BLACK BOOK

AI Fabric Test Methodology

KEYSIGHT

Table of Contents

Fabric Validation

During Large Language Model (LLM) training, massive data transmissions between GPU nodes can lead to bottlenecks and slow down the training process. A well-designed network fabric is crucial to enable efficient data movement, reduce latency, and facilitate faster training times. This black book aims to prescribe a consistent and repeatable test process to provide measurable metrics with quantifiable KPIs (key performance indicator) that can be used to benchmark various implementations and ensure that data center operators optimize infrastructures for AI workloads. Following the methodologies presented here enables an organization to have better performance, scalability, and fault tolerance within AI data centers.

Overview

In the context of Artificial Intelligence (AI) systems, particularly LLMs, a network fabric refers to the underlying infrastructure that connects and enables communication between various components or nodes within the system. The network fabric is crucial in facilitating efficient and effective AI system operation, including LLM training. The network fabric should have the following abilities:

- Data movement
- Scalability
- Fault tolerance
- Inter-node communication
- Resource allocation
- Monitoring and debugging

AI model training

Step 1: Data preparation

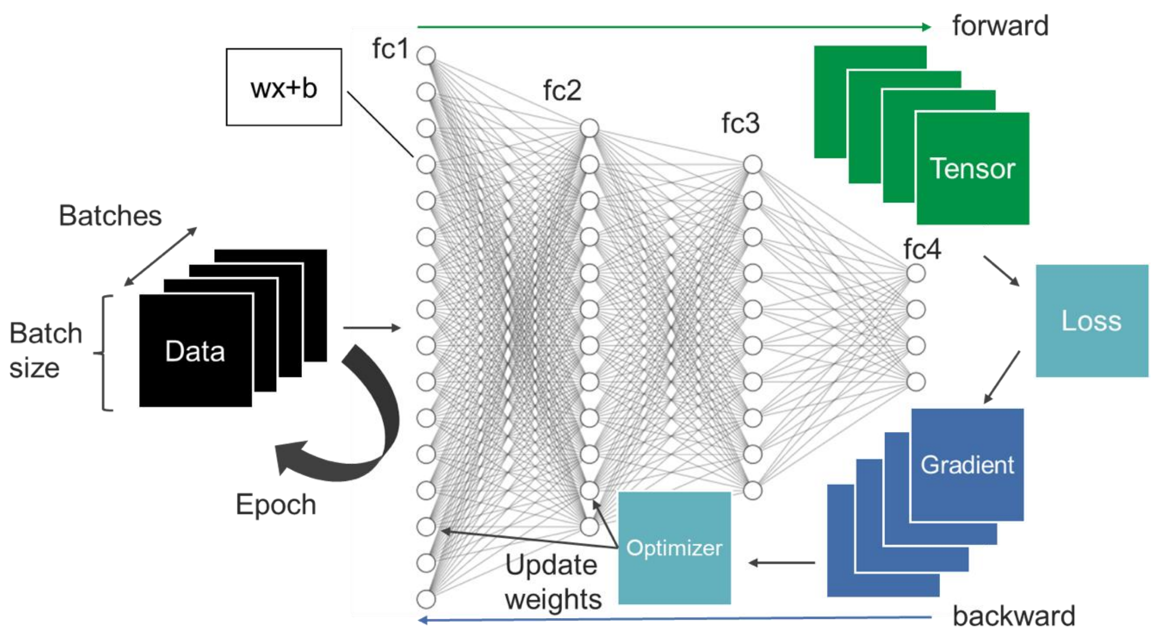
1. Collect and preprocess large datasets (for example, text files, images, and audio).
2. Tokenize and normalize data to ensure consistency and efficiency.
3. Split data into training, validation, and testing sets.

Step 2: Model definition

1. Define the architecture of the AI model (for example, neural network and decision tree).
2. Specify hyperparameters (for example, learning rate, batch size, and number of layers).

Step 3: Model training

1. Initialize the model's weights and biases.
2. Feedforward pass: Compute outputs for each sample in the training set.
3. Backpropagation: Calculate gradients and update model parameters by using an optimization algorithm (for example, Stochastic Gradient Descent and Adam).
4. Repeat the preceding steps until convergence or a stopping criterion is reached.



Parallel training and collective communication

In distributed AI training, ‘parallelism’ refers to the ability to perform multiple computations simultaneously on different nodes or devices. This is achieved by dividing the workload into smaller tasks that can run concurrently while ‘collective communications’ ensure that these tasks can effectively coordinate to achieve the required outcome.

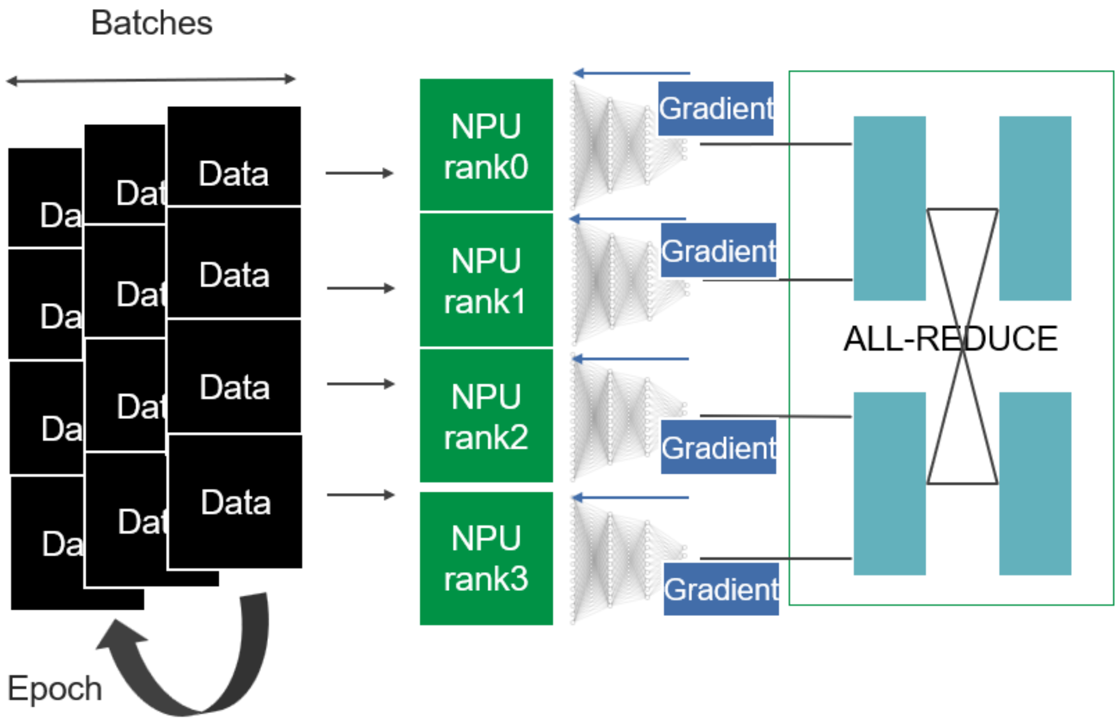
Data parallelism (DP) optimizes the use of multiple Graphical Processing Units (GPUs) to train models in parallel. The training dataset is evenly distributed among all GPUs, with each GPU maintaining a replica of the entire model. In each training batch, all GPUs engage in an all-reduce operation to synchronize their calculated gradients, ensuring consistent and accurate model updates. This efficient use of resources is a key advantage of data parallelism.

Pipeline parallelism (PP) partitions a model into multiple stages, each comprising sequential layers assigned to distinct GPUs. Within this pipeline, each GPU receives input from the preceding stage and forwards its output to the subsequent stage, enabling efficient processing of complex models across distributed computing resources.

Tensor parallelism (TP) enables further horizontal partitioning of models or layers within pipeline-parallelized stages. Each layer is thereby distributed across a cluster of GPUs, with each GPU in the same group collaborating through all-reduce / all-gather operations to synchronize calculated outputs and gradients, facilitating efficient model training on large-scale datasets.

Some key collectives used in parallel training are as follows:

- All-Reduce: A collective operation that reduces a tensor across all processes to a single value. Used for calculating average gradients, model updates, or aggregation of tensors.
- All-Gather: A collective operation that gathers a tensor from each process and returns it as a concatenated tensor. Used for aggregating gradients, features, or labels across processes.
- Reduce-Scatter: A collective operation that reduces a tensor along one dimension and scatters the result to all processes. Used for calculating average gradients or model updates.
- All-to-All: A collective operation that broadcasts tensors from each process to all other processes. Used for sharing intermediate results such as gradients or activations, across processes.
- Broadcast: A collective operation that broadcasts a tensor from one process to all other processes. Used for sharing model parameters, hyperparameters, or other global variables.



Parallel computing techniques can significantly accelerate the training process by processing large datasets and complex models concurrently.

Challenges

LLM has revolutionized AI and cloud services with its tremendous impact. With hundreds of billions of parameters, LLM training relies on massive, distributed training clusters, typically comprising tens of thousands of GPUs. This unique characteristic poses new challenges for designing data center networks, requiring innovative solutions to efficiently manage the scale and complexity of LLM training. Operators now must carefully quantify the following:

- Traffic patterns. The traffic patterns of LLM training differ significantly from those of traditional cloud computing, characterized by elephant flows with low entropy and bursty traffic.
- Multi-tenant and parallel training jobs. Parallel applications or workflows, which generate traffic that competes with other processes for shared infrastructure resources like the network, can negatively impact overall system performance.
- Congestion control. AI systems require high reliability to ensure accurate model training and deployment. Lossless Remote Directory Memory Access (RDMA) guarantees that data is reliably delivered, reducing the risk of errors or data loss during transmission.
- Network Interface Card (NIC) capacity. To optimize network usage, NICs must employ effective congestion control strategies that ensure available bandwidth is used efficiently, thereby reducing the risk of bottlenecks, and enhancing overall system performance.
- Sensitive to faults. The LLM training process is synchronous, requiring all GPUs to cooperate and complete a series of iterations simultaneously. As such, an anomaly in any GPU can delay or crash the entire training process, rendering LLM training more fault-sensitive than traditional cloud computing.

General topology

Testing an LLM training network fabric is crucial to ensure it can handle the massive data transmissions and processing requirements of LLM training, as shown in Figure 1. A robust, standards-based benchmarking methodology is essential for comparing performance results, providing a reliable means of evaluating outcomes by using quantifiable metrics that facilitate accurate assessments and informed decision-making.

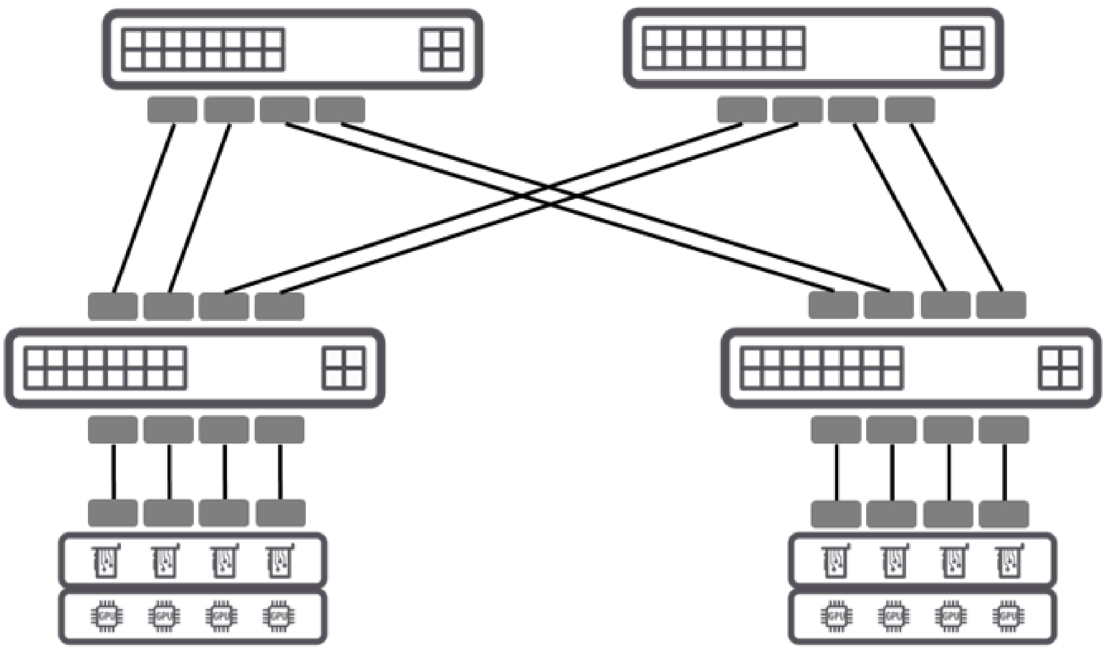


Figure 1. General test topology with GPU / NPU racks

With its complex architecture comprising spine and leaf fabrics, as well as load sharing capabilities across Equal-cost Multipath Routing (ECMP), we require a robust traffic generator / analyzer tool like Keysight AresONE-M running with Keysight Collective Benchmarks to connect to the Internet Protocol (IP) Clos fabric, as shown in Figure 2. This necessitates connecting the Collective Benchmarks (CB) test ports to the Top-of-Rack (ToR) switch to emulate the Neural Processing Unit (NPU) hosts with NICs. The CB and AresONE-M are applications and interfaces supported by the Keysight AI (KAI) Data Center Builder.

Setup

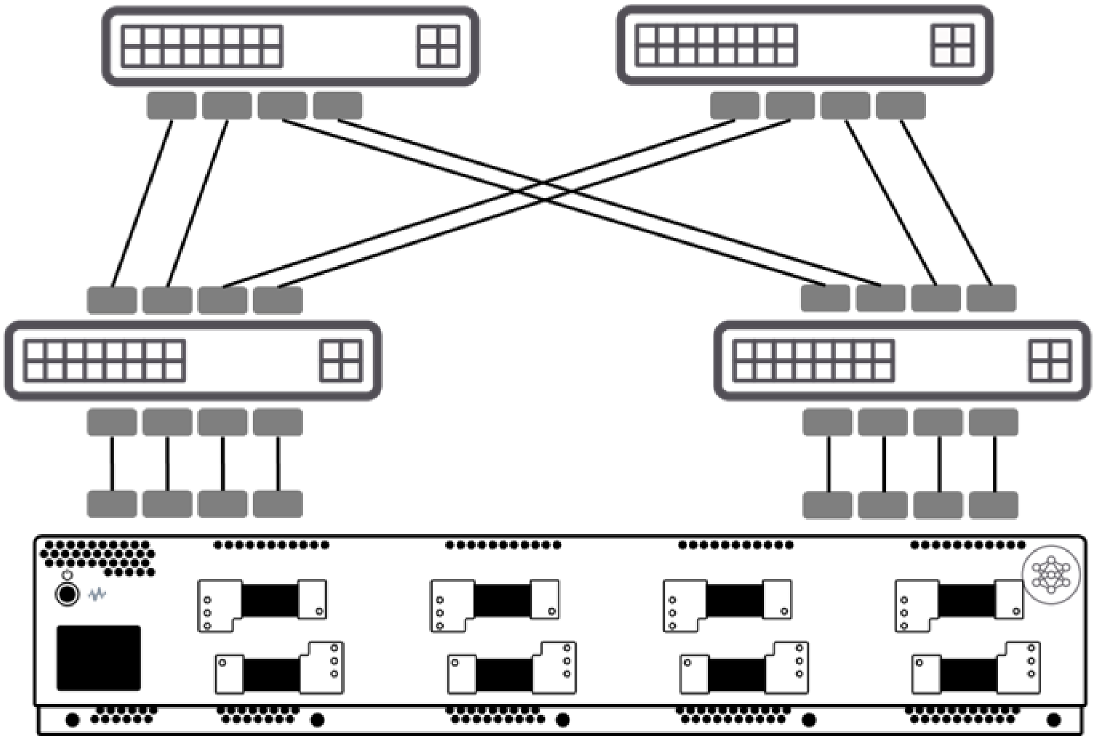


Figure 2. Test topology with CB application

The key benefits of this test topology with CB enable the ability to:

- Emulate the AI workload with collective communications.
- Achieve high-throughput performance even with massive collective data sizes.
- Emulate multi-tenant AI workload.
- Emulate congestion control. To simulate realistic network conditions, we emulate congestion control mechanisms and backpressure effects on the NIC, allowing us to thoroughly test and optimize our system's performance under various traffic scenarios.
- Quantify the training jobs. To gain a comprehensive understanding of our training process, we quantify the completion time of each job and evaluate the algorithm's bandwidth usage, providing valuable insights into optimization opportunities.
- Provide RDMA Queue Pair (QP) based analysis. To gain deeper insights into the underlying performance, we conduct an analysis based on RDMA flow metrics, allowing us to better comprehend the results and identify potential areas for optimization.
- Introduce impairment. To thoroughly assess the robustness of our network fabric, we intentionally introduce deliberate failures and stress testing scenarios to emulate real-world conditions, thereby evaluating its resilience and identifying opportunities for improvement.

Test tool

The KAI Data Center Builder with CB application is designed to run micro-benchmarking for typical AI communications algorithms on the user-provided AI network fabric to:

- Evaluate AI network fabric performance for common types of collective communications.
- Measure performance metrics, including job completion time, algorithm bandwidth, and bus bandwidth. It also allows you to calculate the ideal percentage to quantify deviations from theoretical maximum performance.
- Use AresONE hardware to measure and analyze Queue Pair (QP) (AI data flows) performance, to summarize results as percentiles with drill-down capabilities for further analysis.
- Assess RDMA over Converged Ethernet v2 (RoCEv2) emulation fidelity by comparing AresONE hardware results with metrics that are collected on actual AI systems.

Test Methodology for Job Completion Time

To validate the performance of AI network fabric, we aim to support diverse AI training jobs and parallelism strategies by conducting rigorous testing with variant collective data sizes, thereby achieving high throughput, and ensuring optimal resource usage.

In this experiment, we emulate a scenario featuring two NPU hosts, each equipped with four NPU and NICs. We employ the ring algorithm to test all-gather and all-reduce collective operations, with traffic flowing in a circular pattern from node 0 to 7. Under these conditions, the network fabric experiences no congestion, enabling us to benchmark the maximum performance of the collective operations.

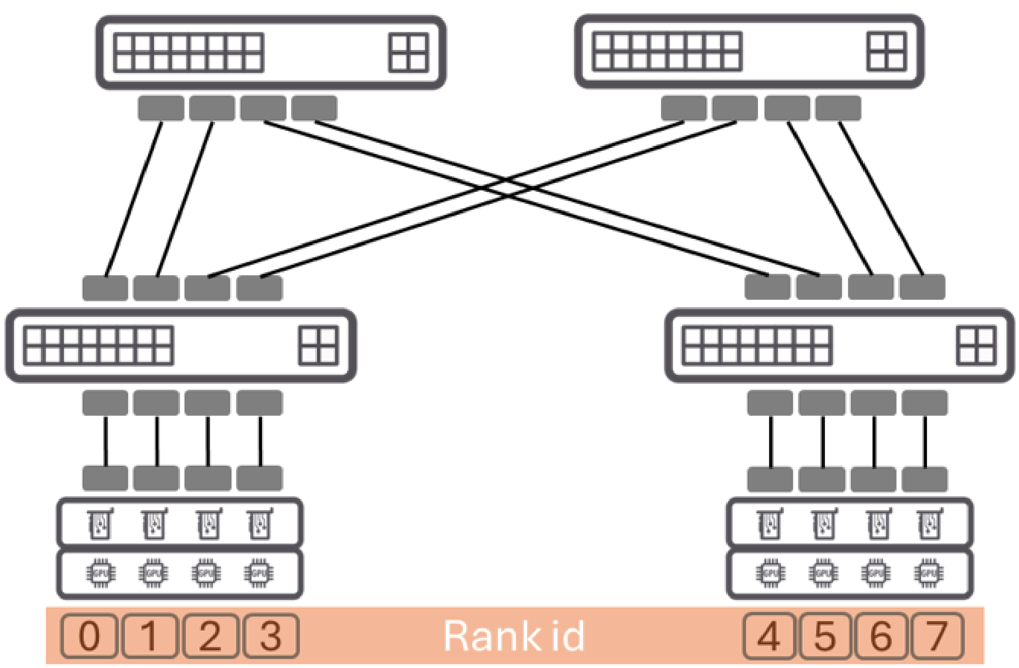


Figure 3. JCT test topology

Perform the following test procedure:

1. Set the peering test ports with IP and gateway.
2. Set the rank id as [0, 1, 2, 3, 4, 5, 6, 7], which represents the emulated NPUs and their association with the physical AresONE Ethernet ports.
3. Set the data size.
4. Set the algorithm.
5. Run trials.

Another variance is to test a single device under test (DUT), which is a leaf or spine switch, as shown in Figure 4.

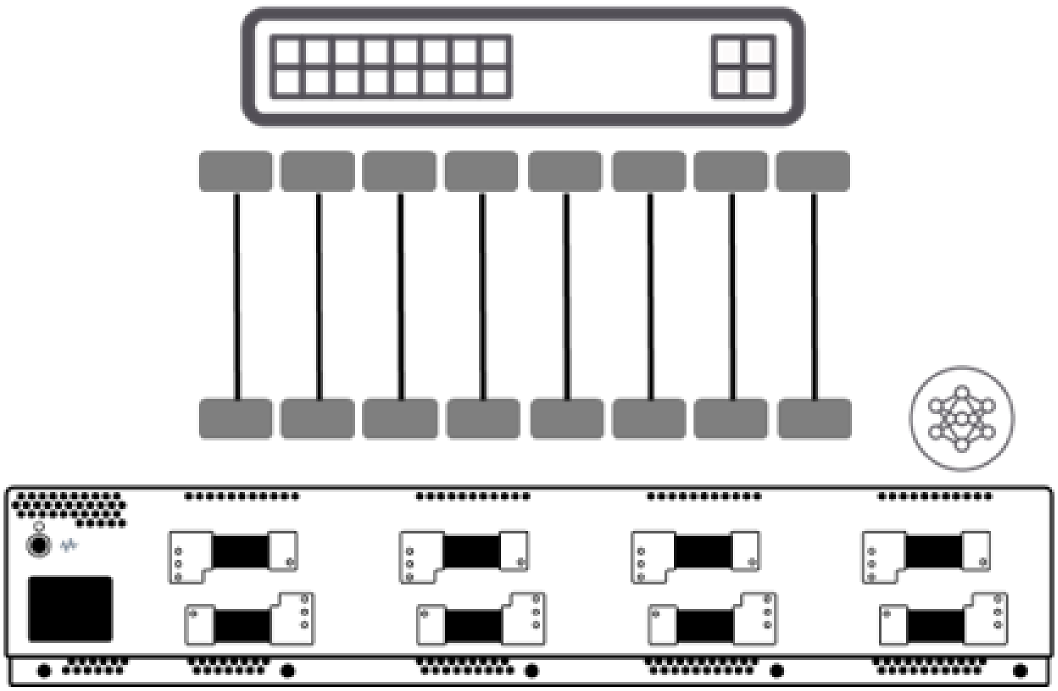


Figure 4. Single DUT test topology

The following test cases demonstrate the test methodologies and detailed step-by-step configuration with automated test packages that are available in CB:

Test collectives	Data size	Write msg size	Parallel QPs	Congestion control
All-reduce ring	16 MB-8 GB	128 KB	1	N
All-to-all	16 MB-8 GB	128 KB	1	N
All-reduce ring	200 GB	128 KB	1	N
All-reduce halving doubling	4 GB	128 KB	1	N

Test case 1: Ring algorithm Job Completion Time (JCT)

Overview

This test case employs an all-reduce unidirectional ring collective operation to evaluate the data forwarding performance of the fabric by using a list of data size to emulate the transmission of numerous data chunks through the ToR and pod switches. The System Under Test (SUT) consists of the ToR switches, which are interconnected with multiple pod switches forming redundant paths through ECMP.

About All-Reduce Ring algorithm

All-reduce unidirectional ring collective is a popular distributed communication pattern used in various parallel computing applications, such as machine learning, scientific simulations, and data processing. It's a fundamental building block for many distributed algorithms, enabling efficient communication and aggregation of data across multiple nodes.

In this algorithm, collective data is split into *n* distinct chunks, where ***n*** corresponds to the number of NPU nodes. Each chunk then propagates through a ring-based communication pattern, comprising ***n-1*** iterations for the reduce-scatter operation and another ***n-1*** iteration for all-gather, as depicted in Figure 5.

For a collective of ***n*** NPU nodes, a total of ***2(n-1)*** units of collective data are transmitted within the fabric during communication. For example, in an all-reduce unidirectional ring with 4 NPU nodes, transmitting 4 GB of collective data requires ***2*3*4*** GB = **24** GB of data to be present in the fabric.

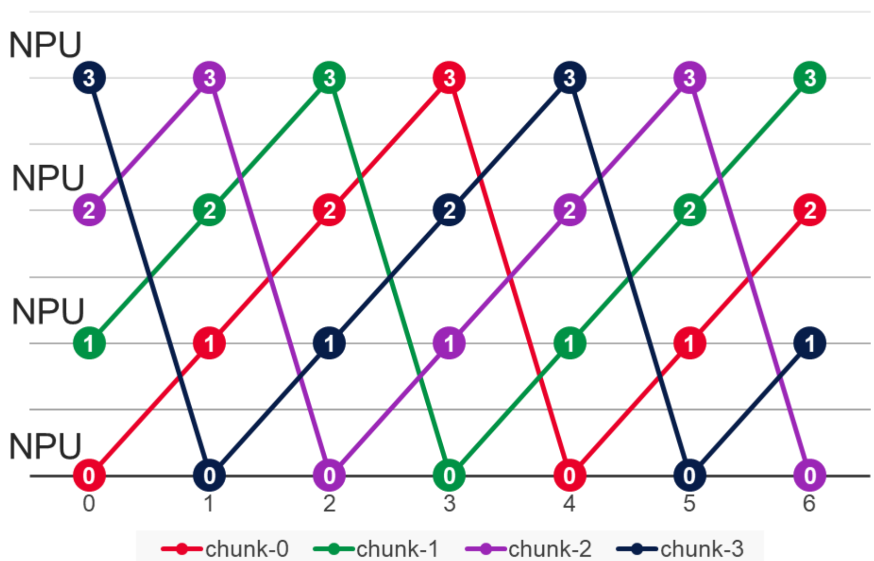


Figure 5. All-reduce unidirectional ring steps for 4 NPUs

Unlike port-based tracking, the ring algorithm relies on a chunk-based approach. To mitigate long-tail latency effects, it's essential to track each data chunk individually, as illustrated in Figure 6.

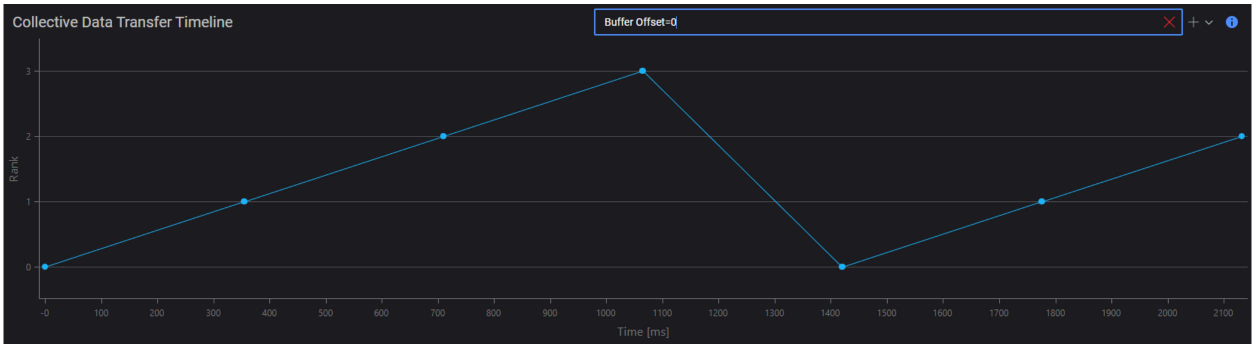
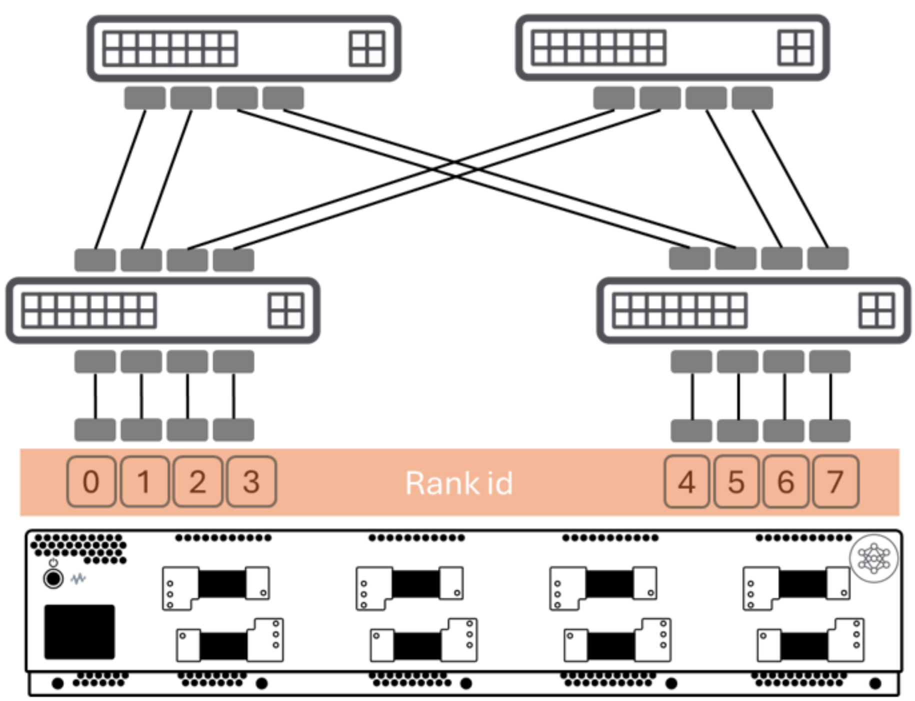


Figure 6. All-reduce unidirectional ring data chunk tracking

Objectives

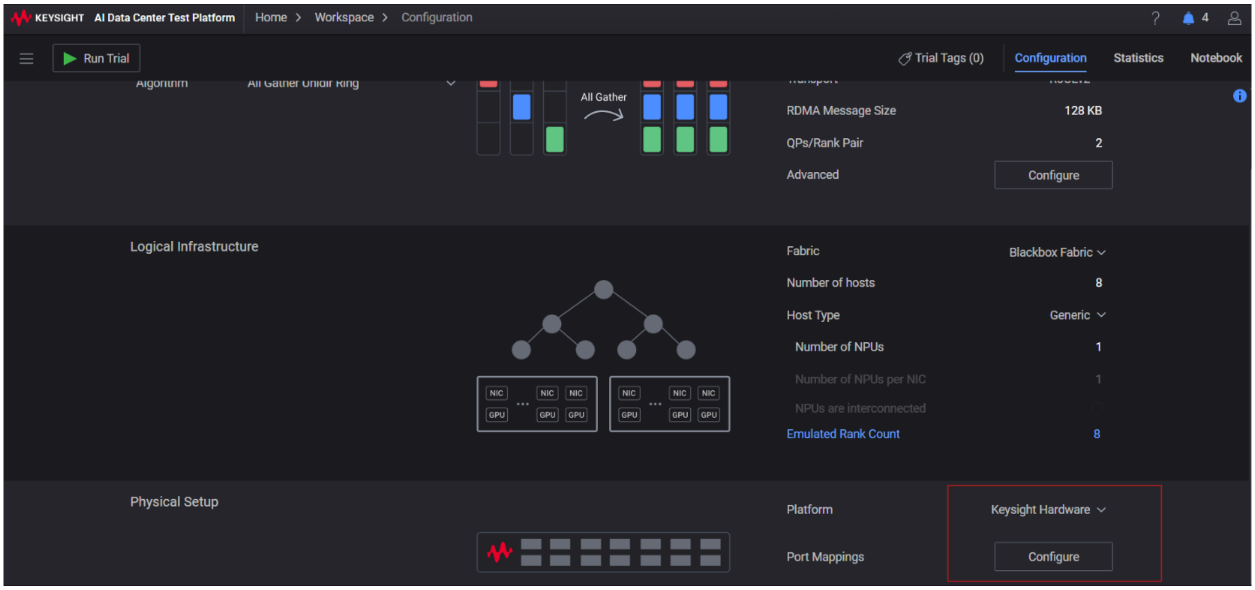
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing on data sizes.
3. Provide chunk-based tracking.

