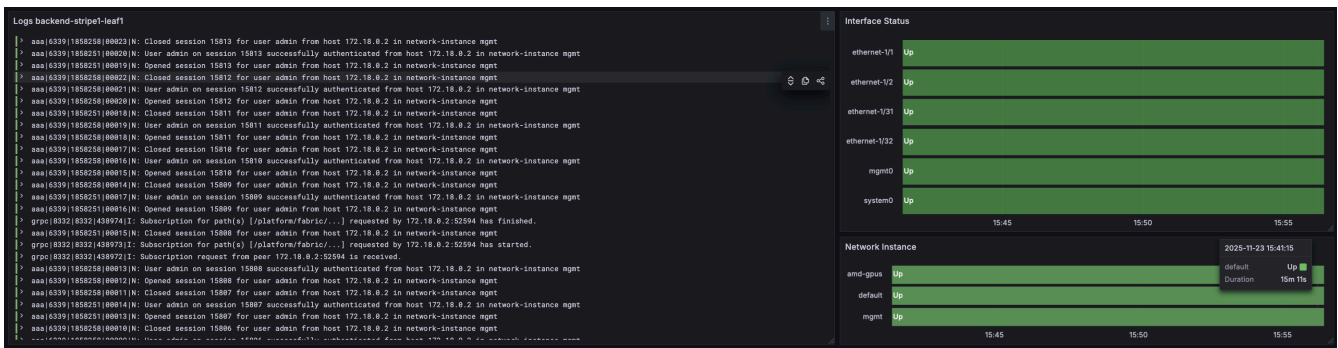
View 1: This dashboard shows statistics collectively from all the data center nodes. At the top right, the display shows the total traffic trend, below that the Top Talkers interfaces in both directions, and to the left, individual nodes with their traffic trend.



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>