Setup

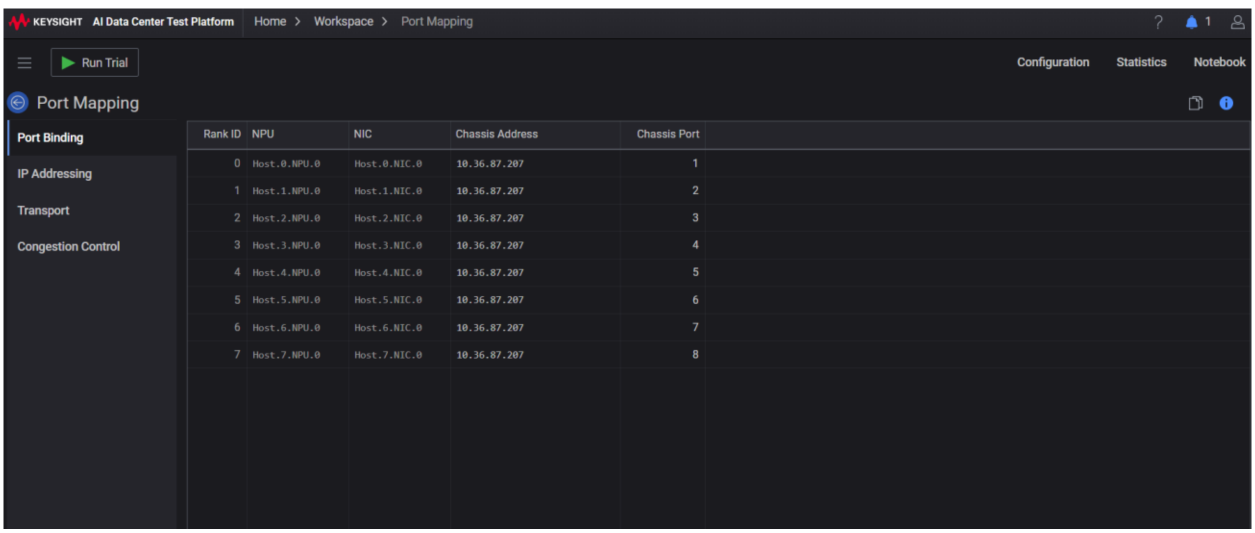


Step-by-step instructions

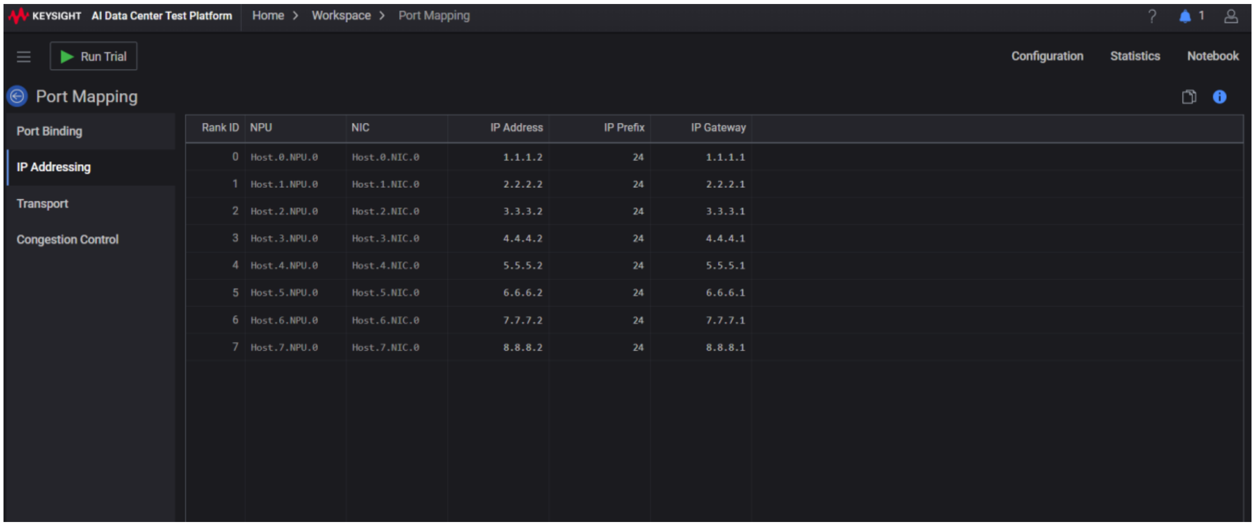
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



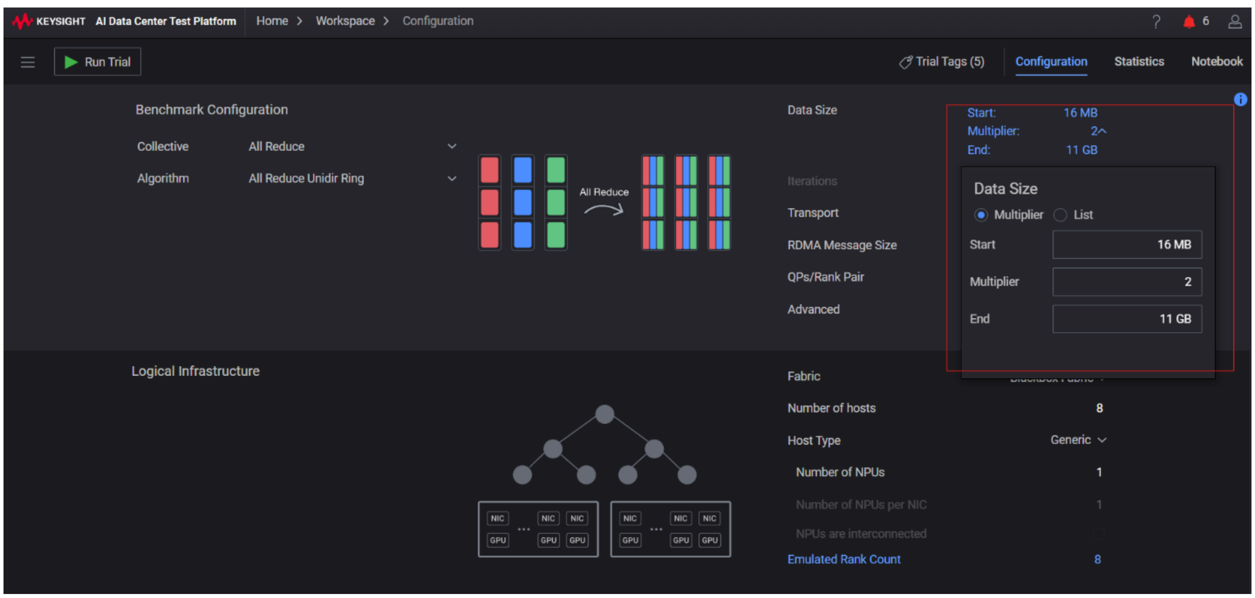
2. Set the port binding.



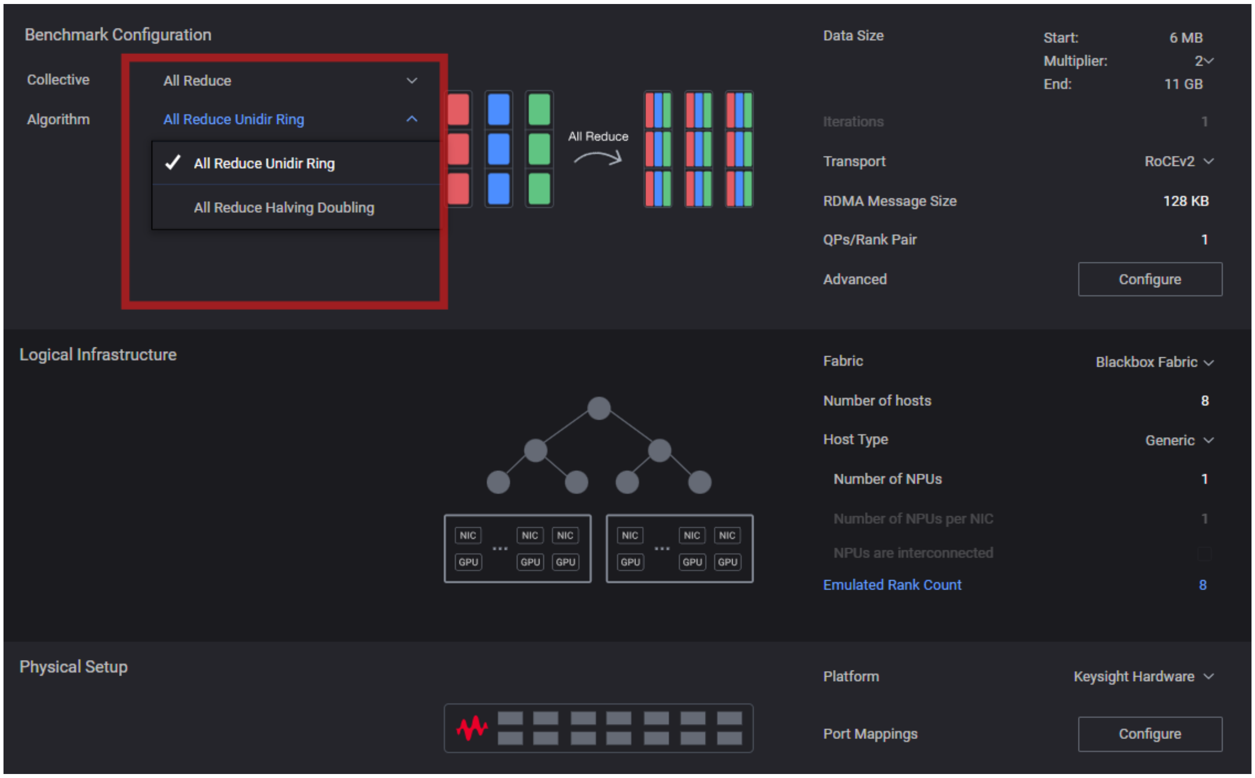
3. Set the IP address and gateway.



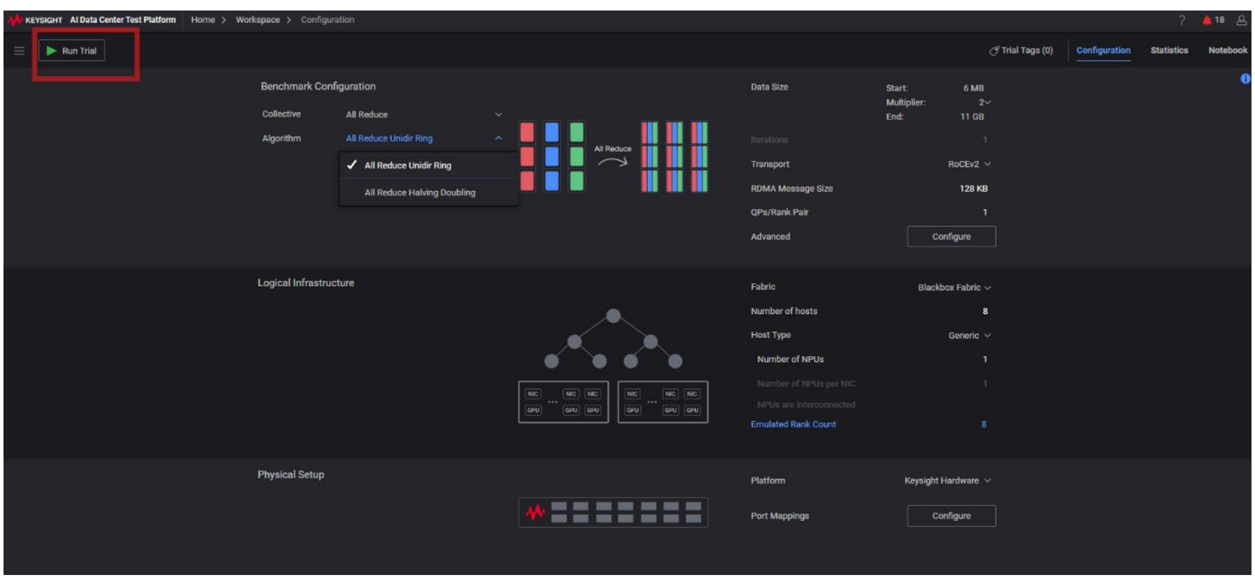
4. Set the data size from 16 MB to 11 GB with a multiplier of 2.



5. Set the collective and algorithms.



6. Select Run Trial.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE Maximum Transmission Unit (MTU)

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and Cyclic Redundancy Check (CRC) fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — InfiniBand (IB) MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other ring algorithms to test: All-Gather-Ring and Reduce-Scatter-Ring.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

- RDMA Write message size

  Within a single Queue Pair (QP) in collectives, multiple RDMA Write operations are performed, with each operation split into separate chunks based on the RDMA write message size for the data chunk. Within these RDMA Write operations, each packet is defined by its corresponding RDMA MTU.

- Parallel-QPs

  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

Results analysis

From the test results, we observed that the completion time is measured in milliseconds and algorithm bandwidth is calculated accordingly. The ideal percentage against bus bandwidth is high, indicating no congestion and high throughput during forwarding. Priority Flow Control (PFC), Explicit Congestion Notification – Congestion Experienced (ECN-CE), and Congestion Notification Packet (CNP) numbers support the congestion statistics.

- Algorithm bandwidth

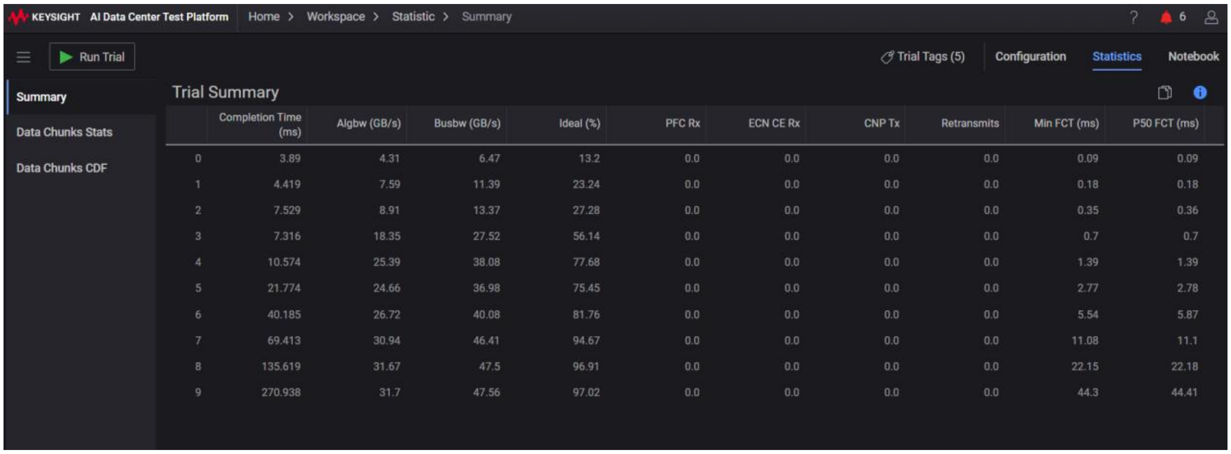
  Algorithm bandwidth, calculated as Collective Size / Time

- Bus bandwidth

  Bus bandwidth, calculated as Algorithm Bandwidth multiplied by a collective-dependent compensation factor

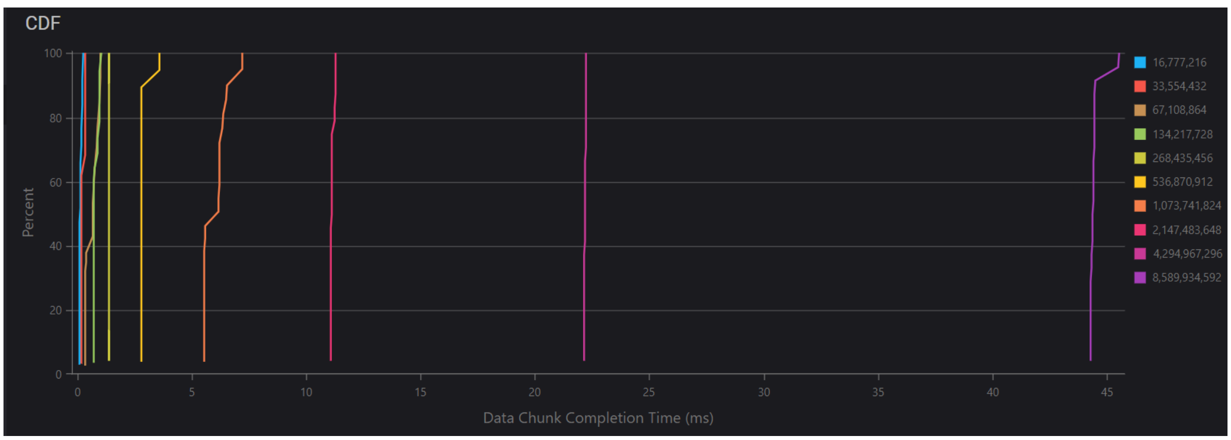
- Ideal percentage

  Bus bandwidth divided by ideal goodput (as percentage)

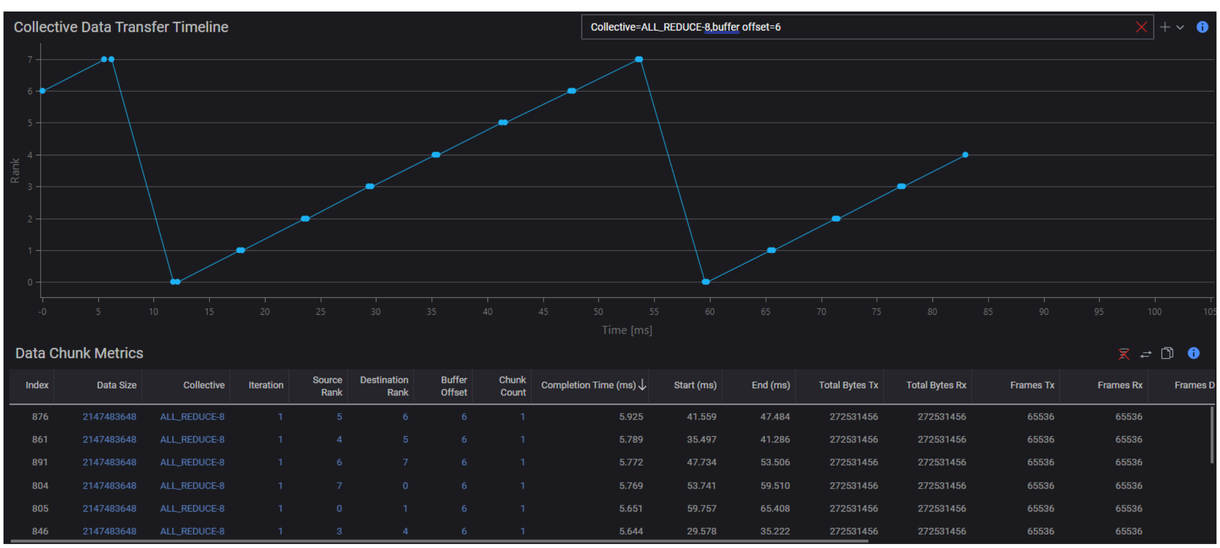


	Completion Time (ms)	Algbw (GB/s)	Busbw (GB/s)	Ideal (%)	PFC Rx	ECN CE Rx	CNP Tx	Retransmits	Min FCT (ms)	P50 FCT (ms)
0	3.89	4.31	6.47	13.2	0.0	0.0	0.0	0.0	0.09	0.09
1	4.419	7.59	11.39	23.24	0.0	0.0	0.0	0.0	0.18	0.18
2	7.529	8.91	13.37	27.28	0.0	0.0	0.0	0.0	0.35	0.36
3	7.316	18.35	27.52	56.14	0.0	0.0	0.0	0.0	0.7	0.7
4	10.574	25.39	38.08	77.68	0.0	0.0	0.0	0.0	1.39	1.39
5	21.774	24.66	36.98	75.45	0.0	0.0	0.0	0.0	2.77	2.78
6	40.185	26.72	40.08	81.76	0.0	0.0	0.0	0.0	5.54	5.87
7	69.413	30.94	46.41	94.67	0.0	0.0	0.0	0.0	11.08	11.1
8	135.619	31.67	47.5	96.91	0.0	0.0	0.0	0.0	22.15	22.18
9	270.938	31.7	47.56	97.02	0.0	0.0	0.0	0.0	44.3	44.41

- Long tail latency distribution for data sizes



- Chunk-based tracking for big completion time QP



Index	Data Size	Collective	Iteration	Source Rank	Destination Rank	Buffer Offset	Chunk Count	Completion Time (ms)	Start (ms)	End (ms)	Total Bytes Tx	Total Bytes Rx	Frames Tx	Frames Rx	Frames D
876	2147483648	ALL_REDUCE-8	1	5	6	6	1	5.925	41.559	47.484	272531456	272531456	65536	65536	
861	2147483648	ALL_REDUCE-8	1	4	5	6	1	5.789	35.497	41.286	272531456	272531456	65536	65536	
891	2147483648	ALL_REDUCE-8	1	6	7	6	1	5.772	47.734	53.506	272531456	272531456	65536	65536	
804	2147483648	ALL_REDUCE-8	1	7	0	6	1	5.769	53.741	59.510	272531456	272531456	65536	65536	
805	2147483648	ALL_REDUCE-8	1	0	1	6	1	5.651	59.757	65.408	272531456	272531456	65536	65536	
846	2147483648	ALL_REDUCE-8	1	3	4	6	1	5.644	29.578	35.222	272531456	272531456	65536	65536	

Conclusion

We observe that with the increment of data size, collectives get higher algorithm bandwidth and bus bandwidth to achieve better ideal ratio.

Test case 2: All-to-all parallel JCT

Overview

This test case employs an all-to-all collective operation to evaluate the data forwarding performance of the fabric by using a list of data sizes to emulate the transmission of numerous data chunks through the ToR and pod switches. The SUT consists of the ToR switches, which are interconnected with multiple pod switches forming redundant paths through ECMP.

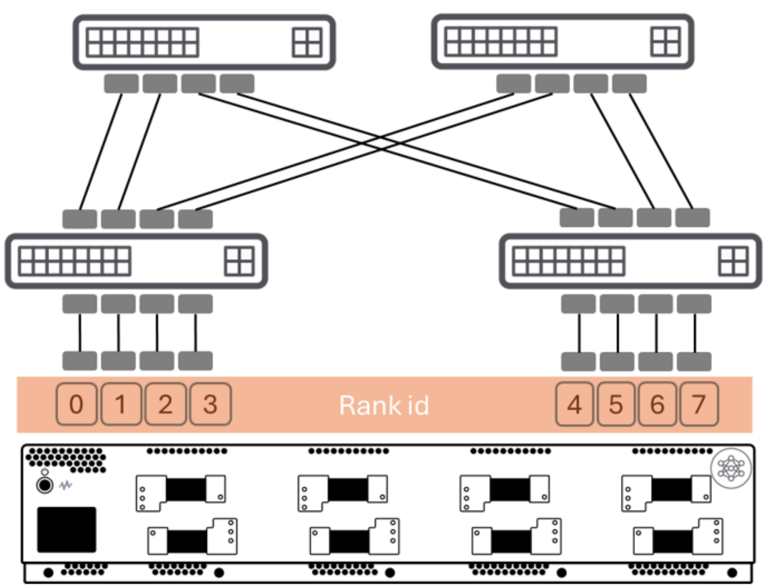
About All-to-all parallel

All-to-all is a collective operation that sends and receives data between all participating nodes parallelly.

Objectives

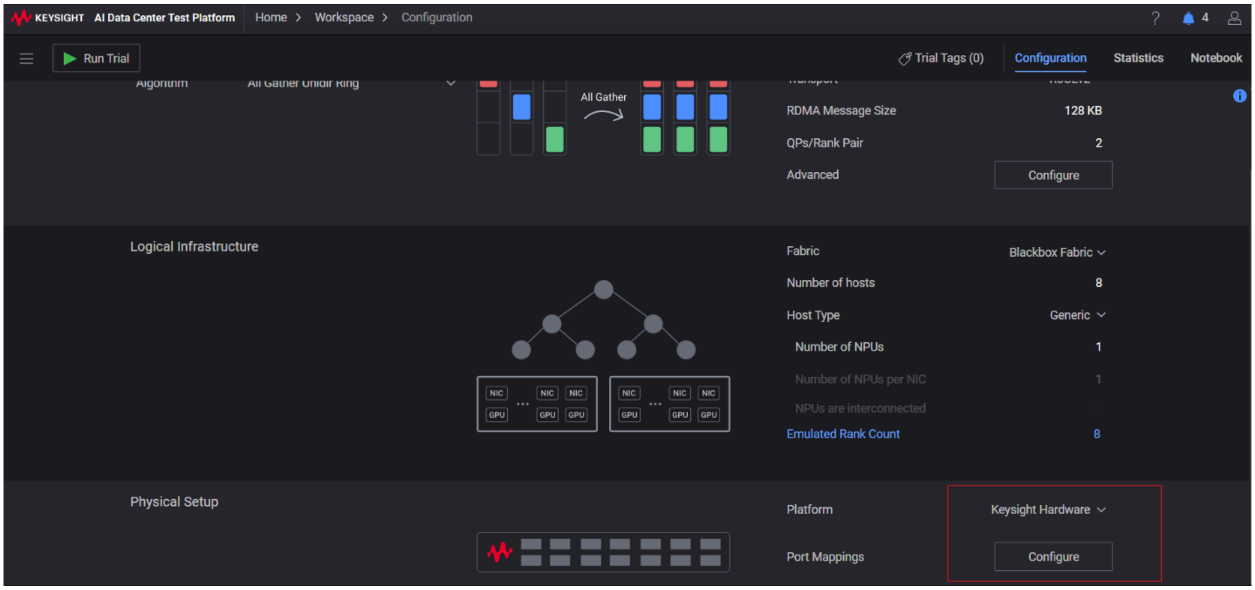
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing on data sizes.

Setup

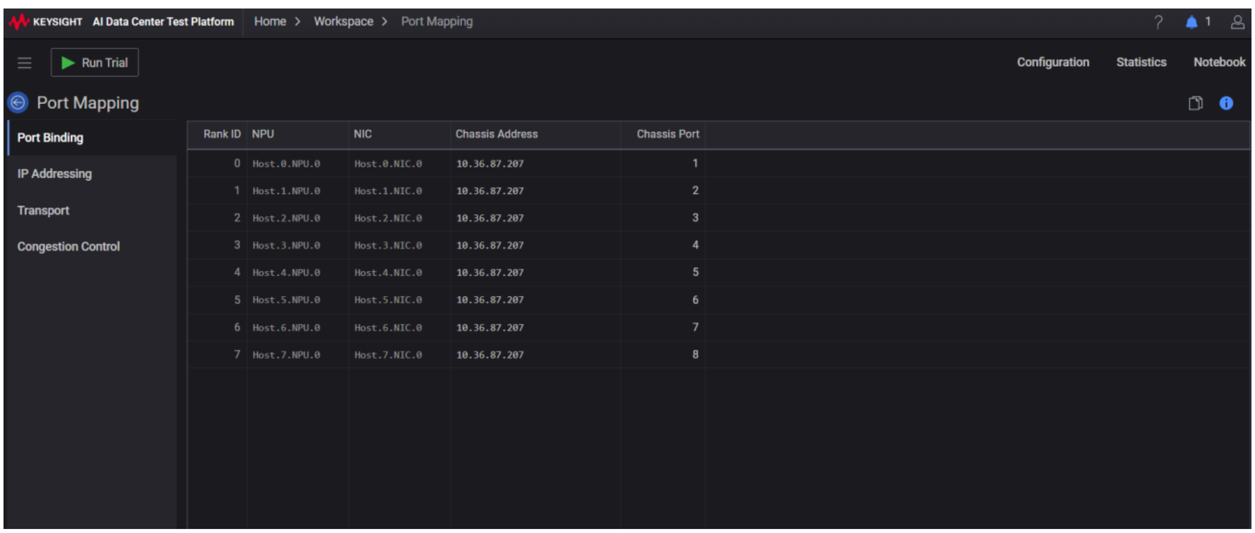


Step-by-step instructions

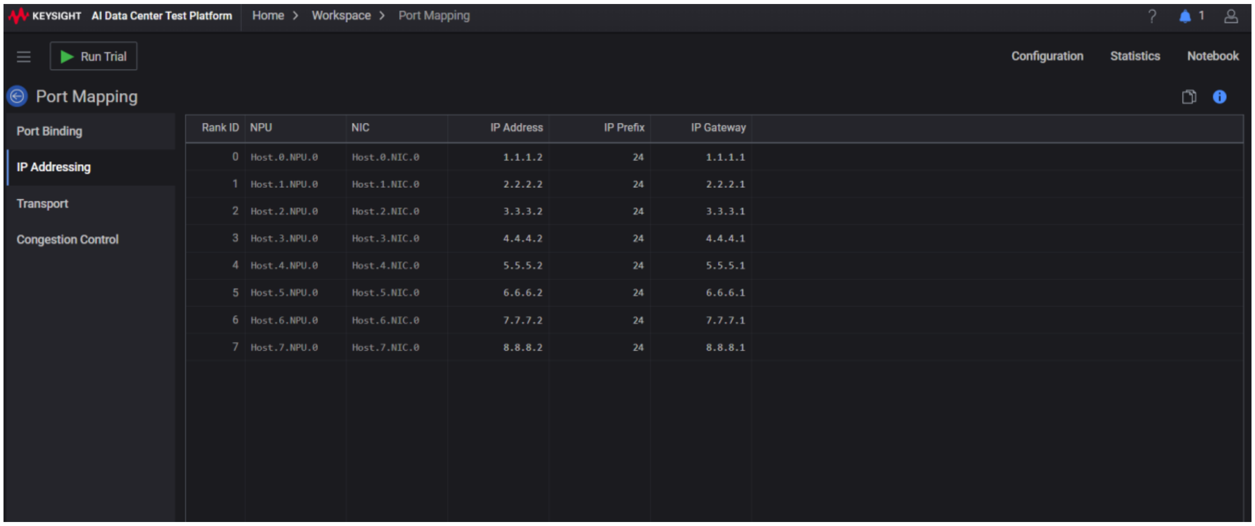
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



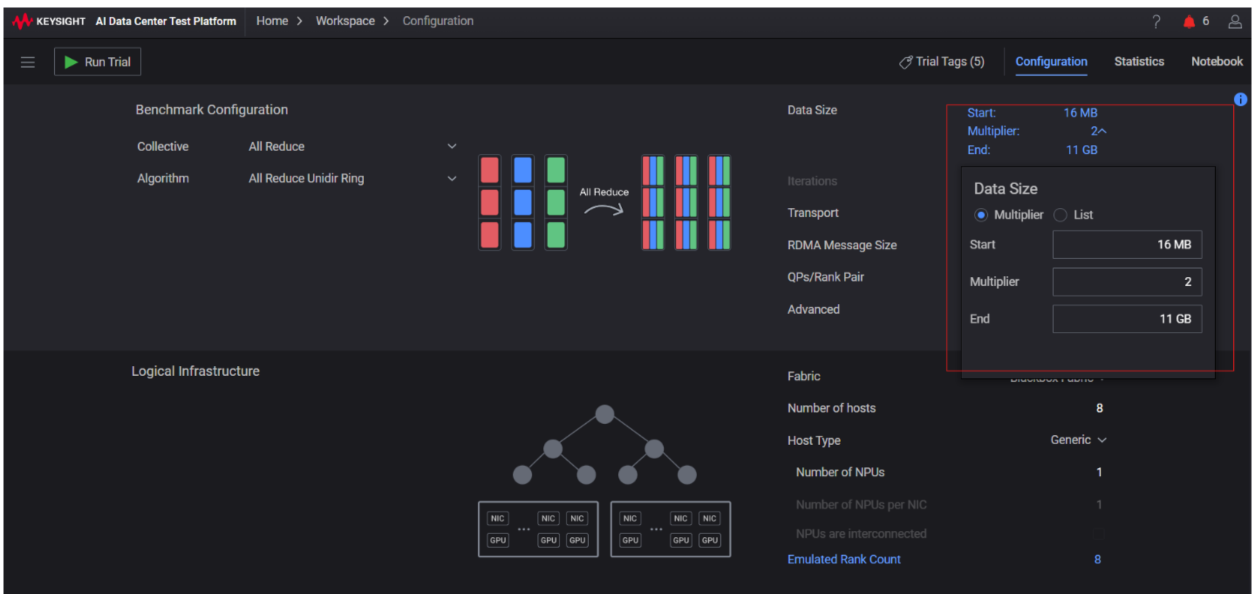
2. Set the port binding.



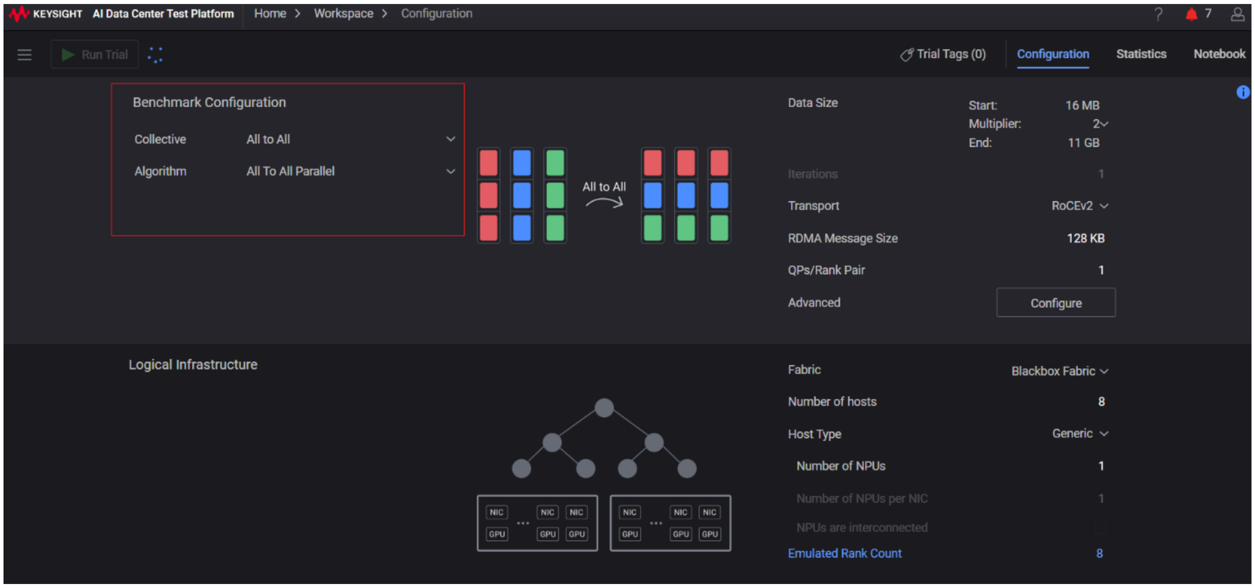
3. Set the IP address and gateway.



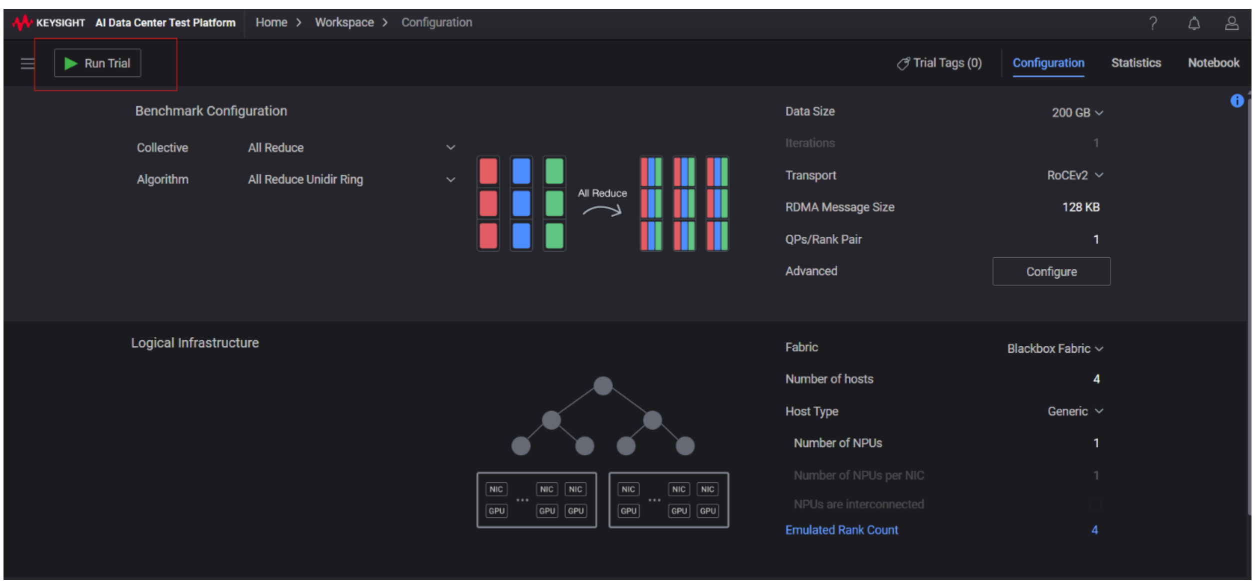
4. Set the data size from 16 MB to 11 GB with a multiplier of 2.



5. Set the collective and algorithms.



6. Select Run Trial.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU Select, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ’active MTU.’

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

Results analysis

From the test results, we observed that the completion time is measured in milliseconds and algorithm bandwidth is calculated accordingly. The ideal percentage against bus bandwidth is high, indicating no congestion and high throughput during forwarding. PFC, ECN-CE, and CNP numbers support the congestion statistics.

- Algorithm bandwidth

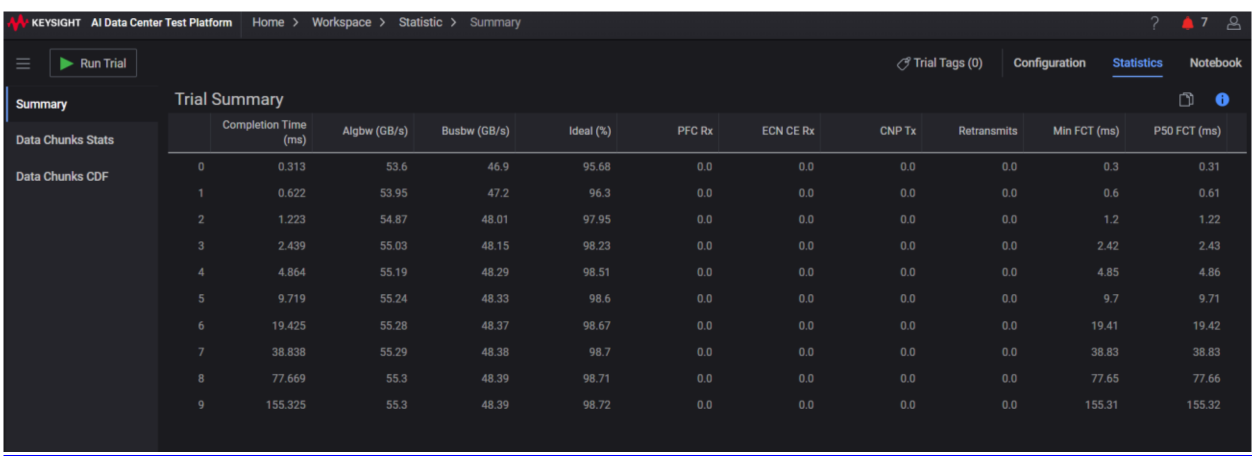
  Algorithm bandwidth, calculated as Collective Size / Time

- Bus bandwidth

  Bus bandwidth, calculated as Algorithm Bandwidth multiplied by a collective-dependent compensation factor

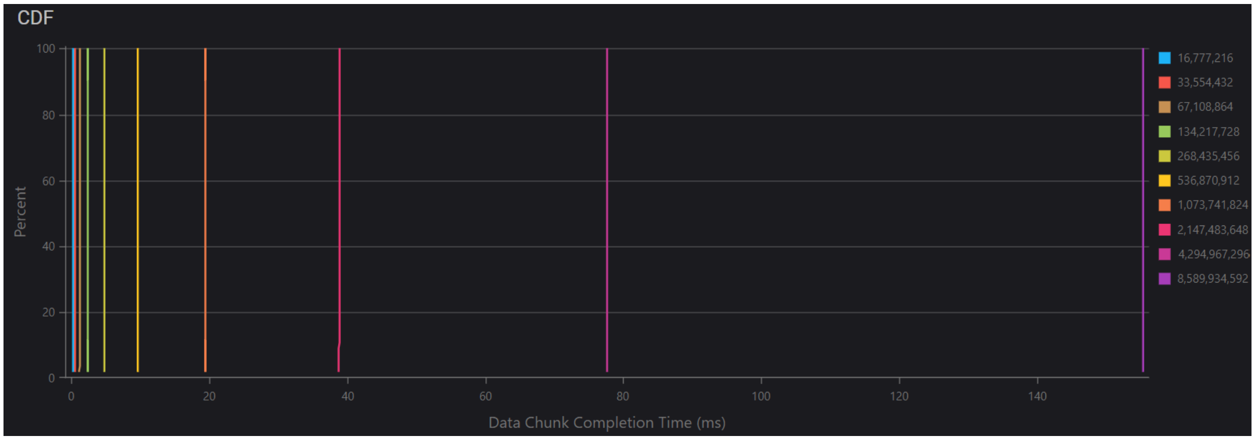
- Ideal percentage

  Bus bandwidth divided by ideal goodput (as percentage)



	Completion Time (ms)	Algbw (GB/s)	Busbw (GB/s)	Ideal (%)	PFC Rx	ECN CE Rx	CNP Tx	Retransmits	Min FCT (ms)	P50 FCT (ms)
0	0.313	53.6	46.9	95.68	0.0	0.0	0.0	0.0	0.3	0.31
1	0.622	53.95	47.2	96.3	0.0	0.0	0.0	0.0	0.6	0.61
2	1.223	54.87	48.01	97.95	0.0	0.0	0.0	0.0	1.2	1.22
3	2.439	55.03	48.15	98.23	0.0	0.0	0.0	0.0	2.42	2.43
4	4.864	55.19	48.29	98.51	0.0	0.0	0.0	0.0	4.85	4.86
5	9.719	55.24	48.33	98.6	0.0	0.0	0.0	0.0	9.7	9.71
6	19.425	55.28	48.37	98.67	0.0	0.0	0.0	0.0	19.41	19.42
7	38.838	55.29	48.38	98.7	0.0	0.0	0.0	0.0	38.83	38.83
8	77.669	55.3	48.39	98.71	0.0	0.0	0.0	0.0	77.65	77.66
9	155.325	55.3	48.39	98.72	0.0	0.0	0.0	0.0	155.31	155.32

- Long tail latency distribution for data sizes



Conclusion

We observe that with the increment of data size, collectives get higher algorithm bandwidth and bus bandwidth to achieve better ideal ratio.

Test Case 3: Collective JCT with extra big data size

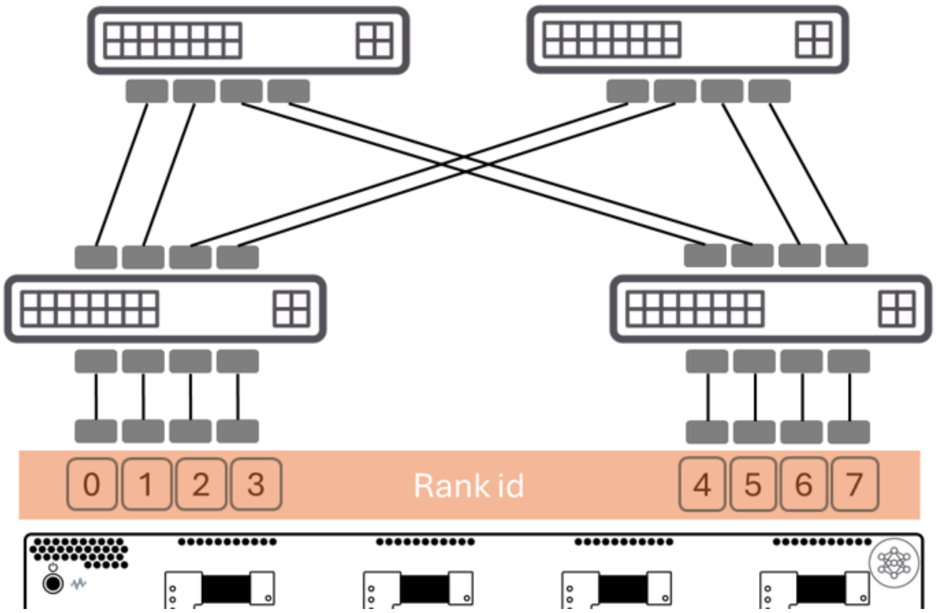
Overview

This test case employs an all-reduce-ring collective operation to evaluate the data forwarding performance of the fabric by using a massive 200G data size to simulate the transmission of numerous data chunks through the ToR and pod switches. The SUT consists of the ToR switches, which are interconnected with multiple pod switches forming redundant paths through ECMP.

Objectives

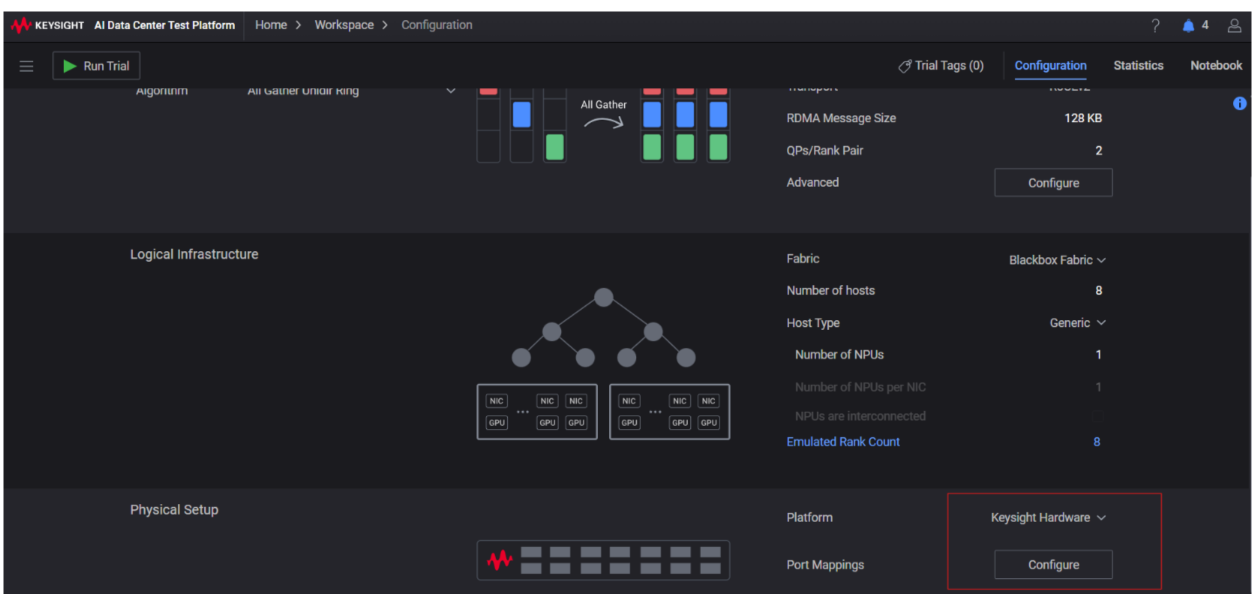
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing.

Setup

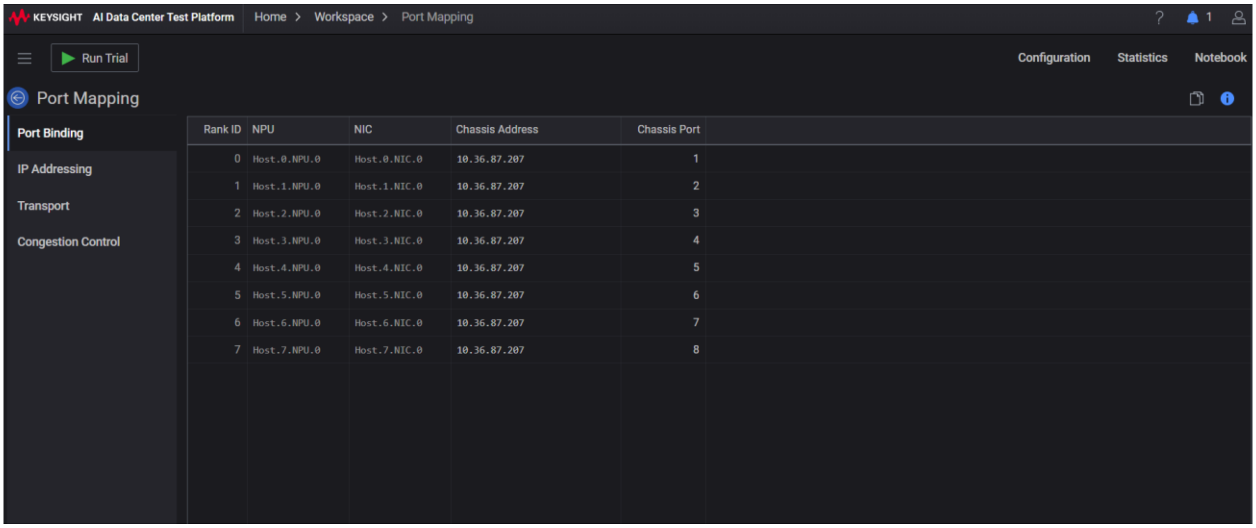


Step-by-step instructions

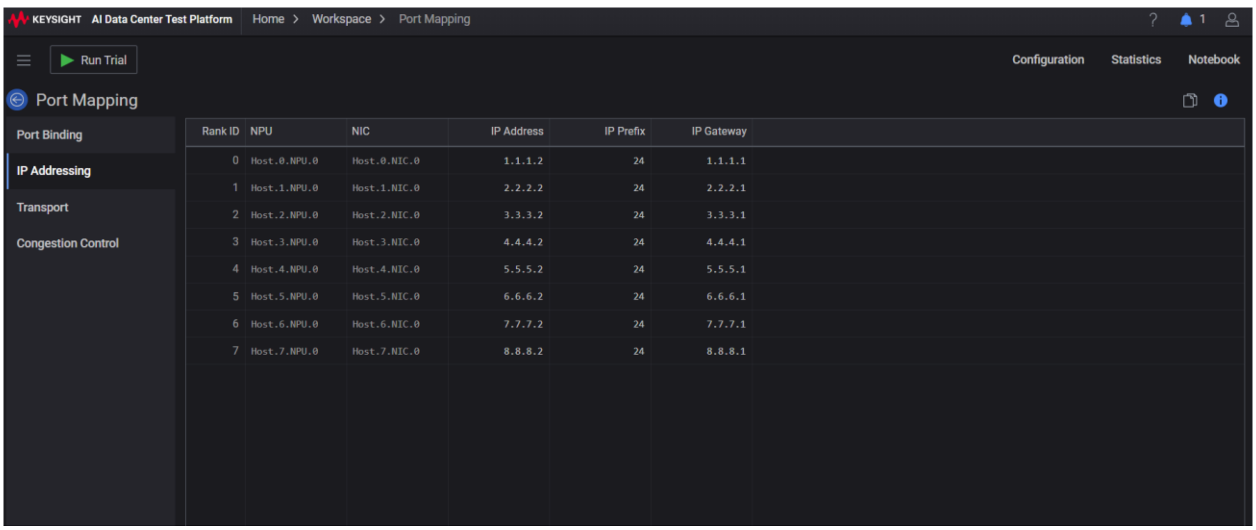
1. Under **Platform**, select **Keysight Hardware.** Select **Configure**.



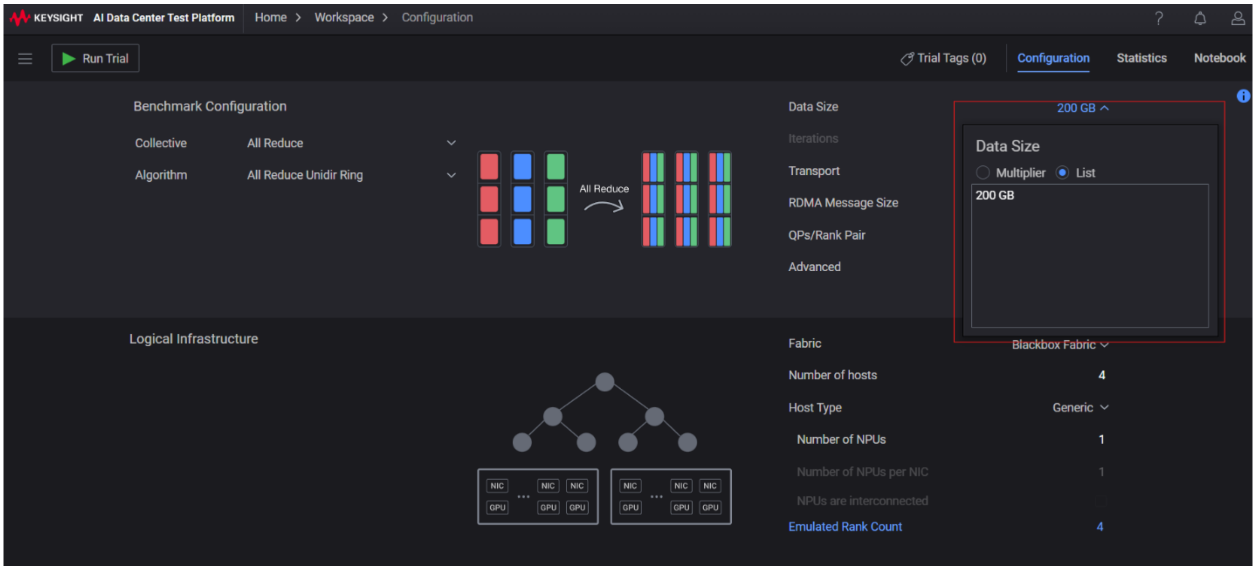
2. Set the port binding.



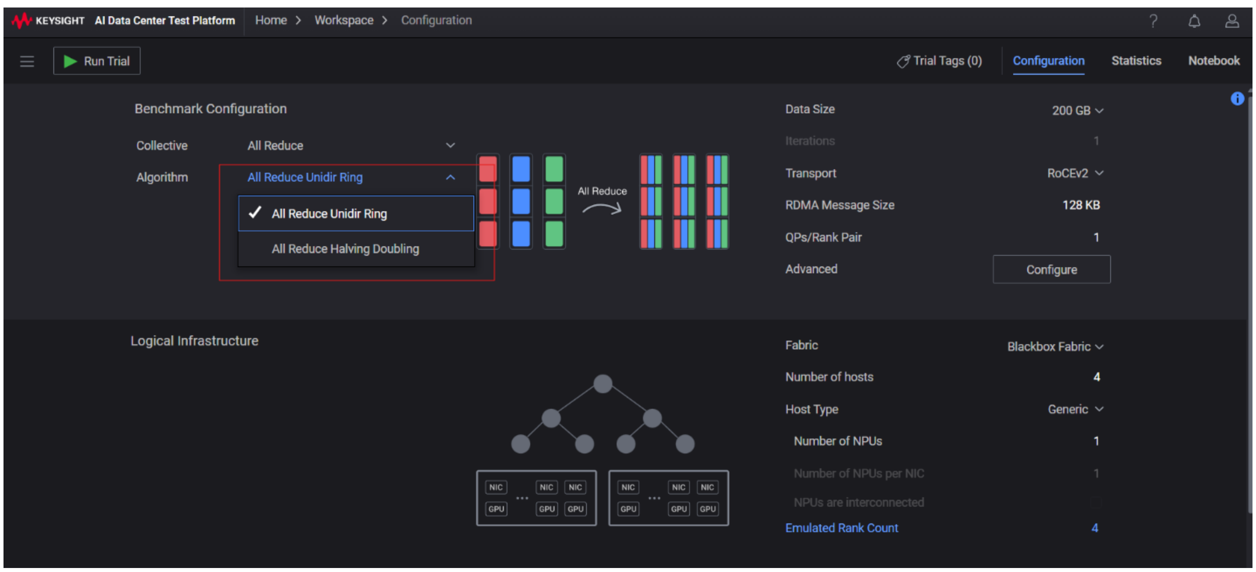
3. Set the IP address and gateway.



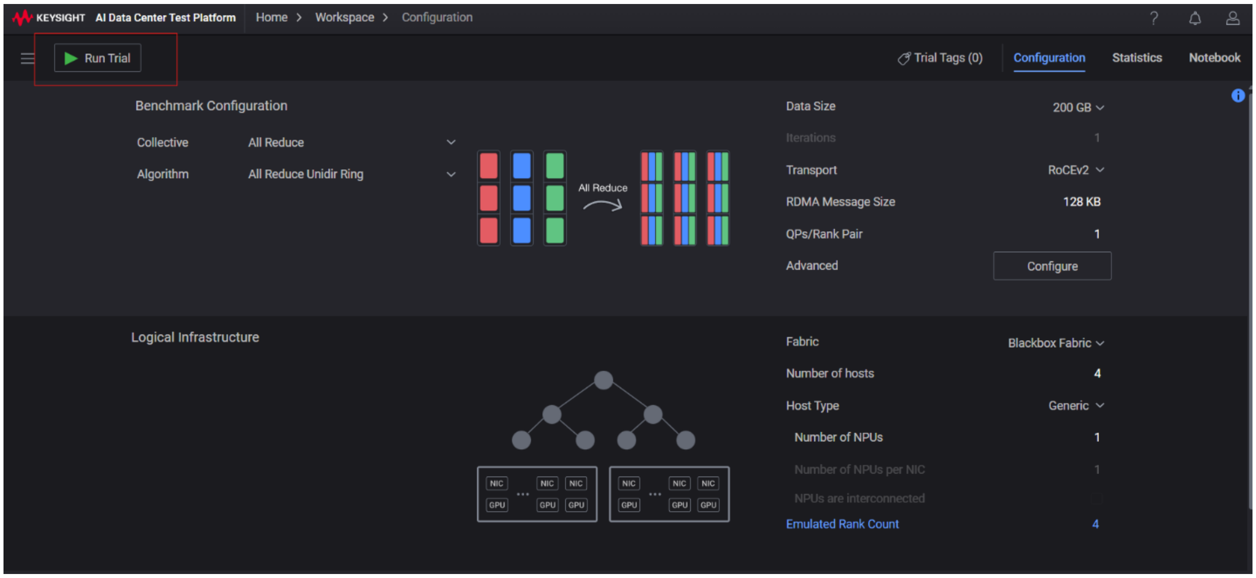
4. Set the data size.



5. Set the collective and algorithms.



6. Select **Run Trial**.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an 'active' MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an 'active MTU.'

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other collective to test: All-to-all, All-Gather-Ring.

- Test iteration

  Iteration reflects the batch in the context of AI training.

  A batch refers to a fixed-size group of input data that is used to train or test an AI model. Collectives for each batch keep the same data size.

- Test trial

  Collective bursts in fabric.

  Within a single batch, one or more than one collective may burst. Test trial may reflect the sequential collectives in the batch. The data size of collectives can vary depending on the specific AI model layer size or gradient size.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

Results analysis

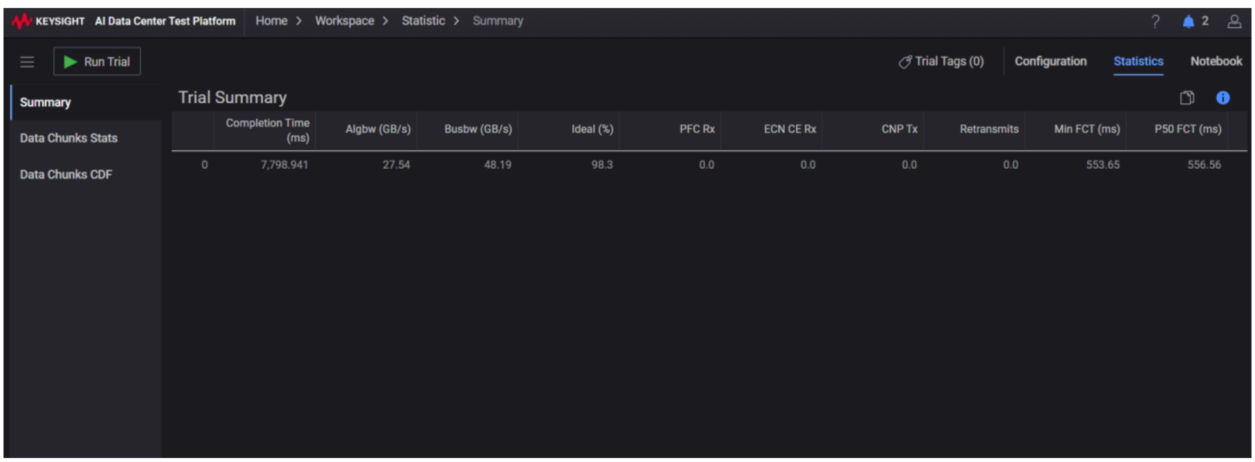
From the test results, we observed that the completion time is measured in milliseconds and algorithm bandwidth is calculated accordingly. The ideal percentage against bus bandwidth is high, indicating no congestion and high throughput during forwarding. PFC, ECN-CE, and CNP numbers support the congestion statistics.

- Algorithm bandwidth

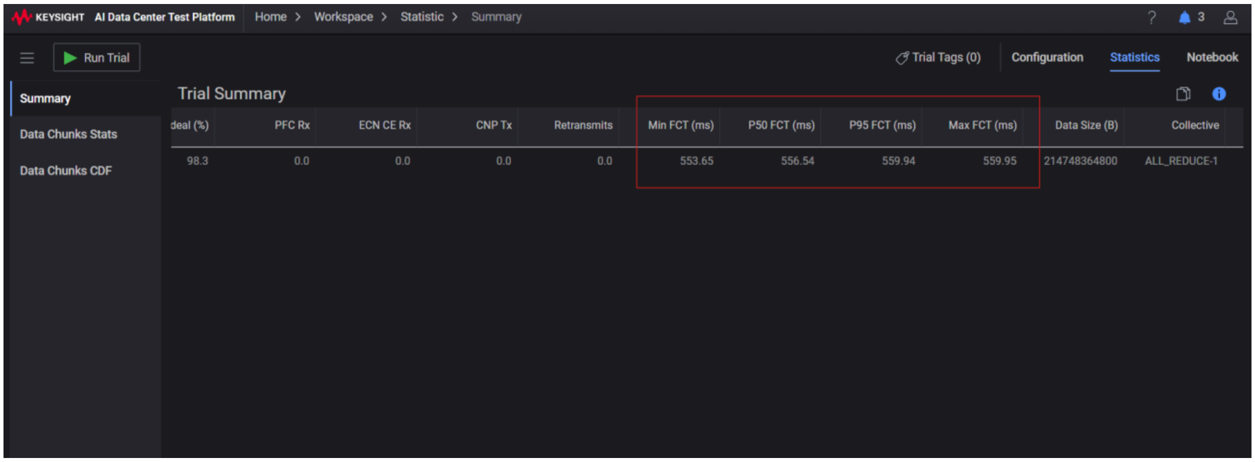
  Algorithm bandwidth, calculated as Collective Size / Time
- Bus bandwidth

  Bus bandwidth, calculated as Algorithm Bandwidth multiplied by a collective-dependent compensation factor
- Ideal percentage

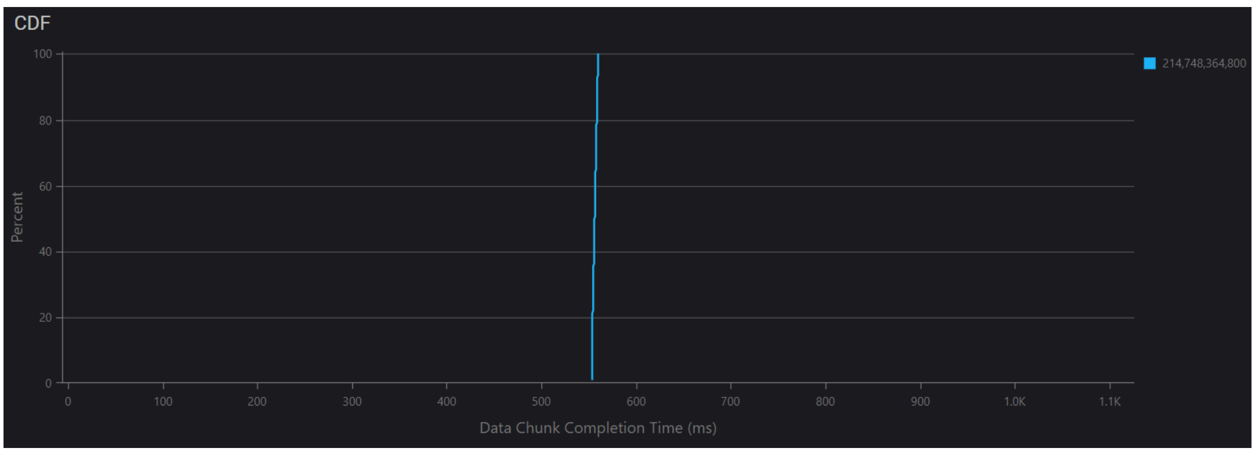
  Bus bandwidth divided by ideal goodput (as percentage)



Long-tail latency is summarized by providing metrics including minimum (Min), maximum (Max), and percentiles (P50 and P95) for the completion time of QPs.



Obtain any required percentile values directly from the cumulative distribution function (CDF) result.

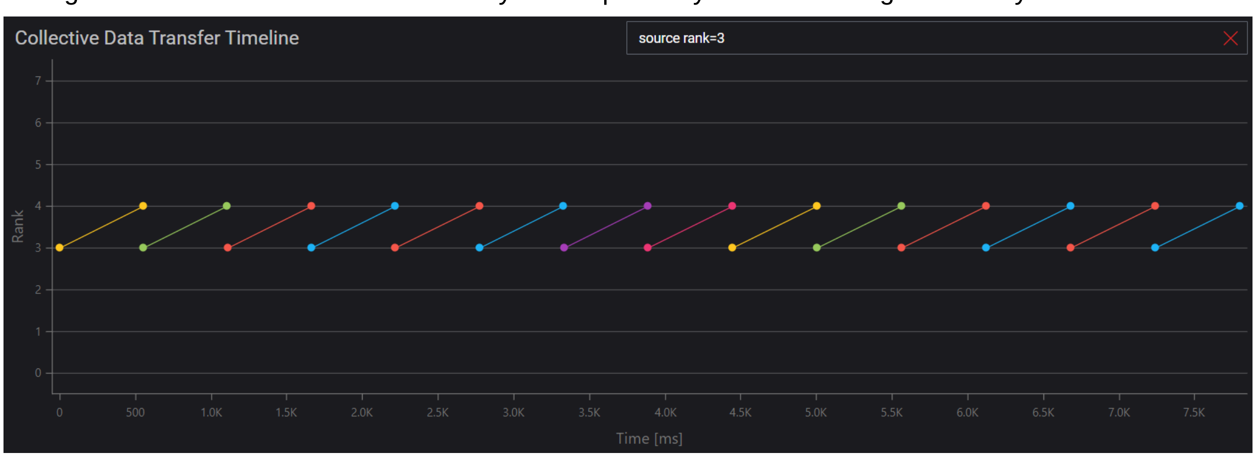


Conclusion

We observe the egress QPs for one of the ToR switches.

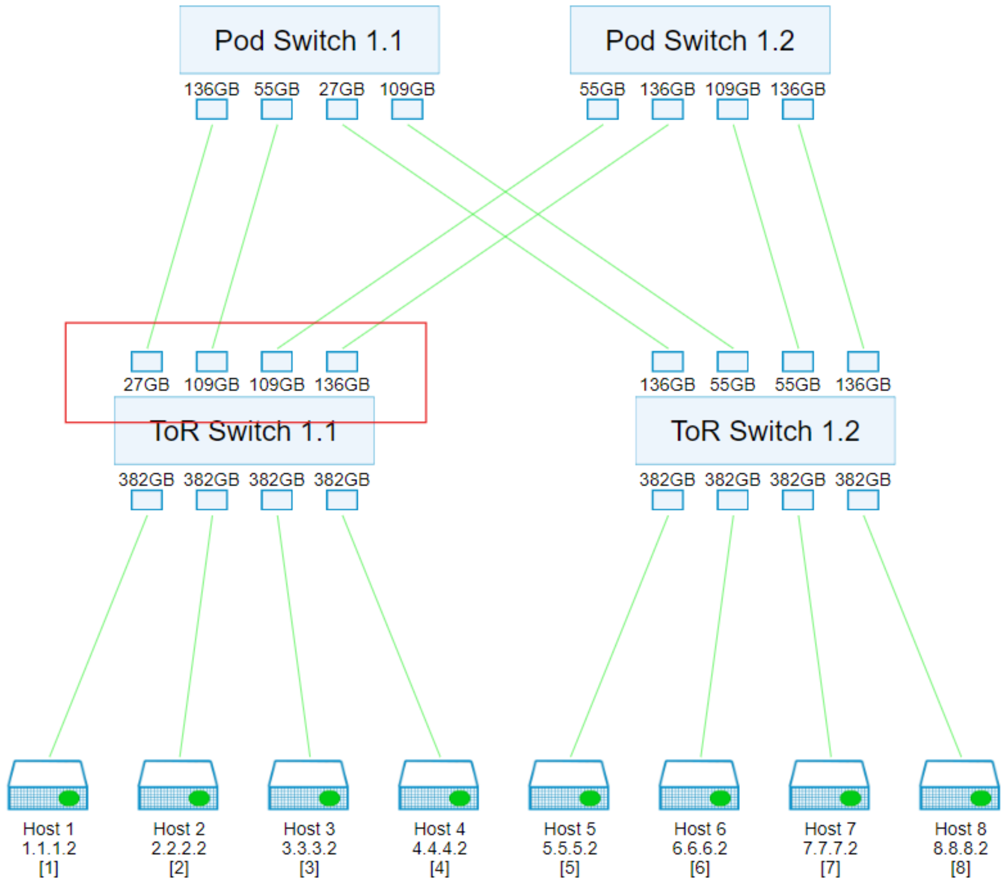
- Egress QPs

  Congestion is not a concern because they are sequentially issued and originate solely from Host 3.



- Egress Tx count (bytes)

  For a super-elephant flow, switching can result in an unbalanced hash distribution.



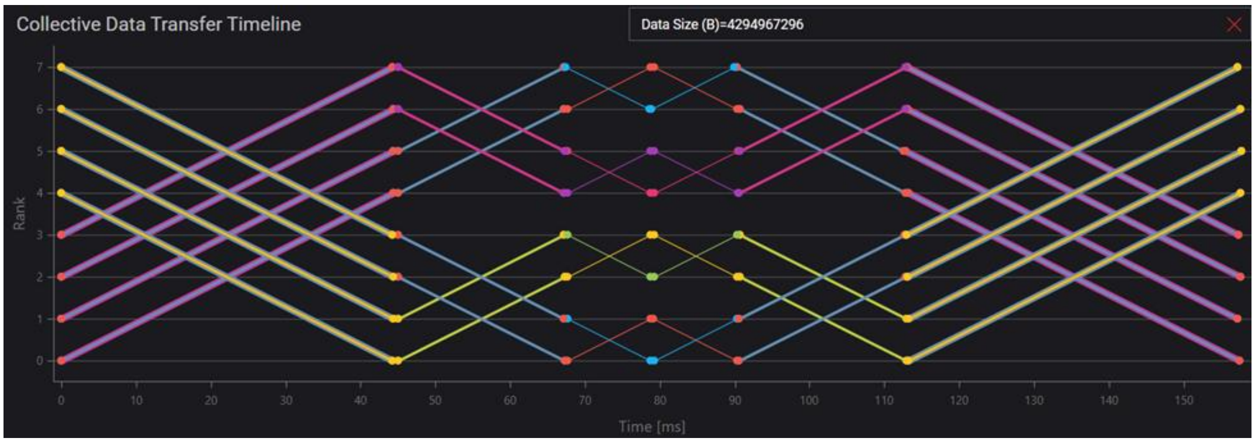
Test case 4: Halving Doubling algorithm JCT

Overview

This test case employs an all-reduce-halving doubling collective operation to evaluate the data forwarding performance of the fabric by using a combination of halving and doubling to emulate the transmission of data chunks through the ToR and pod switches. The SUT consists of the ToR switches, which are interconnected with multiple pod switches forming redundant paths through ECMP.

- *About Halving Doubling algorithm*

  The Halving Doubling Algorithm (HDA) is a popular method for estimating the size of a collection or dataset, particularly in the context of parallel computing and distributed systems. It is an iterative process that uses a combination of halving and doubling to narrow down the search space.



To facilitate this all-reduce operation, the HDA is employed to efficiently exchange data between nodes. Here is how it works:

- Initial data size: The algorithm starts with the initial data size (let us call it \`N\`).
- Distance 4: Halve the data size (\`N / 2\`) and send this reduced size to nodes at distance 4 (that is, four hops away). This is being done to reduce the amount of data transmitted.
- Distance 2: Send one-fourth of the original data size (\`N / 4\`) to nodes at distance 2 (that is, two hops away).
- Distance 1: Send one-eighth of the original data size (\`N / 8\`) to nodes at distance 1 (that is, directly adjacent).

By using this HDA, each node only needs to transmit a fraction of its original data, reducing communication overhead.

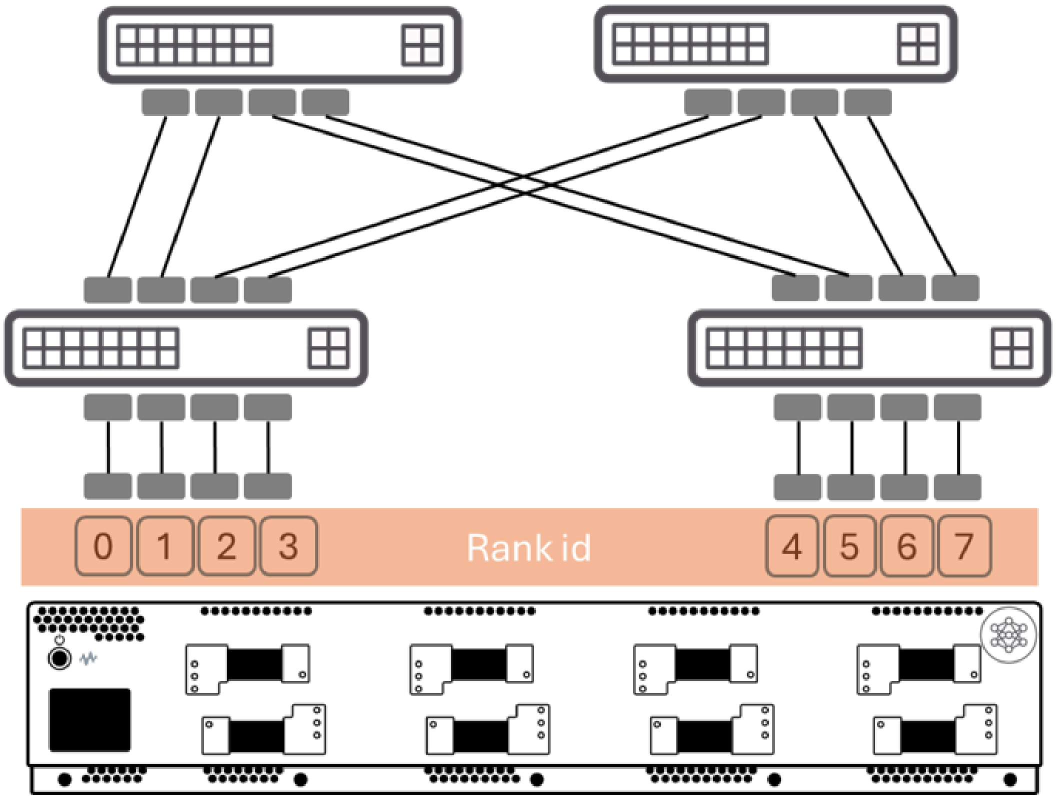
Doubling phase:

After all the partial gradients have been exchanged and accumulated, the algorithm doubles the data size at each hop (from distance 1 to 4) to ensure that the result is correctly combined. This process ensures that the overall gradient is correctly computed across all nodes.

Objectives

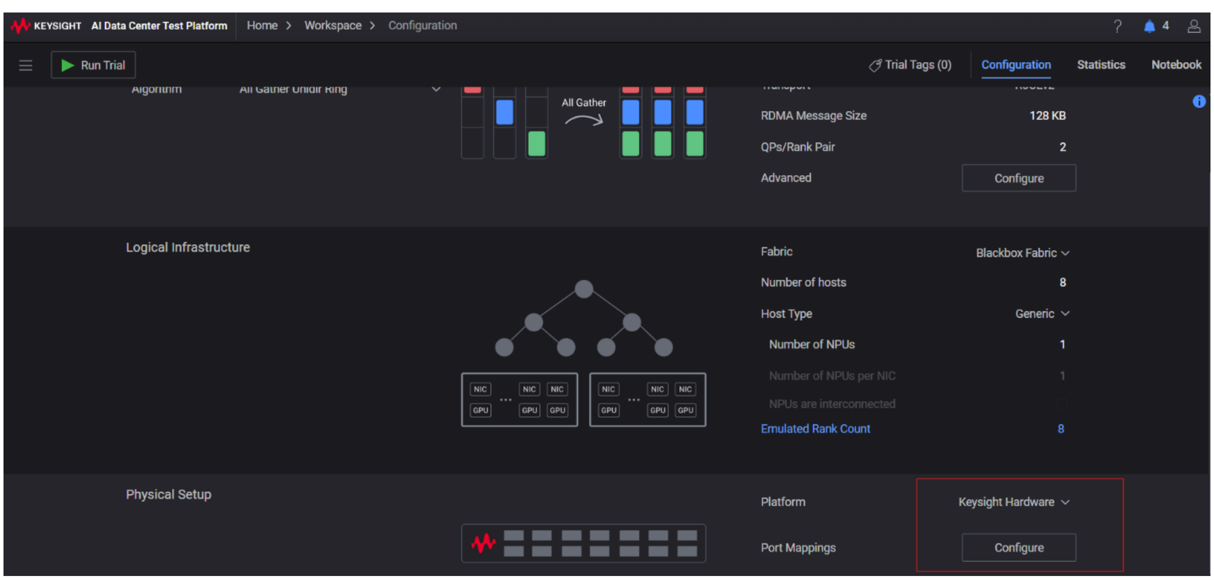
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing.

Setup

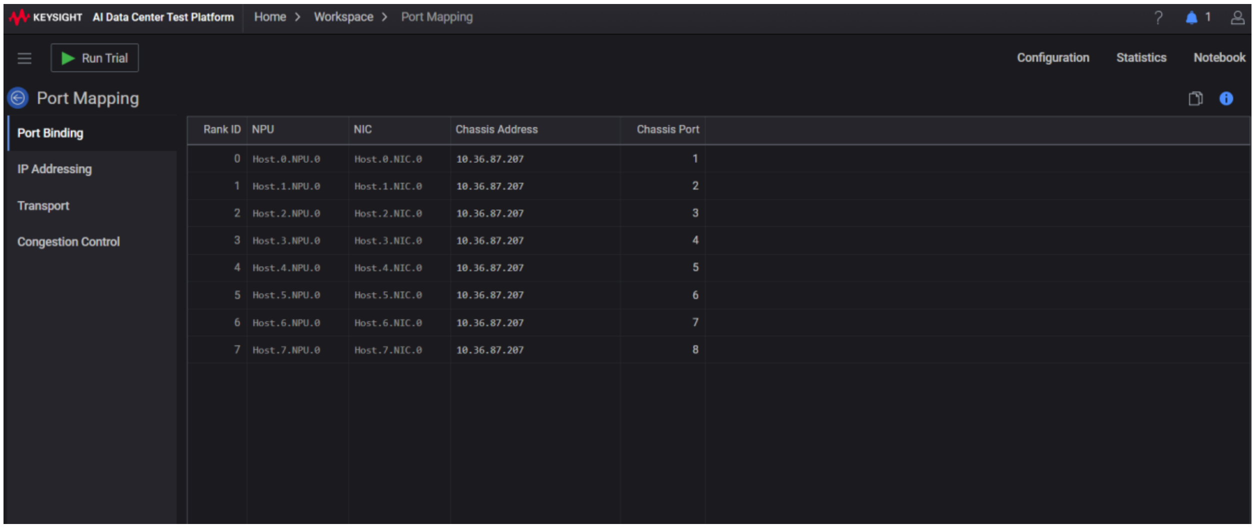


Step-by-step instructions

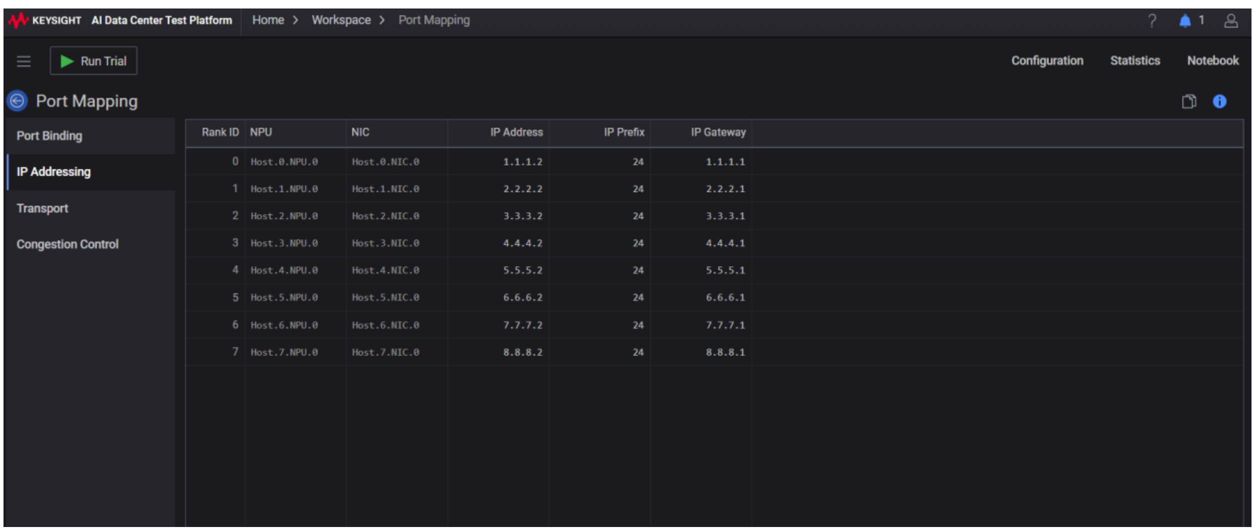
1. Under **Platform**, select **Keysight Hardware.** Select **Configure**.



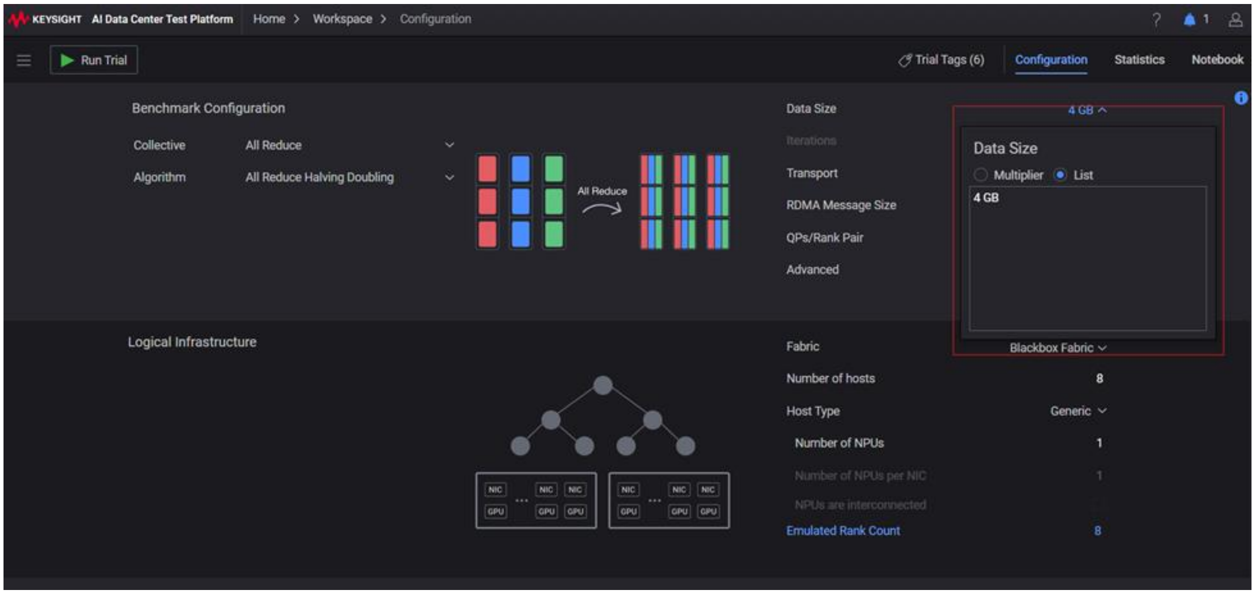
2. Set the port binding.



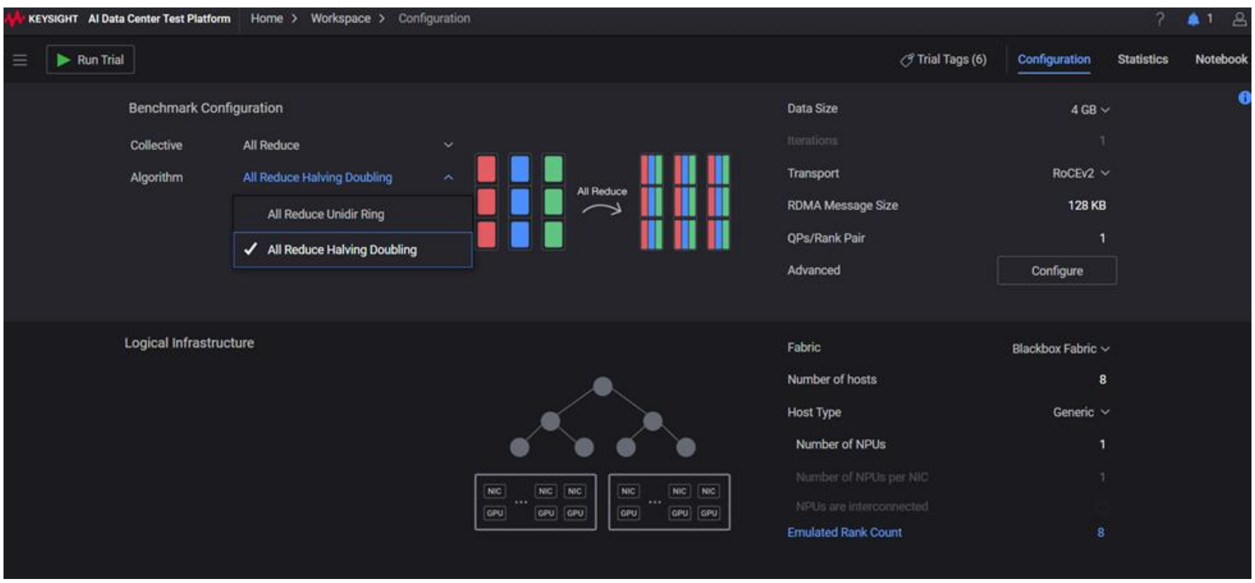
3. Set the IP address and gateway.



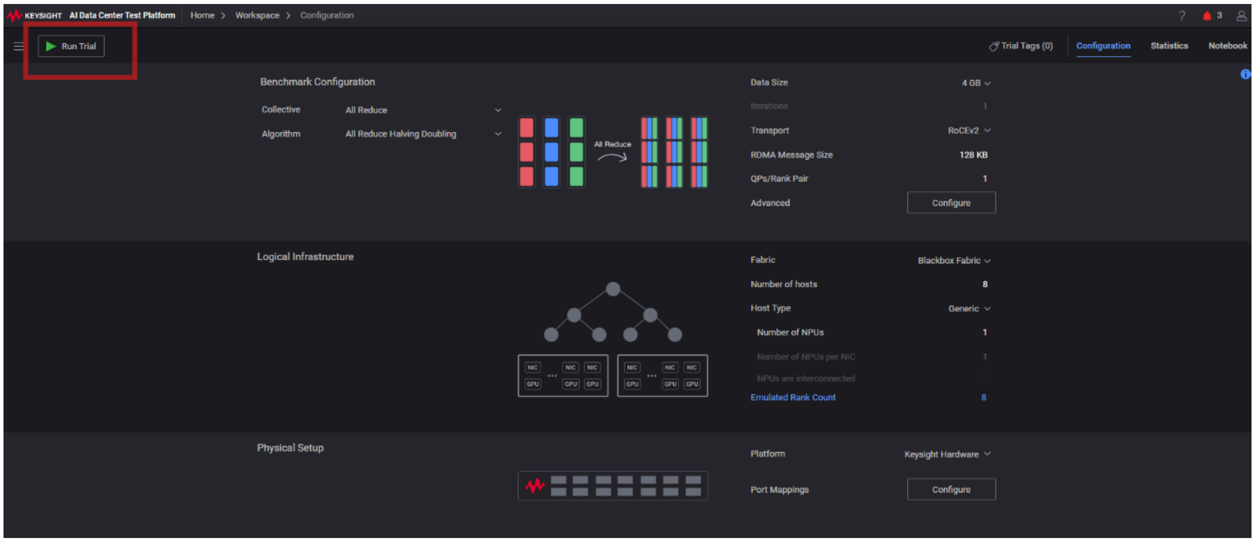
4. Set the data size.



5. Set the collective and algorithms.



6. Select **Run Trial**.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an 'active' MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an 'active MTU.'

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Test iteration

  Iteration reflects the batch in the context of AI training.

  A batch refers to a fixed-size group of input data that is used to train or test an AI model. Collectives for each batch keep the same data size.

- Test trial

  Collective bursts in fabric.

  Within a single batch, one or more than one collective may burst. Test trial may reflect the sequential collectives in the batch. The data size of collectives can vary depending on the specific AI model layer size or gradient size.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

Results analysis

From the test results, we observed that the completion time is measured in milliseconds and algorithm bandwidth is calculated accordingly. The ideal percentage against bus bandwidth is high, indicating no congestion and high throughput during forwarding. PFC, ECN-CE, and CNP numbers support the congestion statistics.

- Algorithm bandwidth

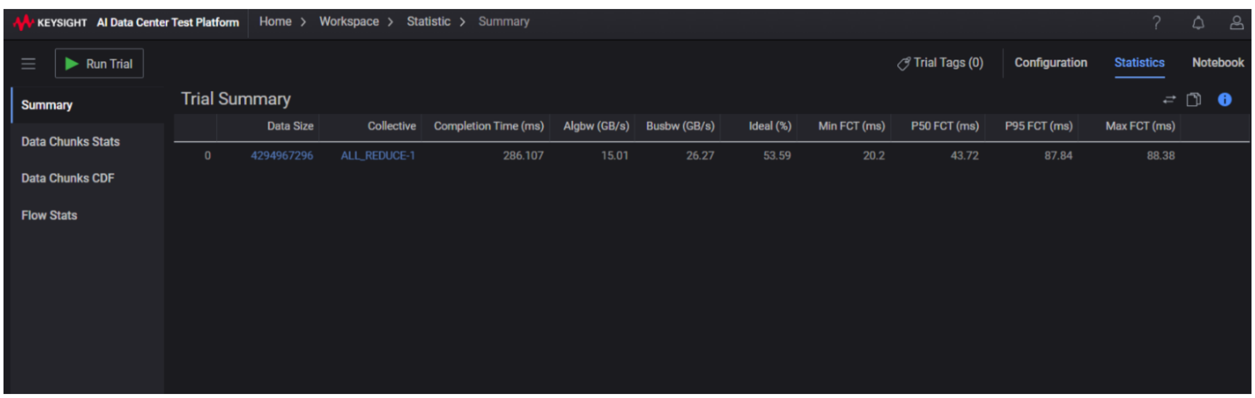
  Algorithm bandwidth, calculated as Collective Size / Time

- Bus bandwidth

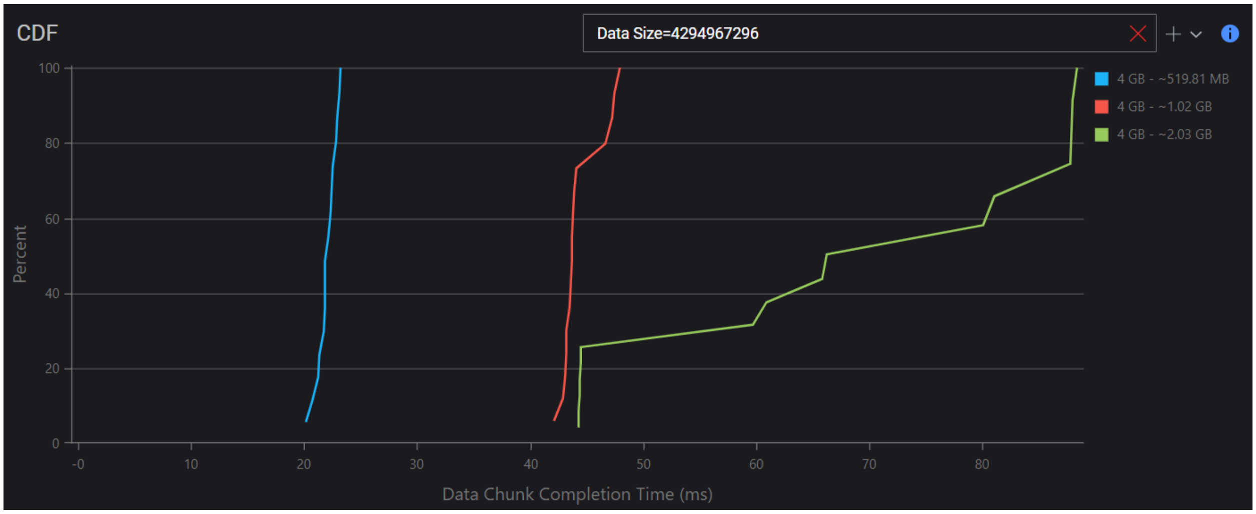
  Bus bandwidth, calculated as Algorithm Bandwidth multiplied by a collective-dependent compensation factor

- Ideal percentage

  Bus bandwidth divided by ideal goodput (as percentage)



- Long tail latency distribution for data sizes



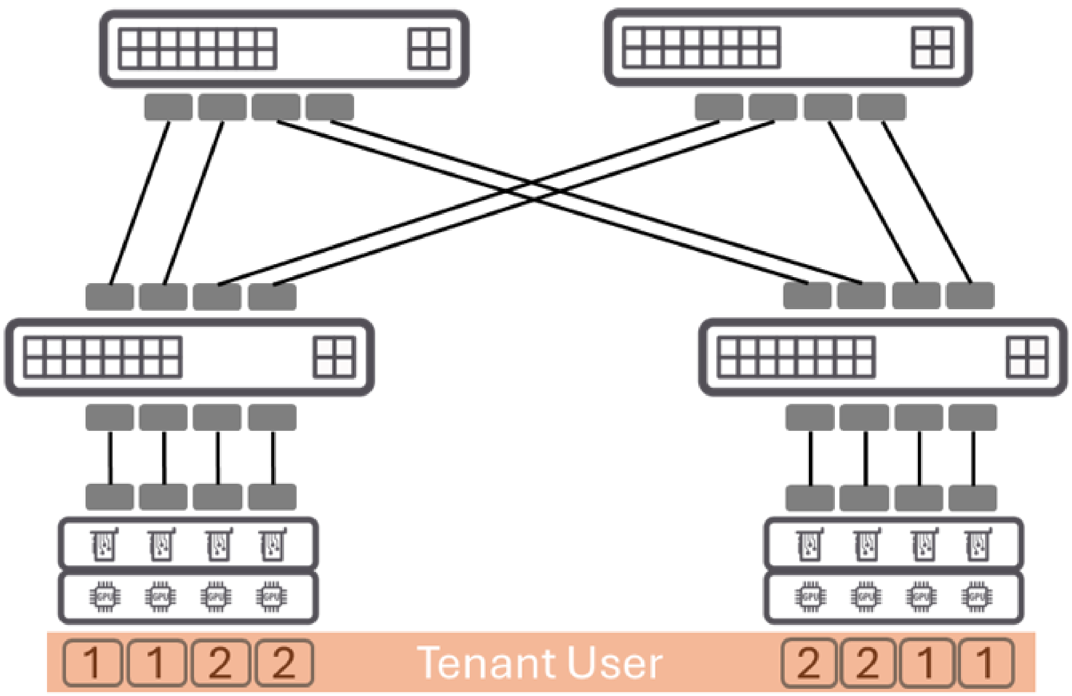
Conclusion

The results show a variant CDF of completion time for all-reduce HD exchanges with different data sizes, highlighting subtle differences in performance between algorithms by using the same dataset.

Test Methodology for Performance Isolation

AI cloud infrastructure must support multiple users (tenants) and parallel applications or workflows, which generates traffic that competes for the infrastructure's shared resources — such as the network — and can impact performance.

In this experiment, we emulate a scenario featuring four NPU hosts, each equipped with two NPU and NICs. We employ two jobs running simultaneously on these hosts. Under these conditions, the network fabric may experience congestion, enabling us to validate the performance under certain resource competing scenarios.



Perform the following test procedure:

1. Set the peering test ports with IP and gateway.
2. Split the ports into two groups, one for background noise, the other for under test.
3. Set the rank id for Tenant User1, Tenant User2.
4. Set the data size.
5. Set the algorithm.
6. Run background trials Job1 from Tenant User 1 first.
7. Run trials Job2 from Tenant User 2 and observe the performance result.

The following test cases demonstrate the test methodologies and detailed step-by-step configuration with automated test packages available in CB:

Test collectives	Data size	Write msg size	Parallel QPs	Congestion control
All-reduce Ring / All-to-all noise	4 GB / 4 GB	128 KB	1	PFC+ECN
All-reduce Ring / All-reduce Ring noise	4 GB / 4 GB	128 KB	1	PFC+ECN
2*All-reduce Ring / 2*All-to-all	512 MB-64 GB / 64 GB	128 KB	1	PFC+ECN

Test case 1: All-reduce-ring collective JCT with all-to-all neighbor

Overview

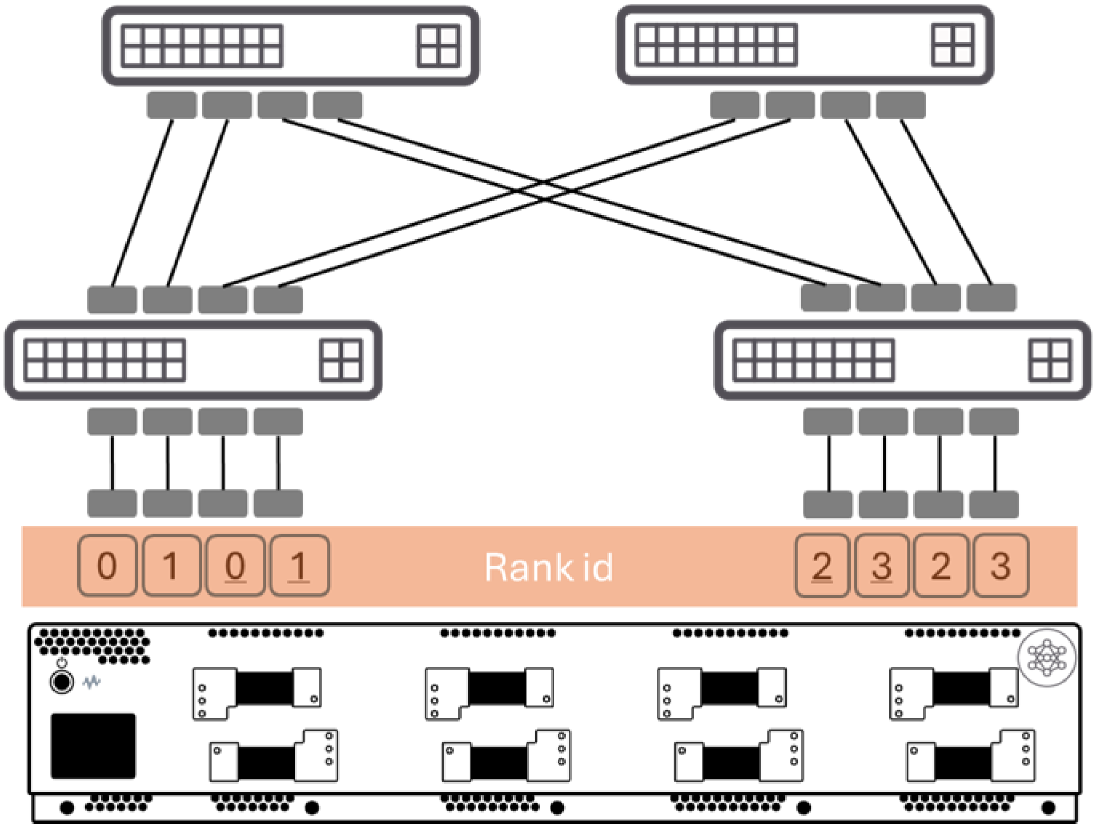
This comprehensive test case leverages a continuous all-to-all collective as a background operation and an all-reduce-ring collective to rigorously evaluate the data forwarding performance of the fabric by using a synthetic behavior with realistic loading patterns to emulate the transmission of data chunks through the ToR and pod switches. The SUT comprises the ToR switches, which are interconnected with multiple pod switches, forming redundant paths through ECMP.

Objectives

1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing.

Setup

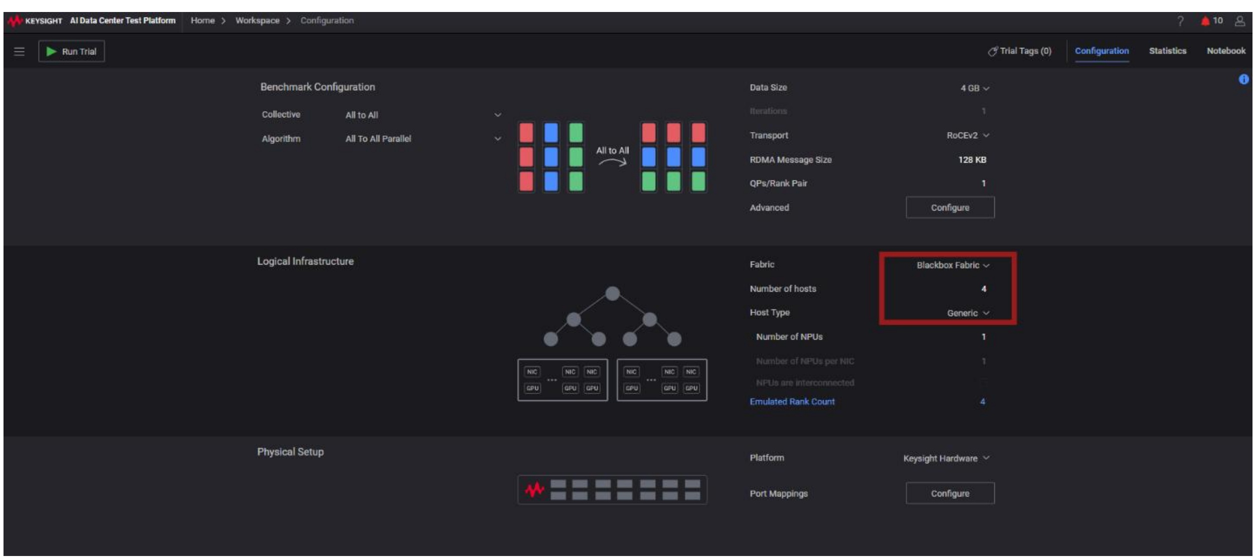
In this setup, two concurrent collectives are run within the fabric. The first collective, [0, 1, 2, 3], employs an all-reduce-ring algorithm, while the second collective, [0, 1, 2, 3], runs all-to-all concurrently. On the DUT, PFC is enabled to ensure the control of congestion.



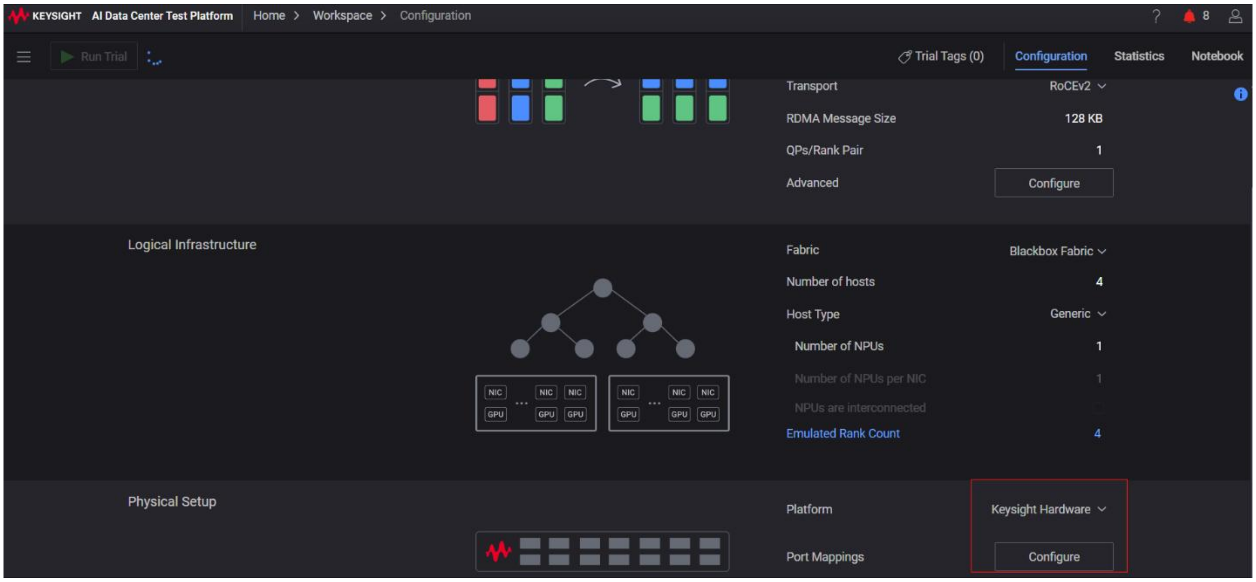
Step-by-step instructions

Background collective

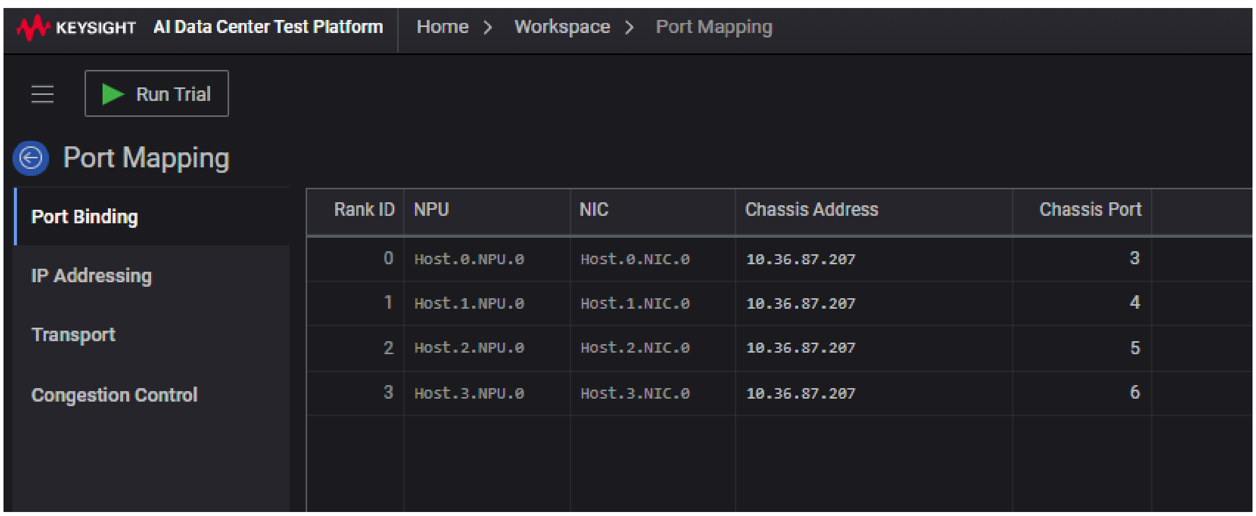
1. Set the number of hosts.



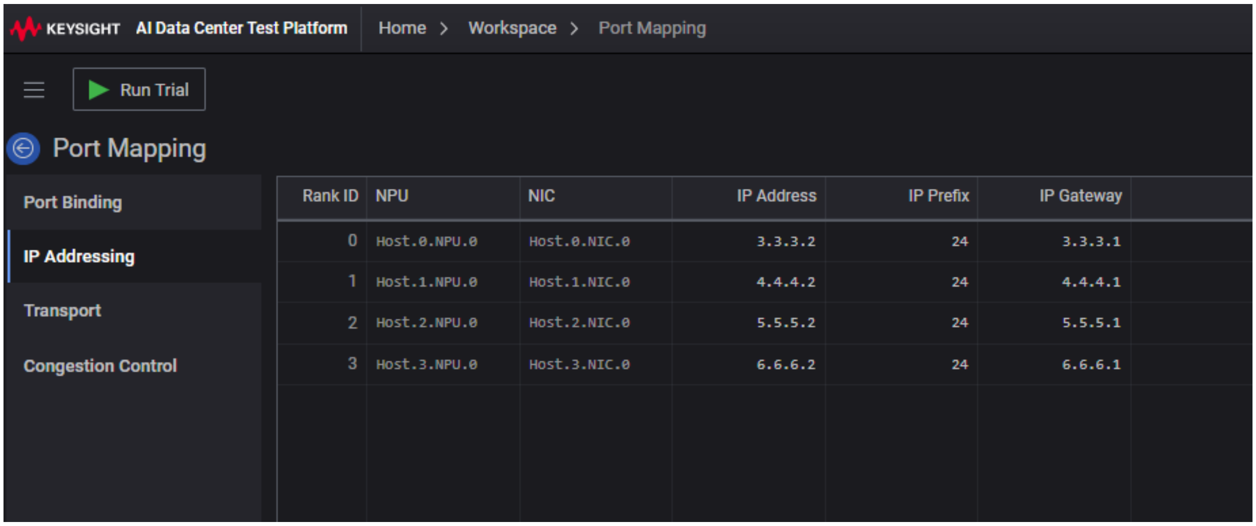
2. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



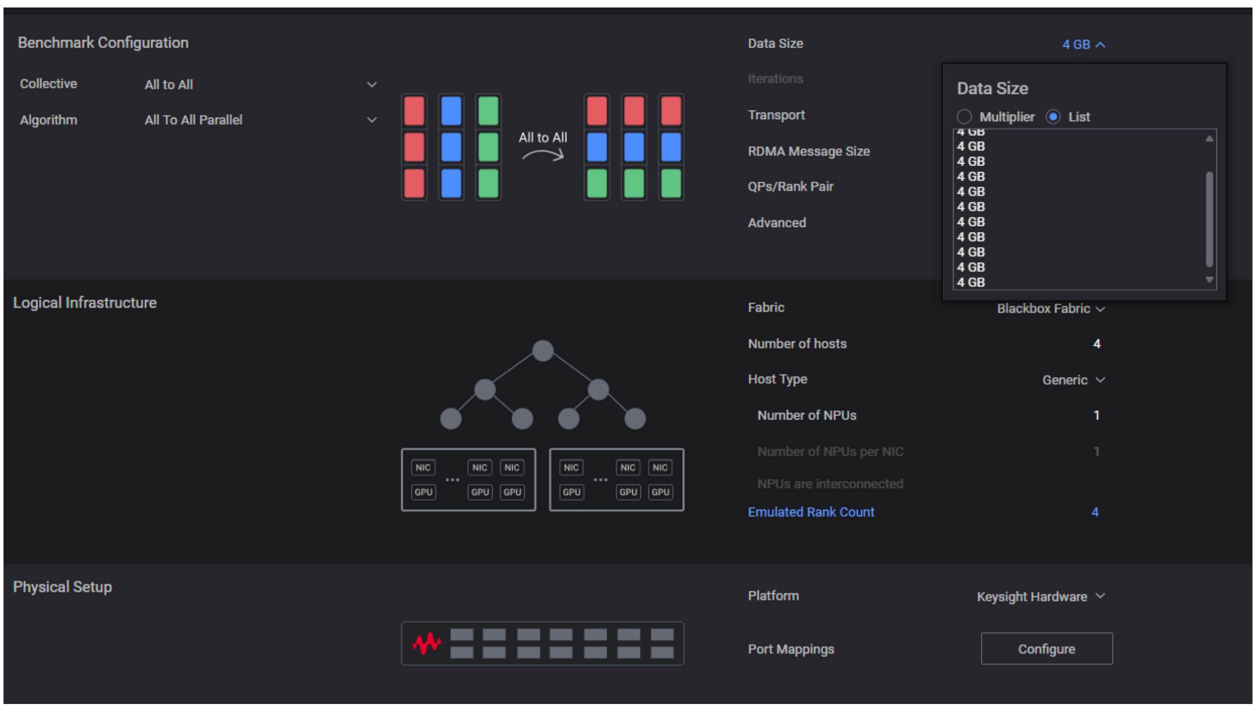
3. Set the port binding.



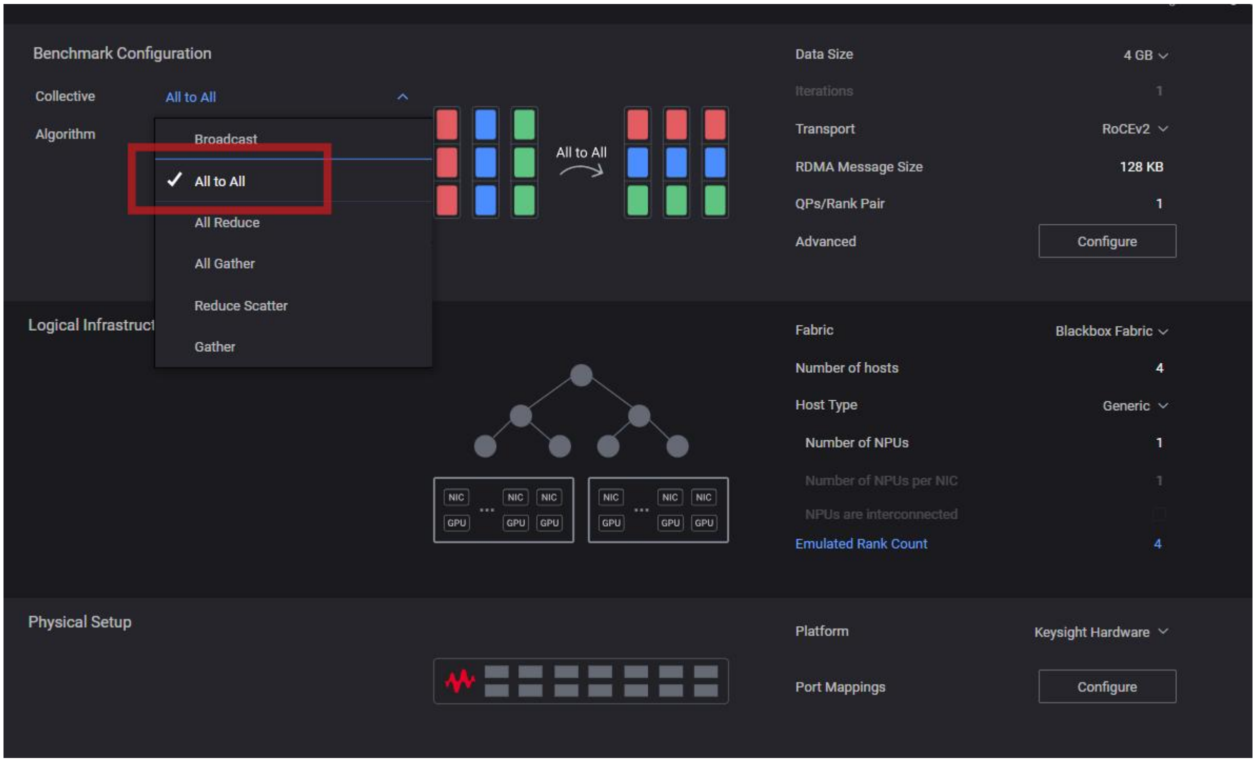
4. Set the IP address and gateway.



5. Set the data size, make background data size a fixed or random list.

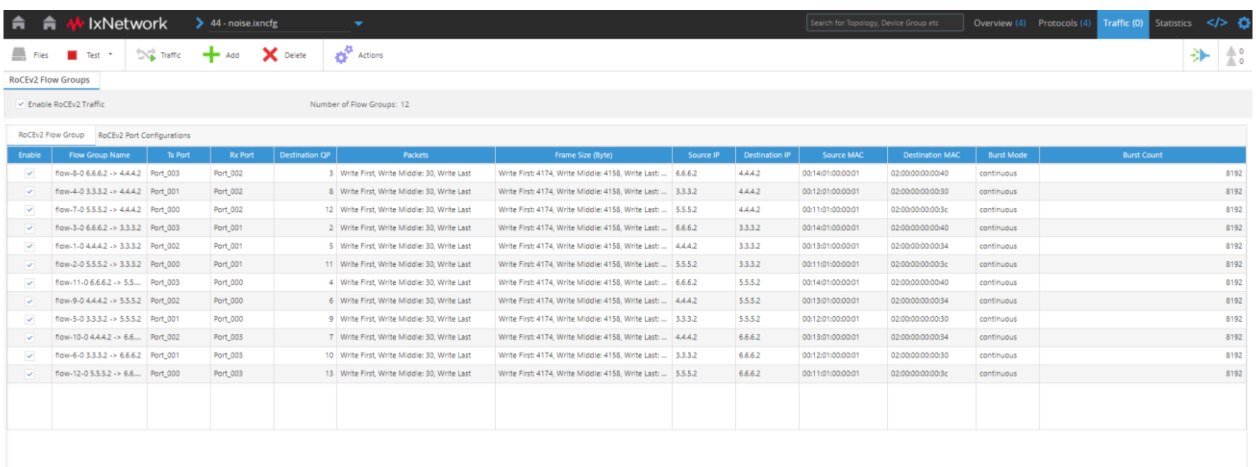


6. Set the collective and algorithms.

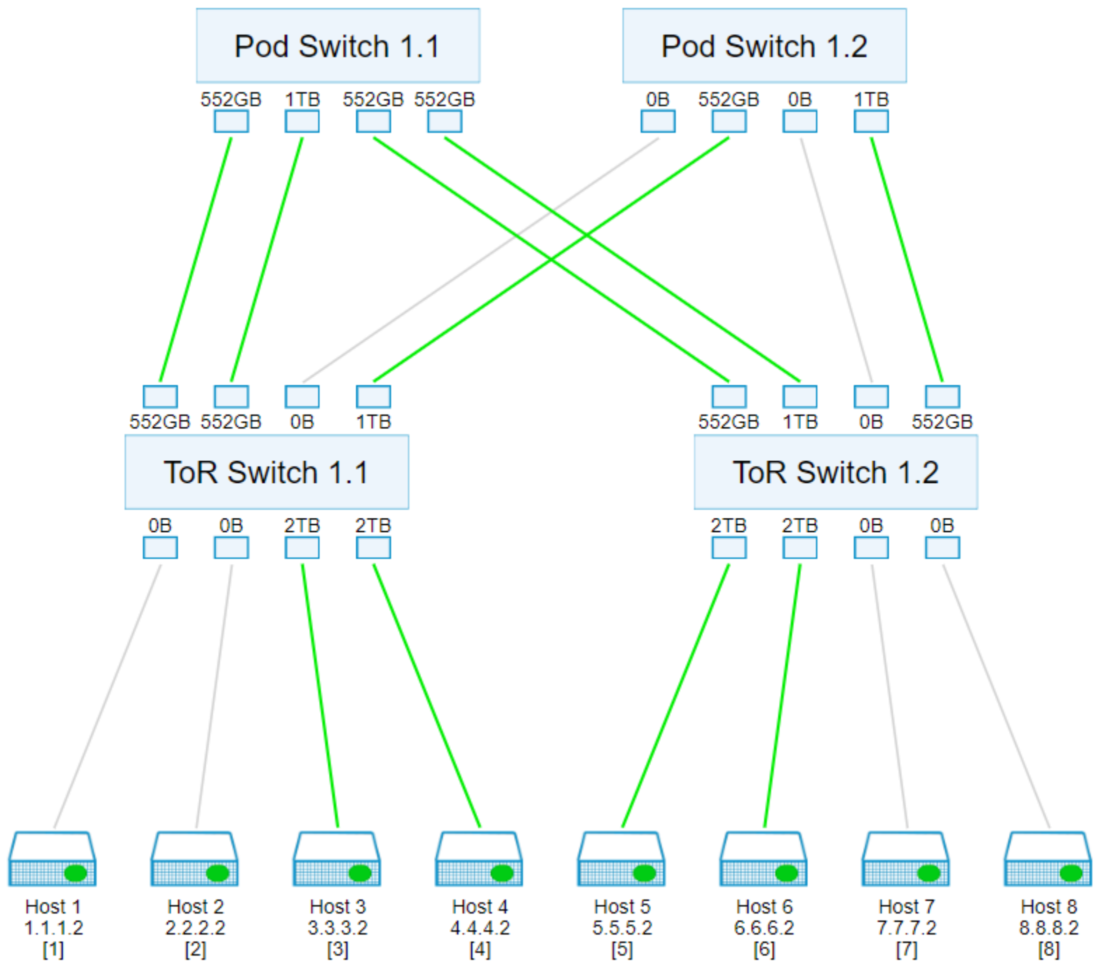


7. Select **Run Trial** to run background collectives first.

8. Alternatively, run with **IxNetwork**. Select **Start Traffic**.

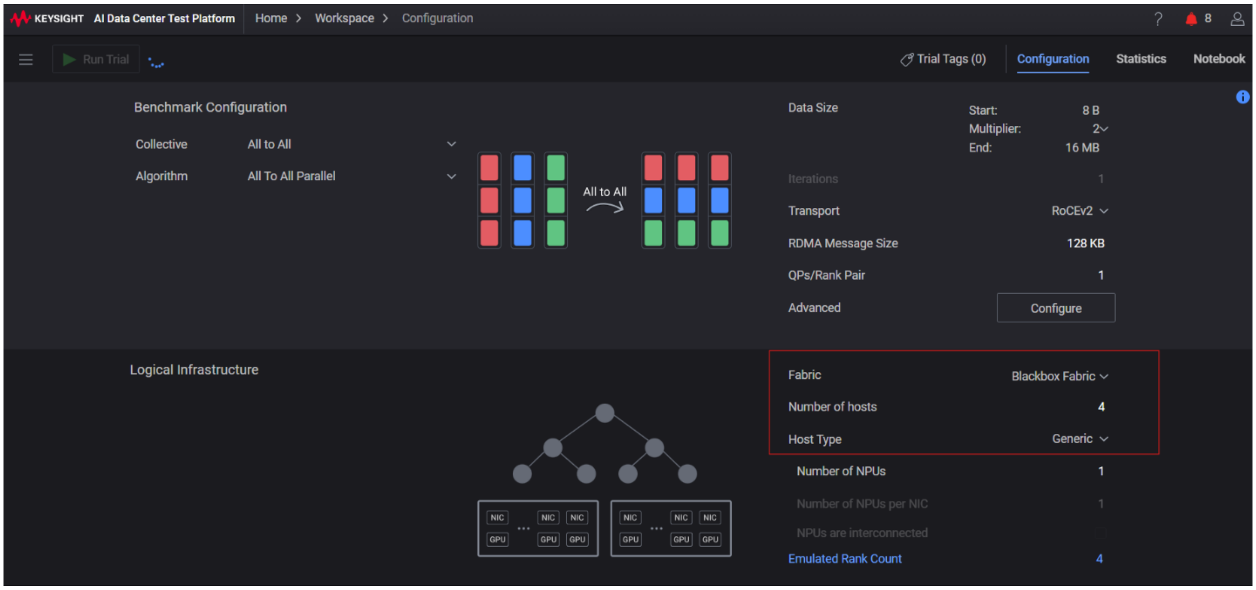


9. Make sure collective queue pairs are transmitting in the fabric.

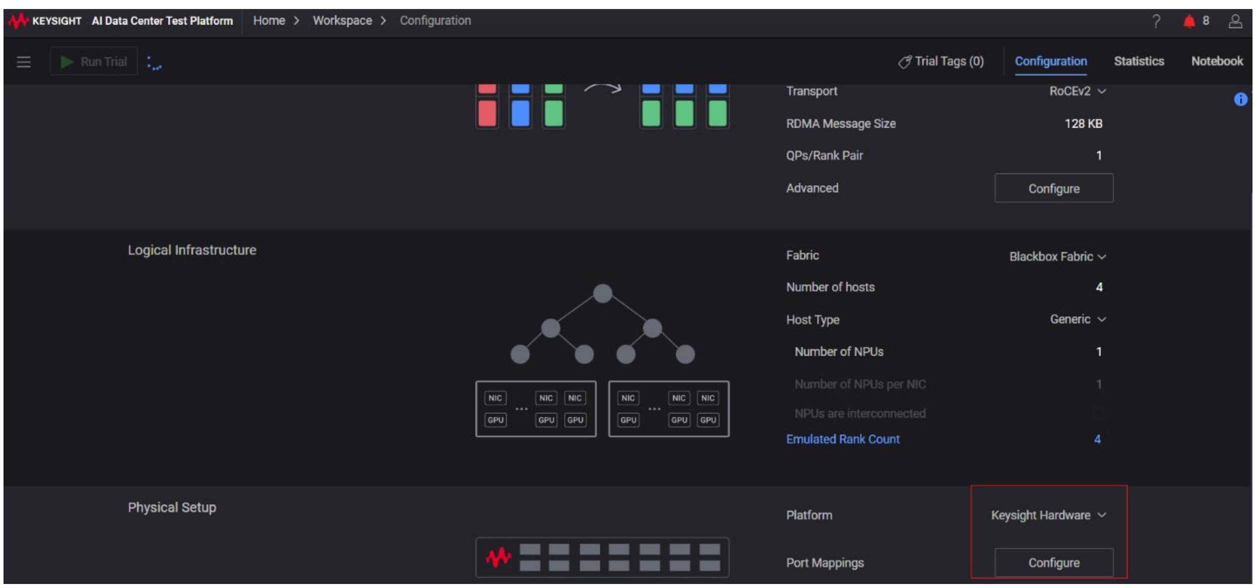


Collective under test

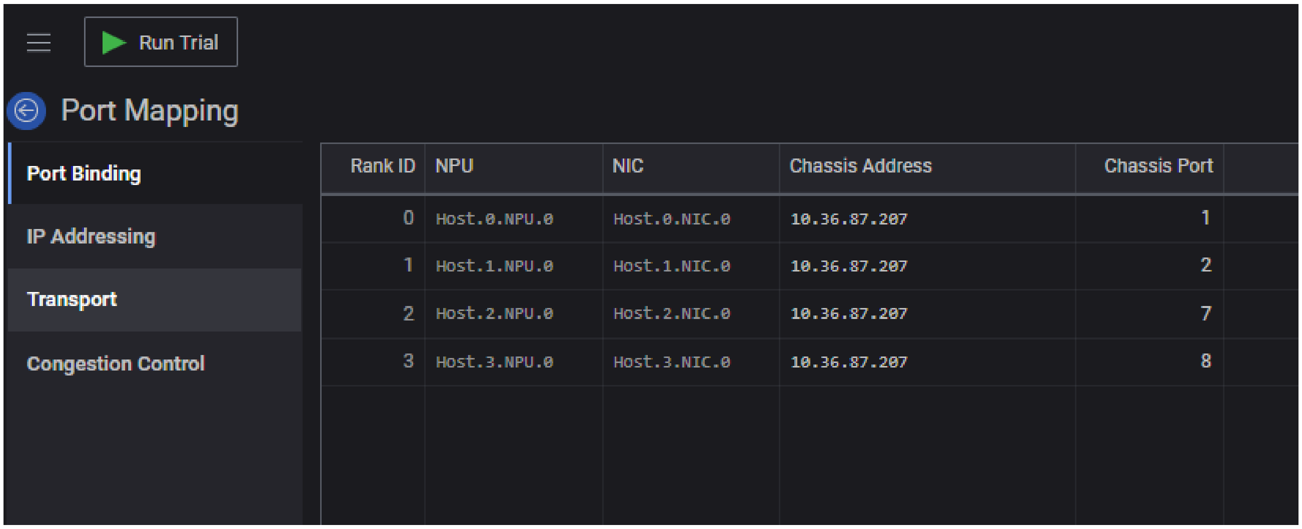
1. Set the number of hosts.



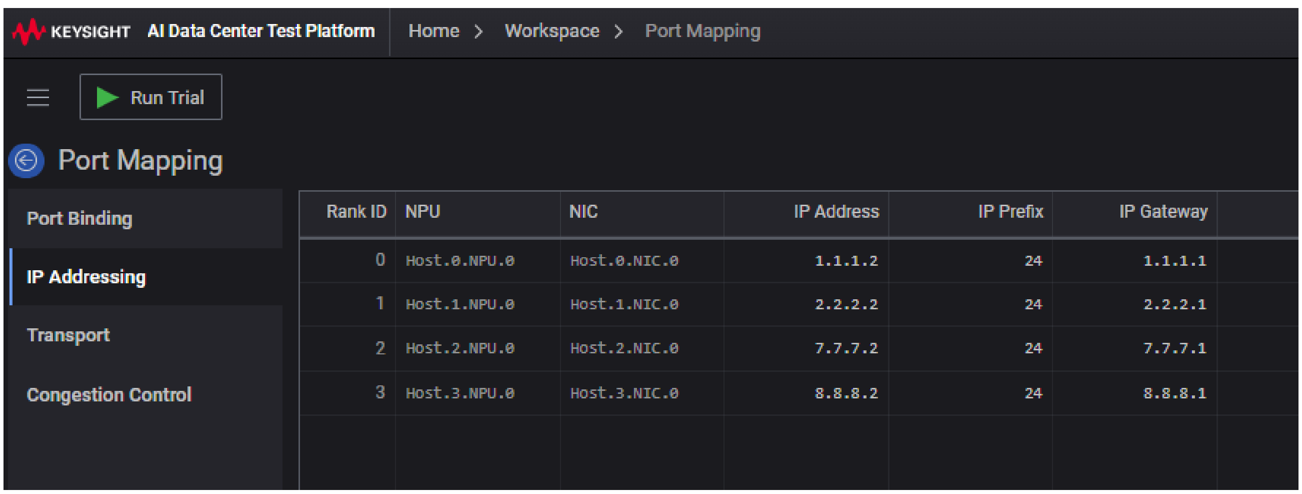
2. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



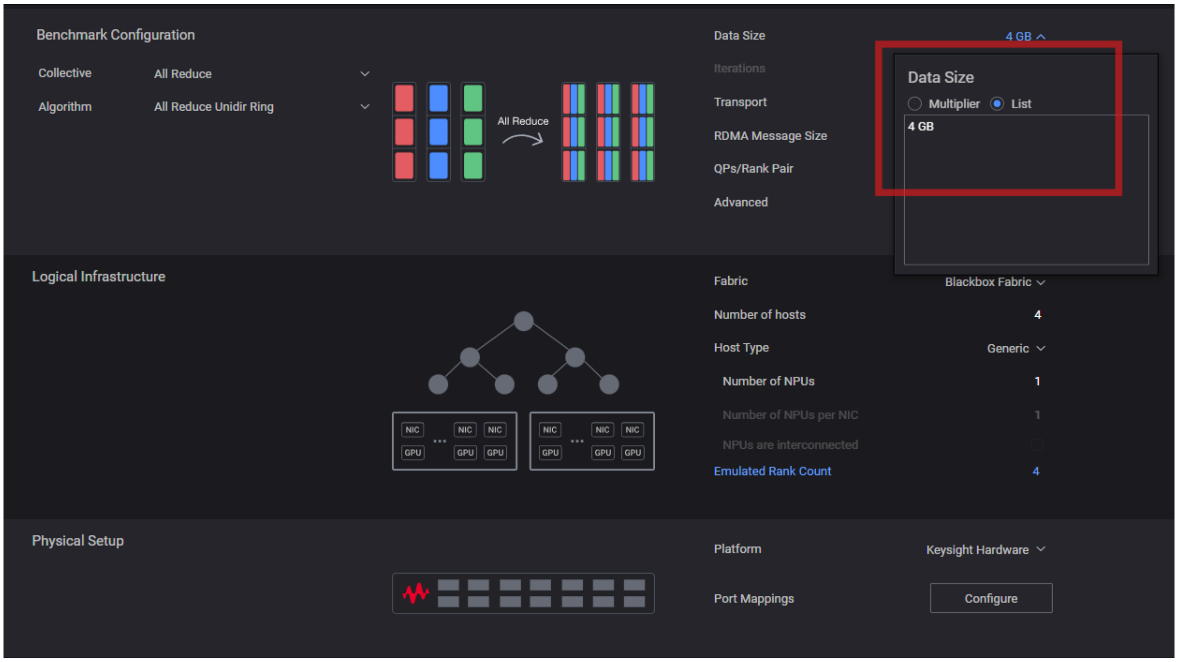
3. Set the port binding.



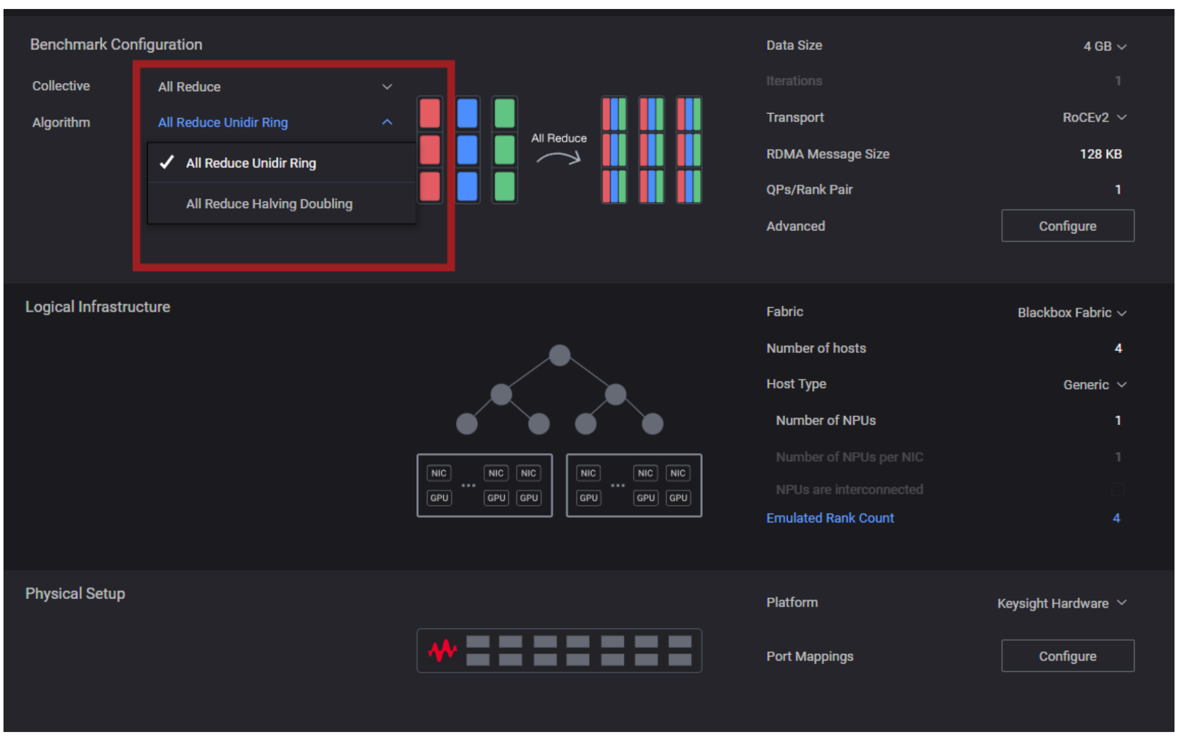
4. Set the IP address and gateway.



5. Set the data size.



6. Set the collective and algorithms.



7. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other collective to test: All-to-all, All-Gather-Ring.

- Test iteration

  Iteration reflects the batch in the context of AI training.

  A batch refers to a fixed-size group of input data that is used to train or test an AI model. Collectives for each batch keep the same data size.

- Test trial

  Collective bursts in fabric.

  Within a single batch, one or more than one collective may burst. Test trial may reflect the sequential collectives in the batch. The data size of collectives can vary depending on the specific AI model layer size or gradient size.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

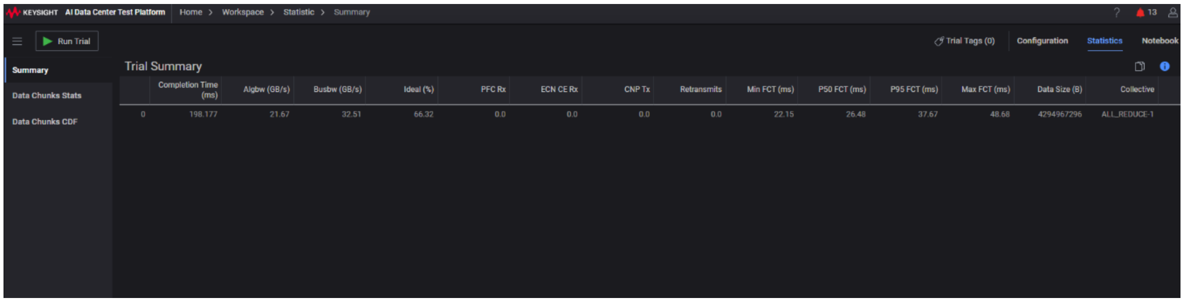
  Tensor parallel: Training data tensor size in bytes.

- Variant noises

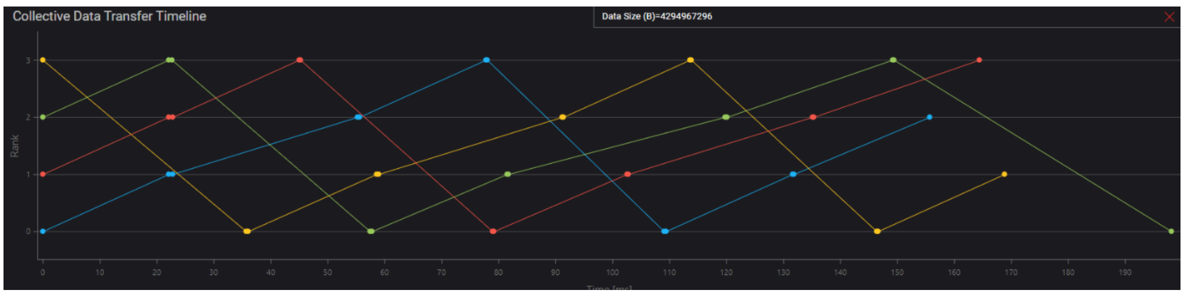
  Allocate multiple groups of noises of variant collectives or data size.

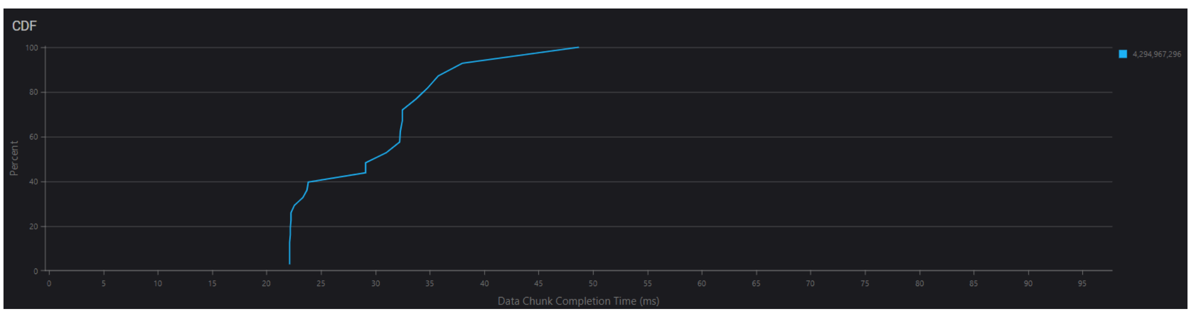
Results analysis

- The algorithm and bus bandwidth are not high.

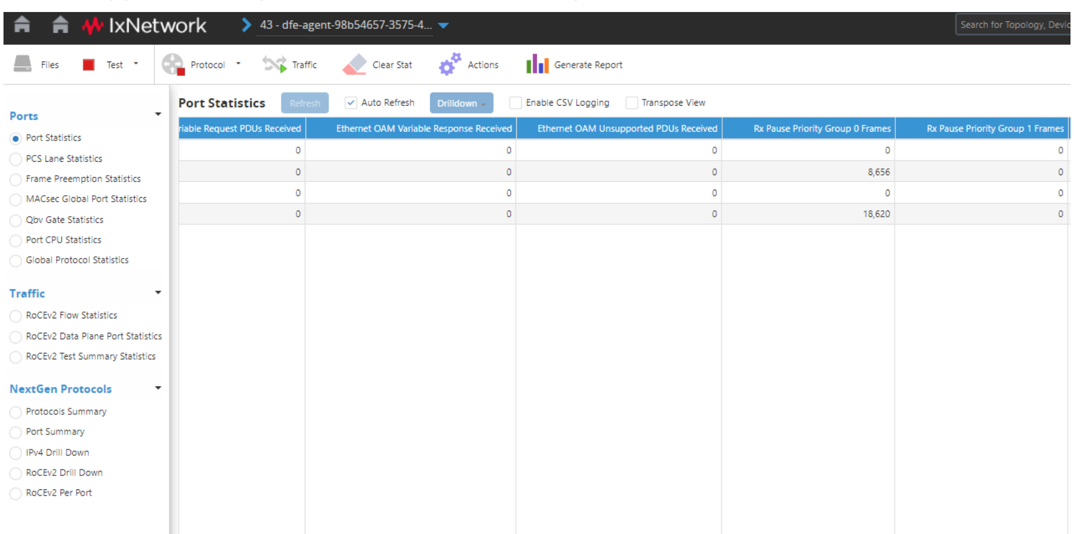


- Long-tail latency is observed.





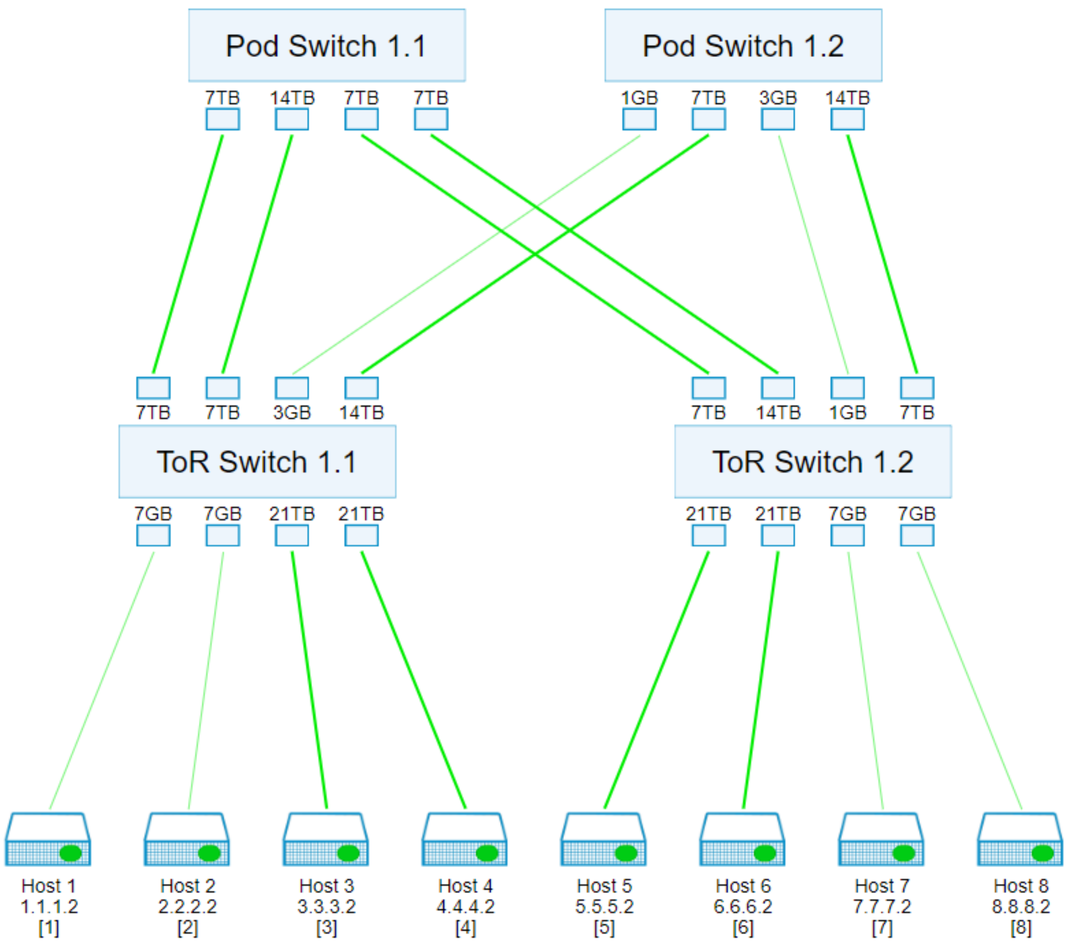
- PFC is triggered during tests with a sense of congestion.



Conclusion

With four ranks, an all-reduce-ring collective can achieve excellent results without any noisy neighbor. However, it can only achieve approximately 70 percent of its performance when using a single all-to-all collective to share the same network infrastructure, which is a more realistic scenario.

Tx count (bytes)



Test case 2: All-reduce-ring collectives JCT with all-reduce-ring neighbor

Overview

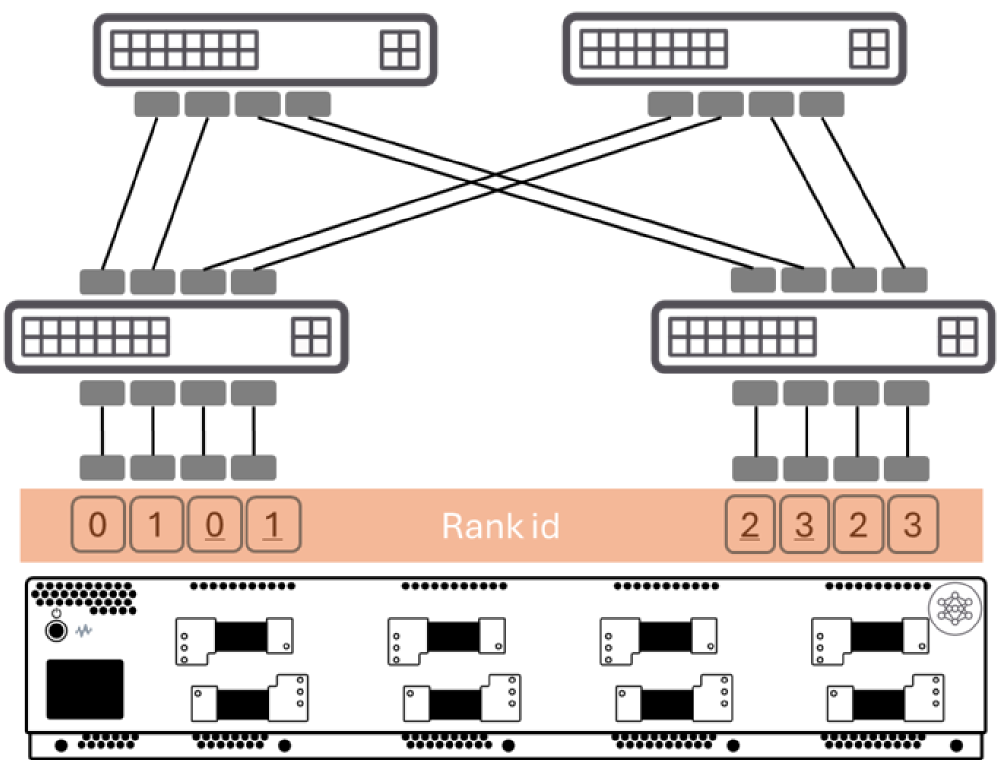
This comprehensive test case leverages a continuous all-reduce-ring collective as a background operation and an all-reduce-ring collective to rigorously evaluate the data forwarding performance of the fabric by using a synthetic behavior with realistic loading patterns to emulate the transmission of data chunks through the ToR and pod switches. The SUT comprises the ToR switches, which are interconnected with multiple pod switches forming redundant paths through ECMP.

Objectives

1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing.

Setup

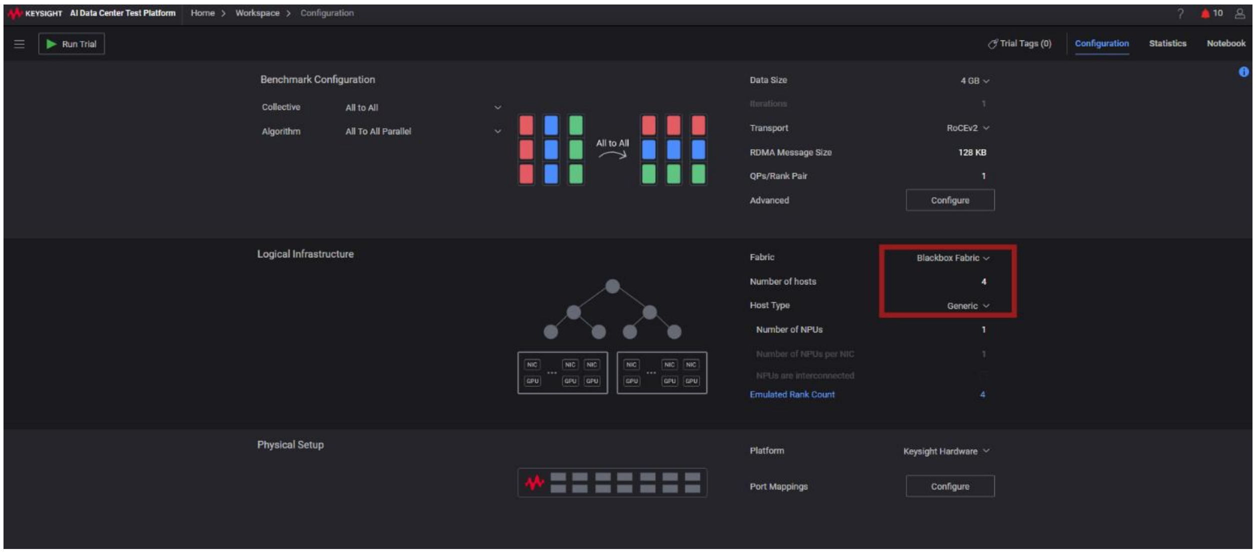
In this setup, two concurrent collectives are run within the fabric. The first collective, [0, 1, 2, 3], employs an all-reduce ring algorithm, while the second collective, [0, 1, 2, 3], runs all-reduce ring concurrently. On the DUT, PFC is enabled to ensure the control of congestion.



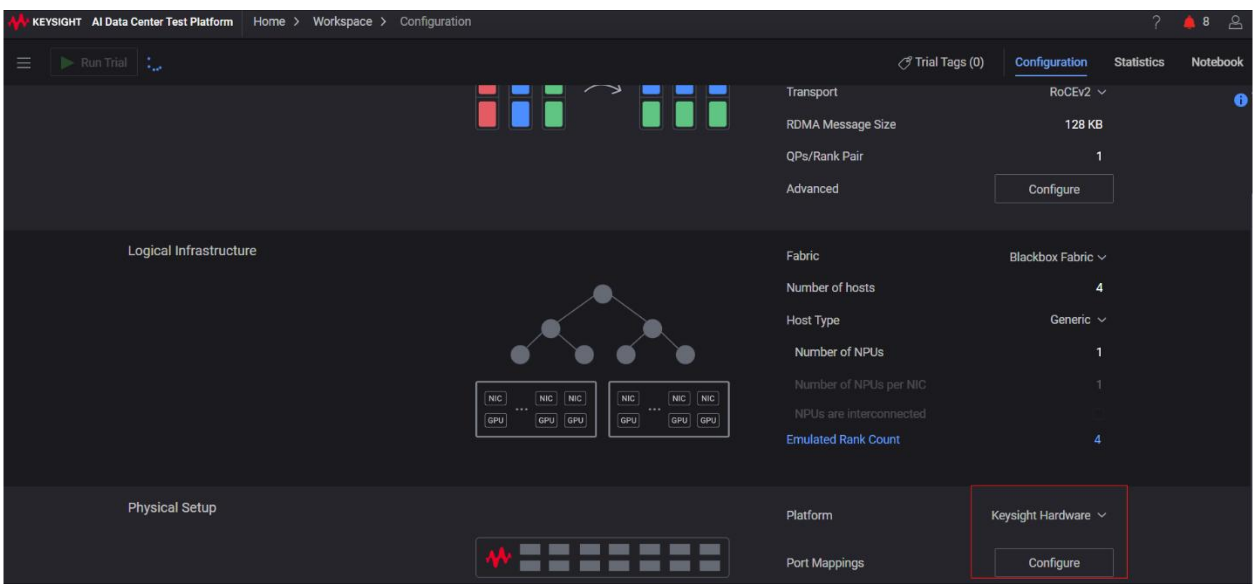
Step-by-step instructions

Background collective

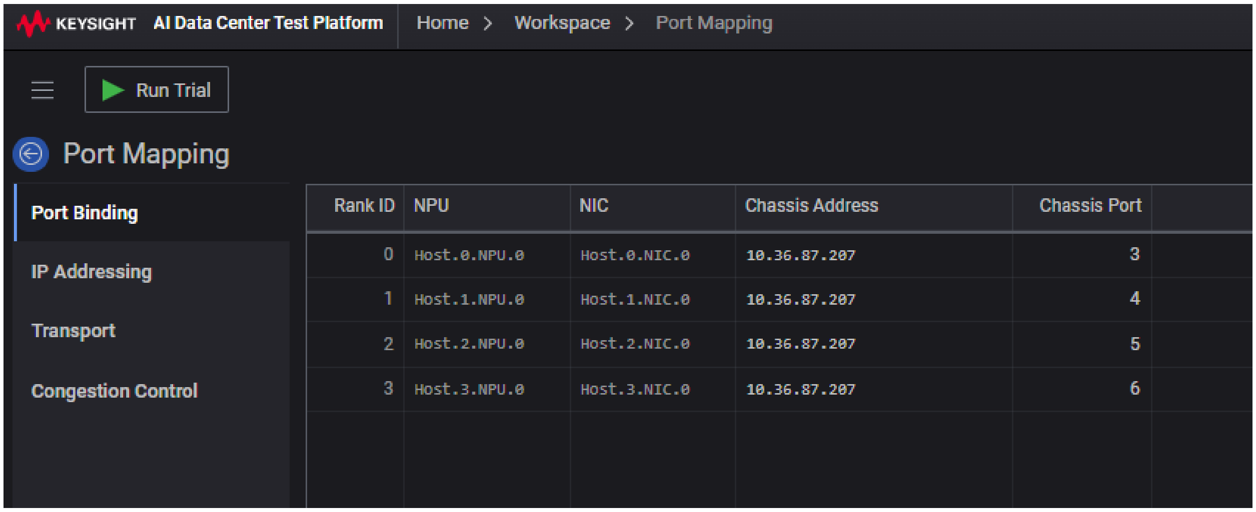
1. Set the number of hosts.



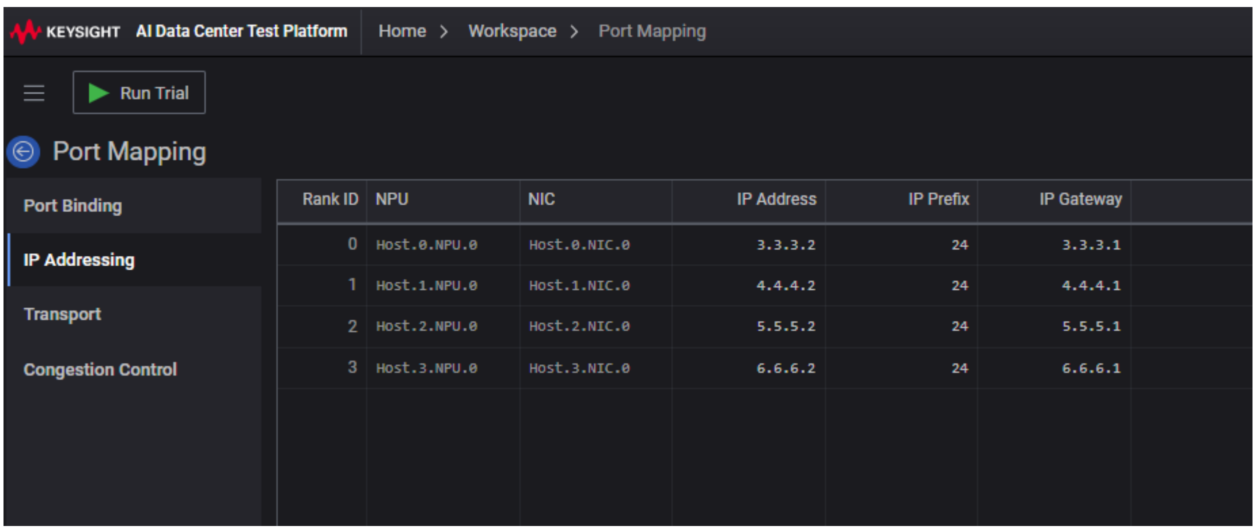
2. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



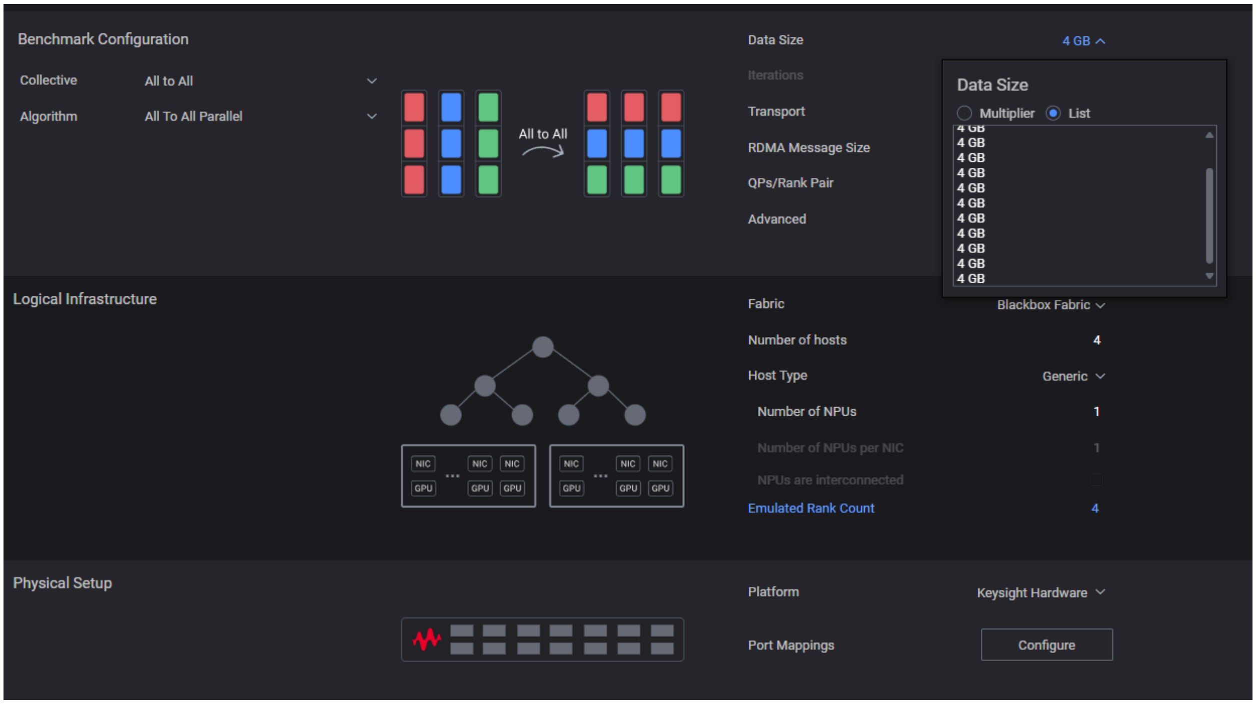
3. Set the port binding.



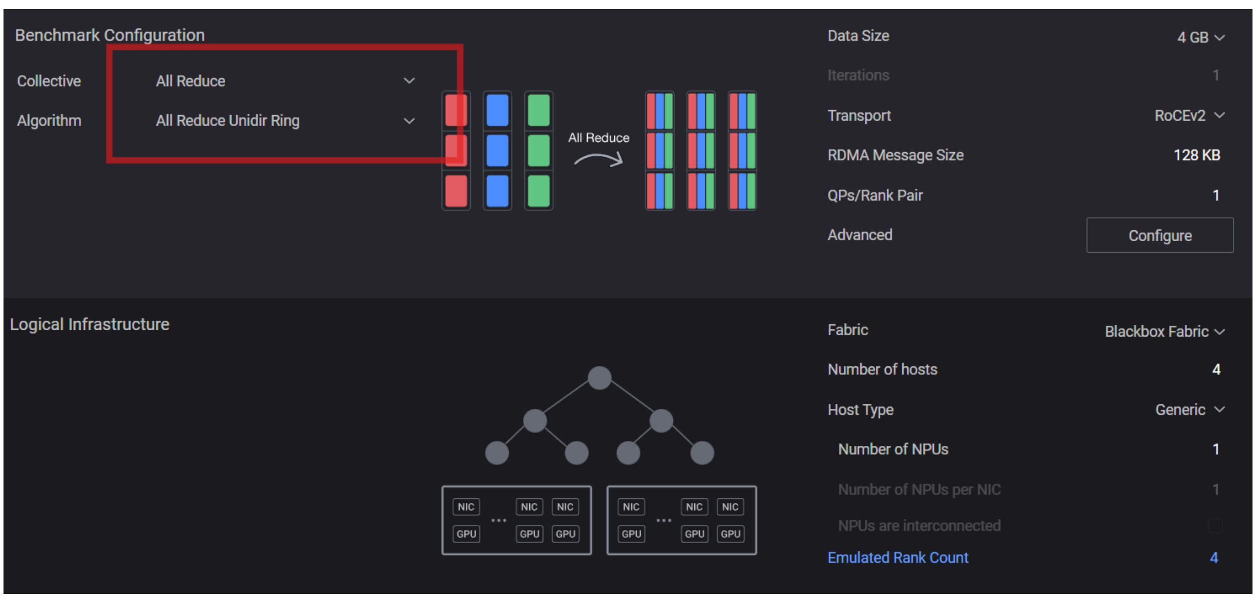
4. Set the IP address and gateway.



5. Set the data size, make background data size a fixed or random list.

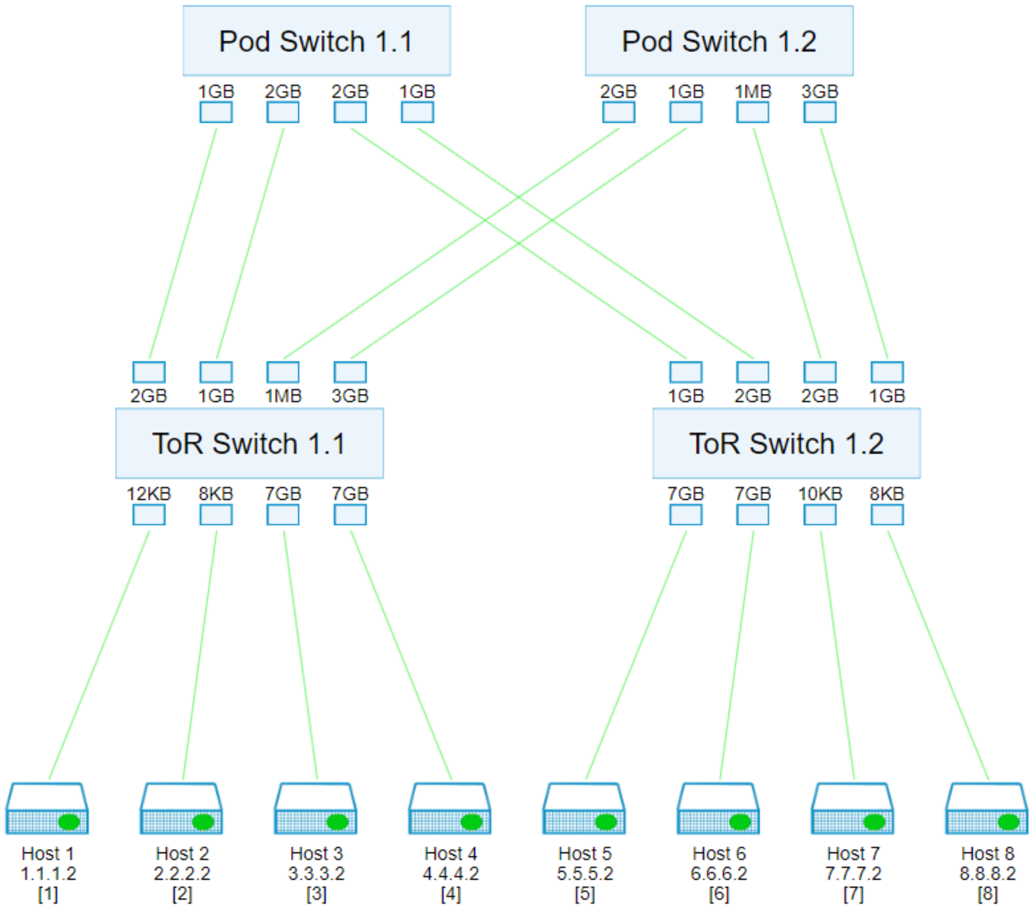


6. Set the collective and algorithms.



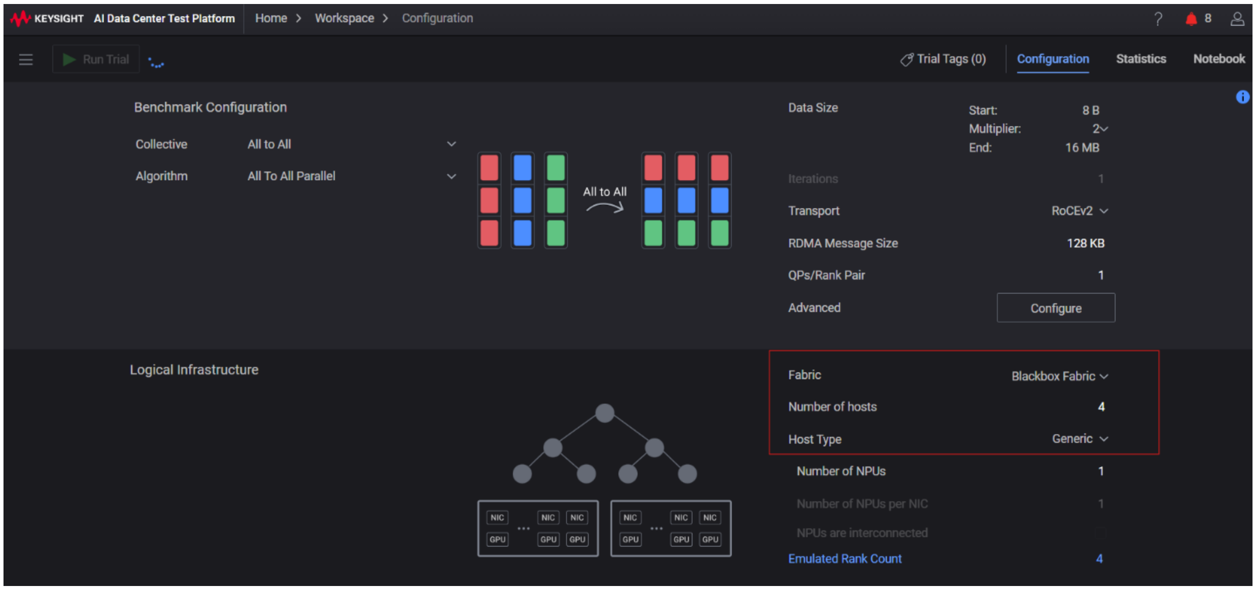
7. Select **Run Trial** to run background collectives first.

8. Make sure collective queue pairs are transmitting in the fabric.

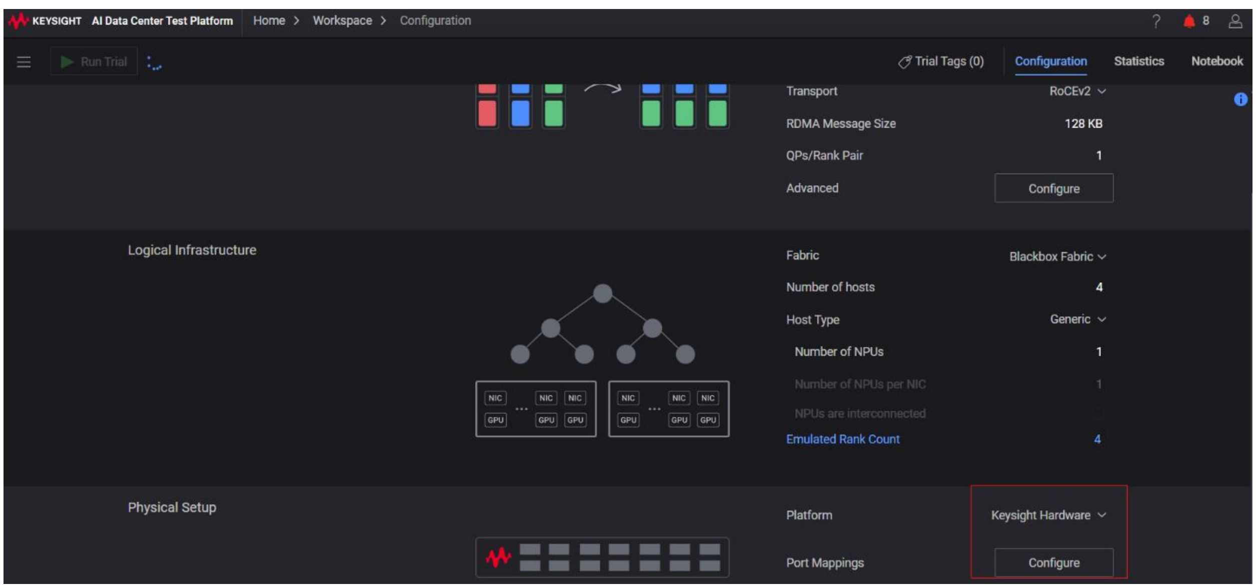


Collective under test

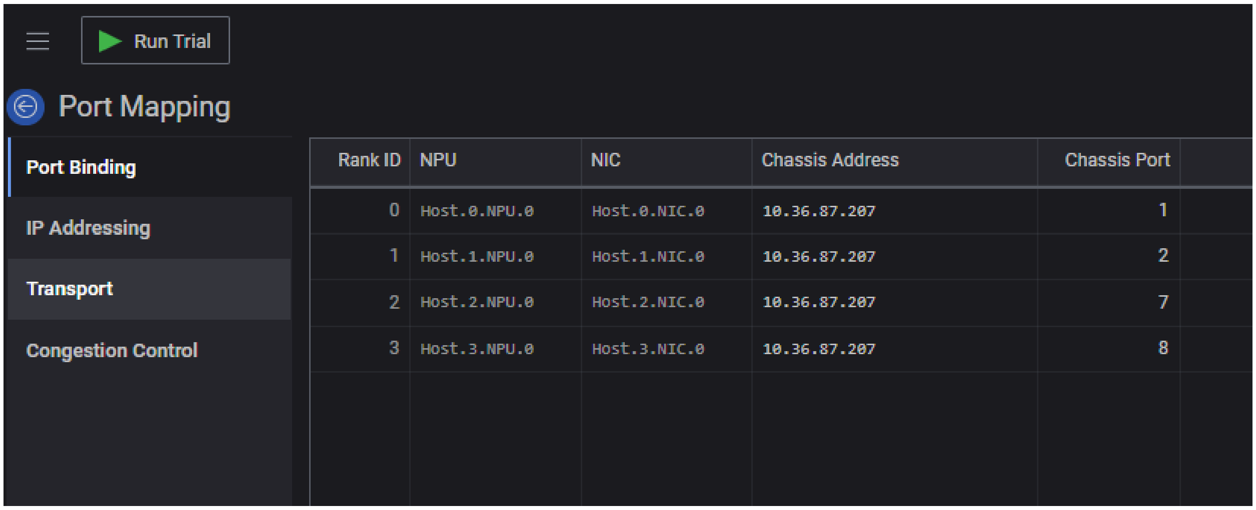
1. Set the number of hosts.



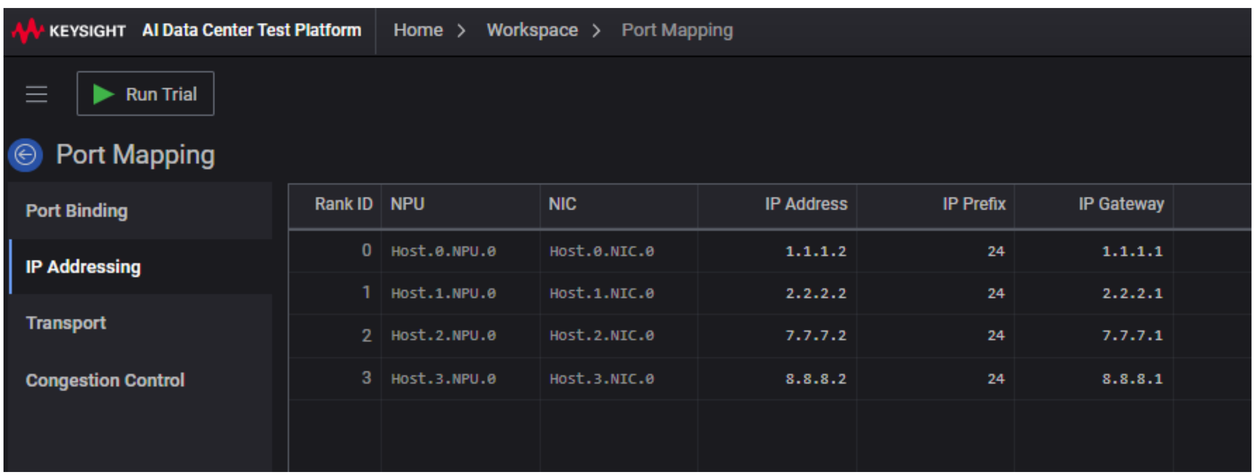
2. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



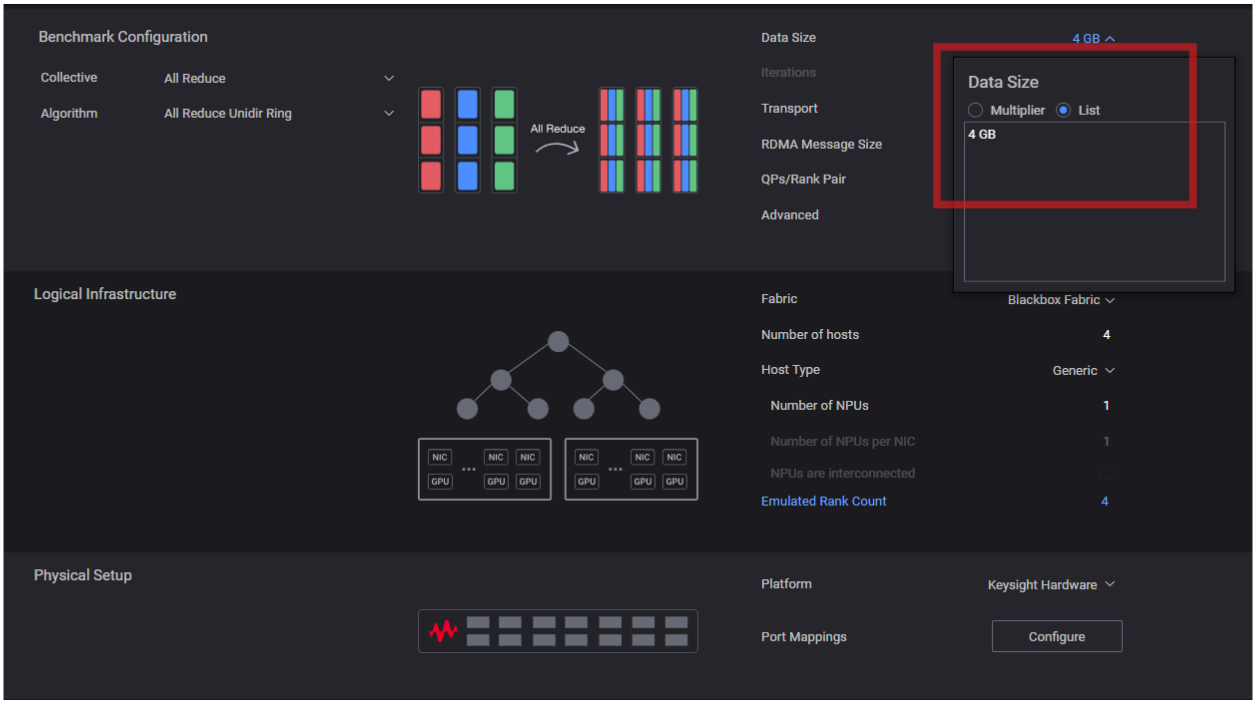
3. Set the port binding.



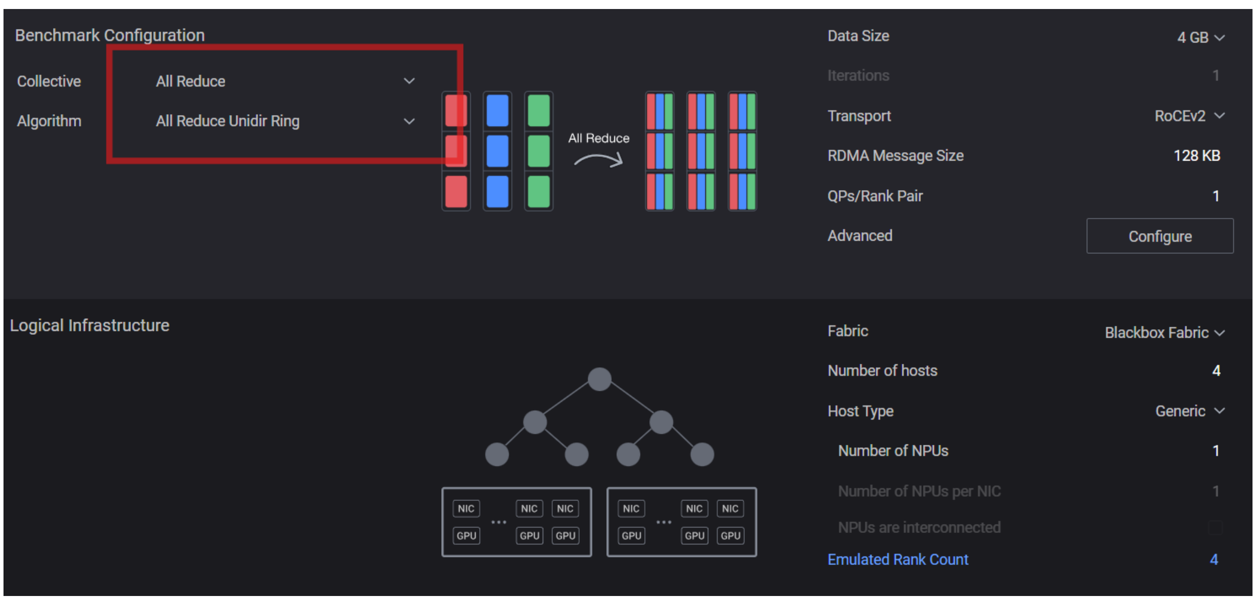
4. Set the IP address and gateway.



5. Set the data size.



6. Set the collective and algorithms.



7. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048 or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other collective to test: All-to-all, All-Gather-Ring.

- Test iteration

  Iteration reflects the batch in the context of AI training.

  A batch refers to a fixed-size group of input data that is used to train or test an AI model. Collectives for each batch keep the same data size.

- Test trial

  Collective bursts in fabric.

  Within a single batch, one or more than one collective may burst. Test trial may reflect the sequential collectives in the batch. The data size of collectives can vary depending on the specific AI model layer size or gradient size.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

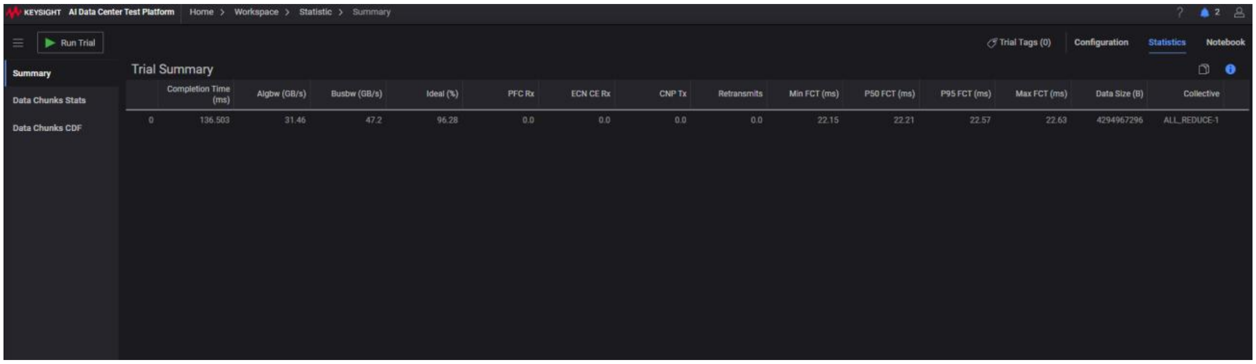
  Tensor parallel: Training data tensor size in bytes.

- Variant noises

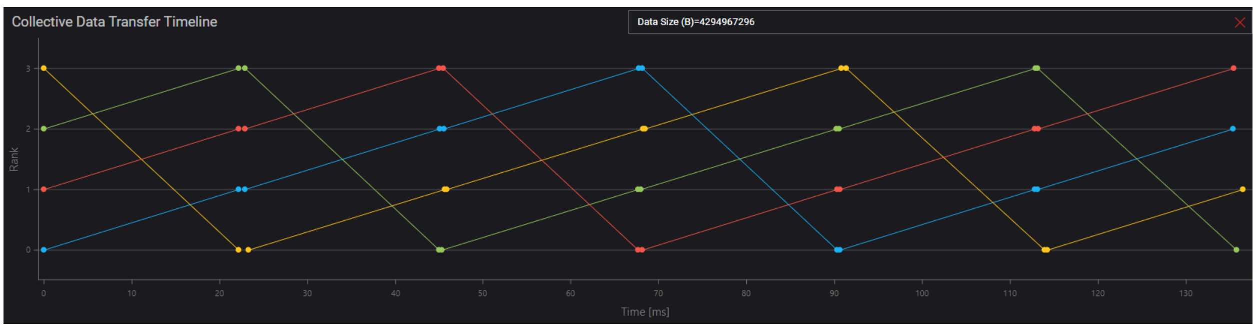
  Allocate multiple groups of collectives or data sizes as noisy background.

Results analysis

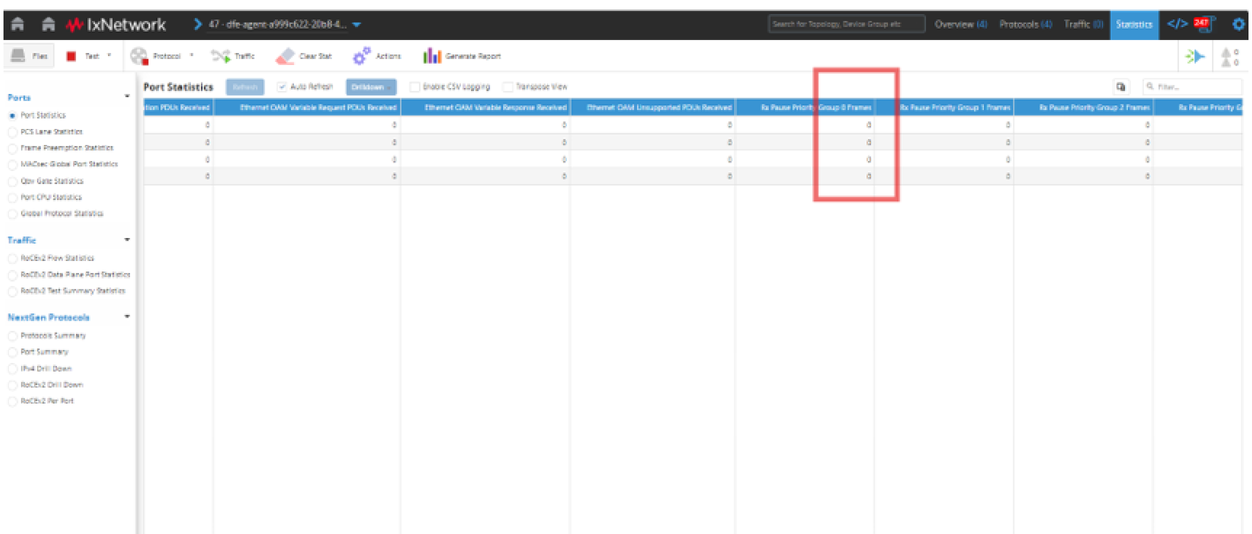
- The algorithm and bus bandwidth are high.



- No long tail observed.



- No PFC is triggered.



Conclusion

Under two all-reduce-ring collectives, target all-reduce ring collective can achieve excellent results.

Test case 3: Collective JCT with multiple tenants

Overview

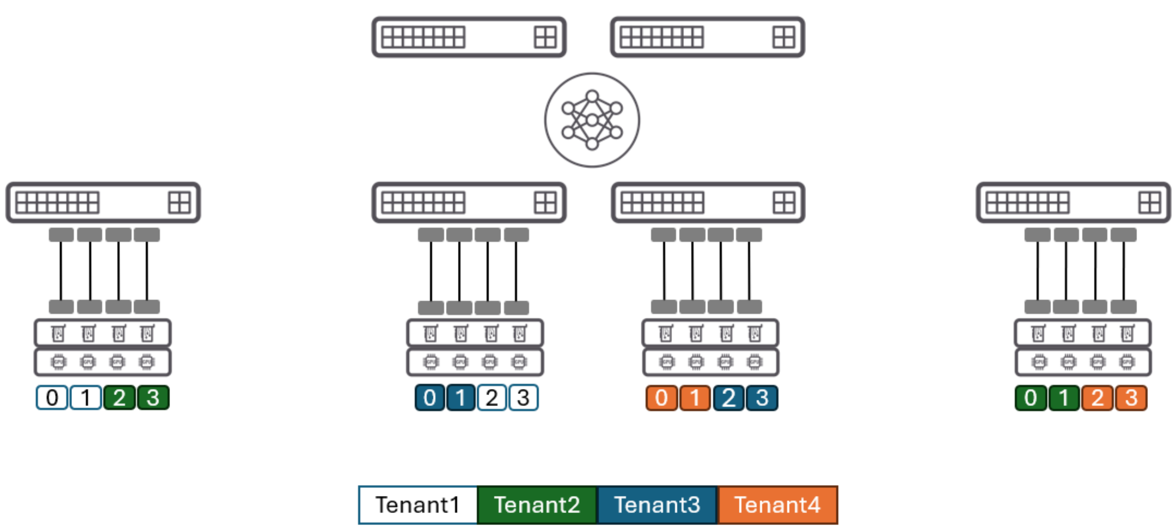
This test case leverages two continuous all-to-all collectives as noisy neighbors and two all-reduce-ring collectives to rigorously evaluate the data forwarding performance of the fabric by using a synthetic behavior with realistic loading patterns to emulate the transmission of data chunks through the ToR and pod switches. The SUT comprises the ToR switches, which are interconnected with multiple pod switches forming redundant paths through ECMP.

Objectives

1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Benchmark different implementations of load balancing.

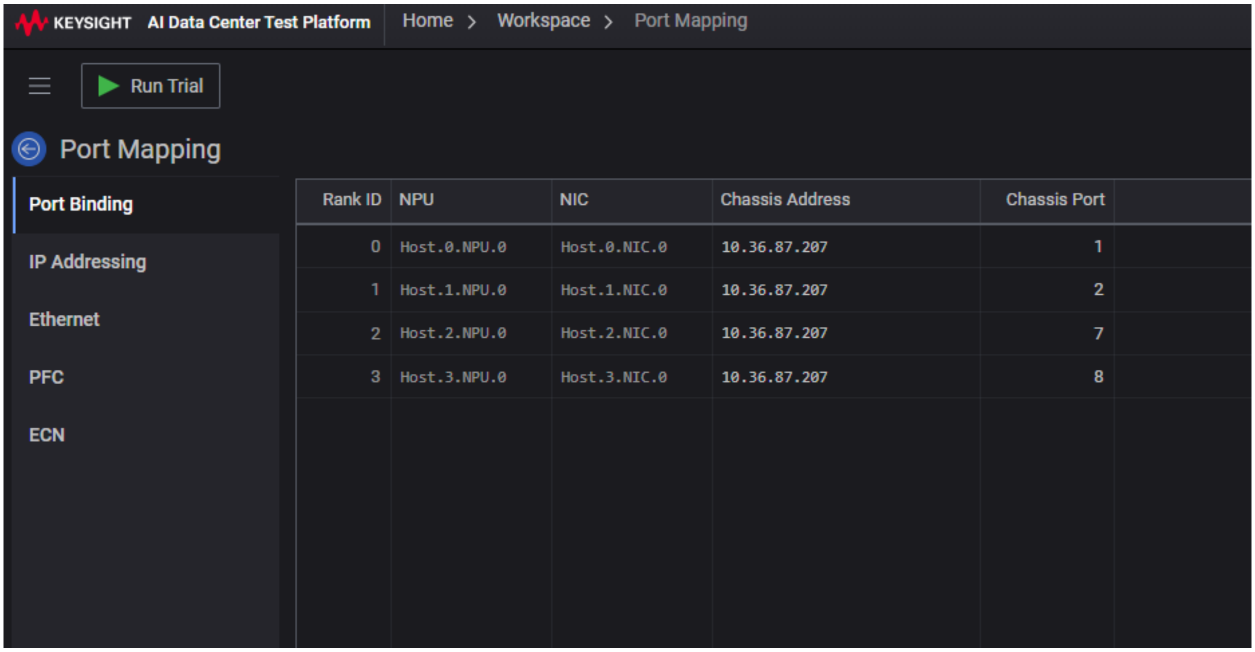
Setup

In this setup, four concurrent collectives are run within the fabric. The ranks of collectives are allocated at different leaf switches. On the DUT, PFC and ECN are enabled to ensure the control of congestion.

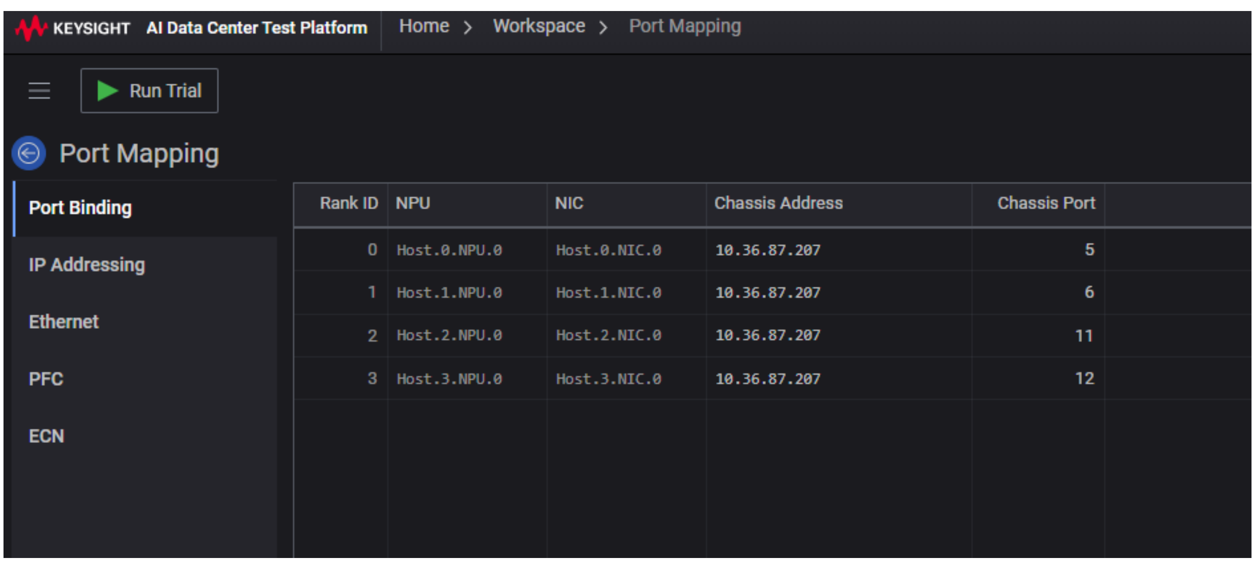


Step-by-step instructions

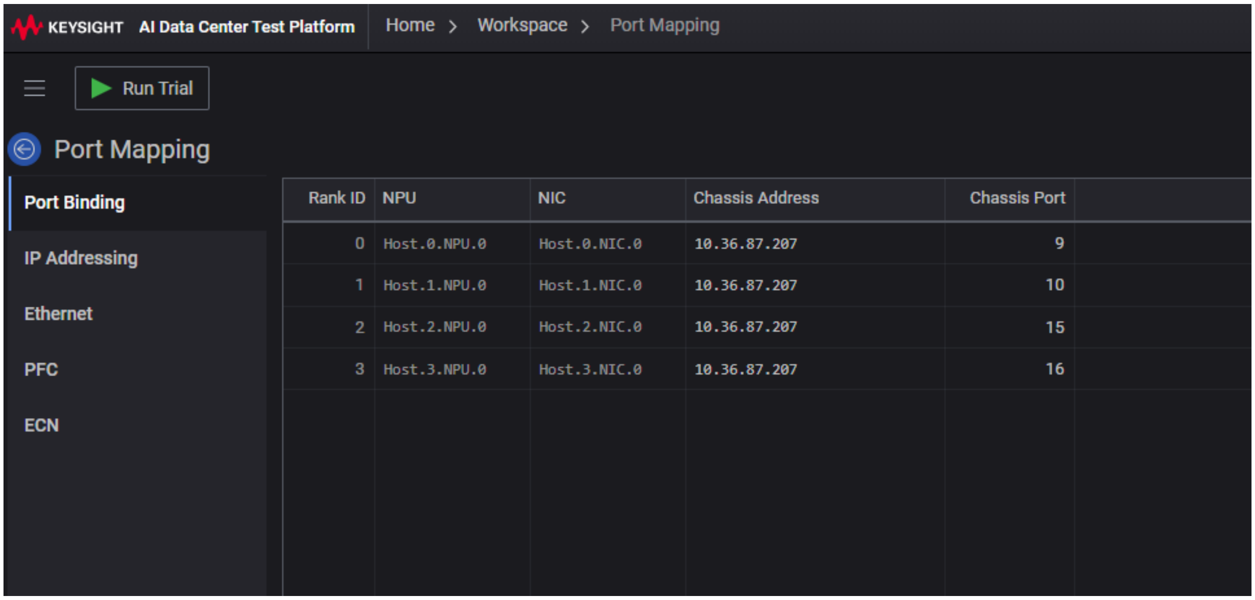
1. Set 4 port groups for Tenants.
2. Set the number of hosts to 4.
3. Set the port binding for Tenant 1.



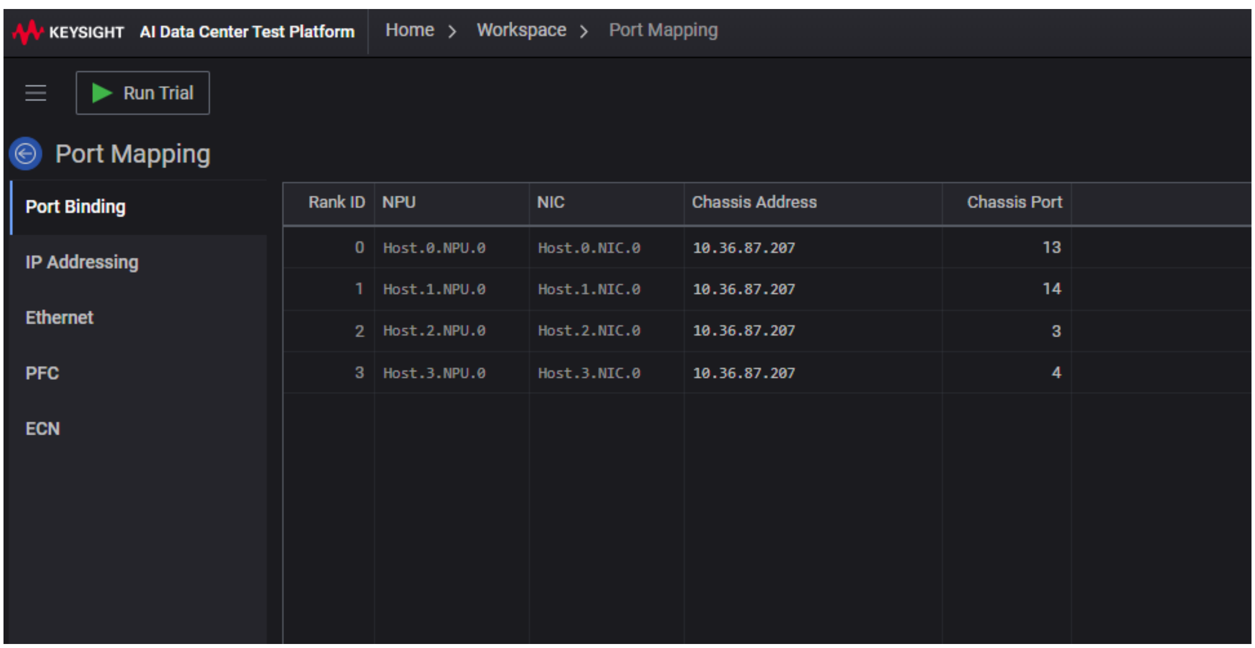
4. Set the port binding for Tenant 2.



5. Set the port binding for Tenant 3.



6. Set the port binding for Tenant 4.



7. Set data size, collective algorithm for different tenant users.

8. Select **Run Trial**

During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048 or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other collective to test: All-to-all, All-Gather-Ring.

- Test iteration

  Iteration reflects the batch in the context of AI training.

  A batch refers to a fixed-size group of input data that is used to train or test an AI model. Collectives for each batch keep the same data size.

- Test trial

  Collective bursts in fabric.

  Within a single batch, one or more than one collective may burst. Test trial may reflect the sequential collectives in the batch. The data size of collectives can vary depending on the specific AI model layer size or gradient size.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

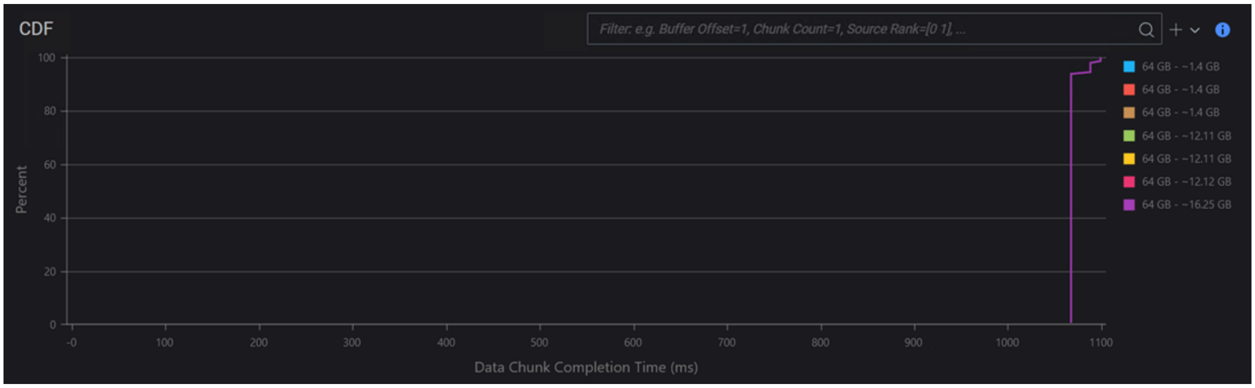
  Tensor parallel: Training data tensor size in bytes.

- Variant noises

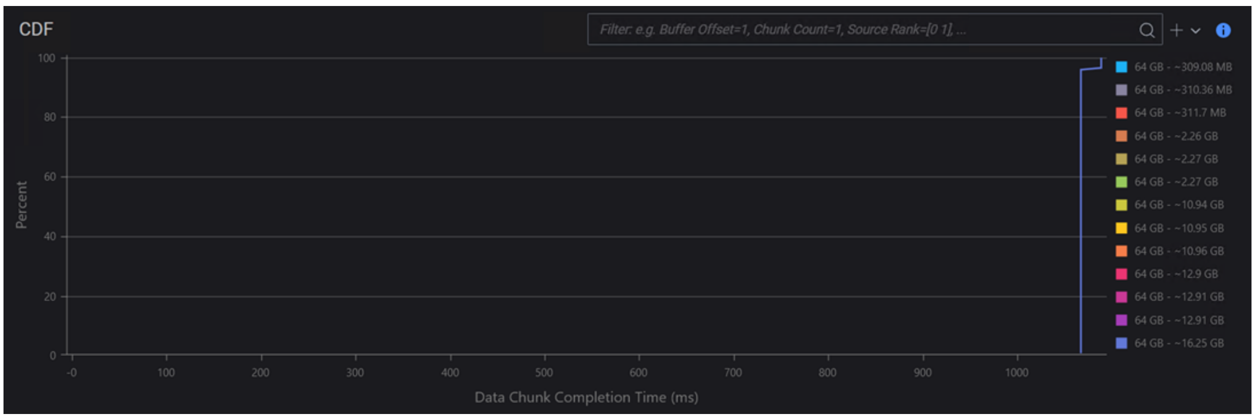
  Allocate multiple groups of noises of variant collectives or data size.

Results analysis

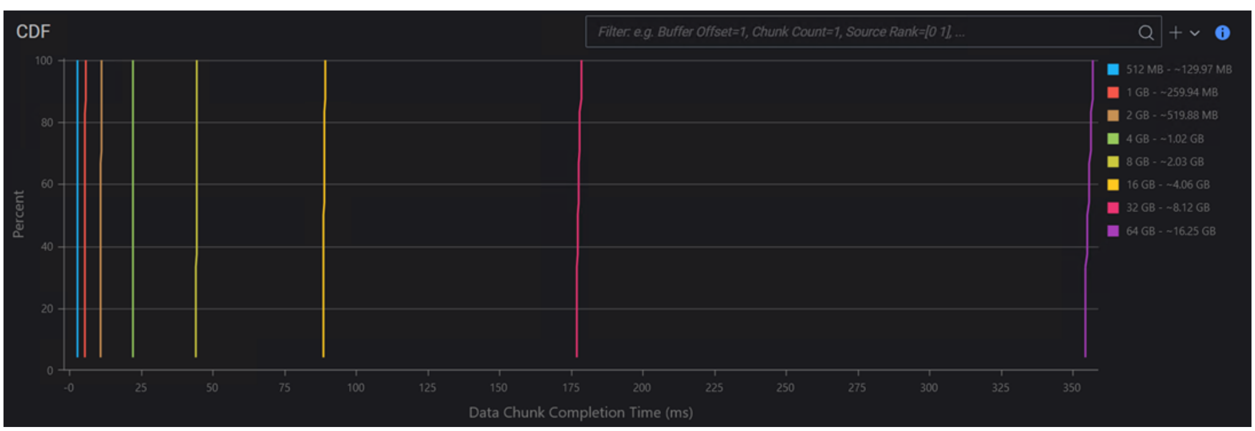
- Tenant 1



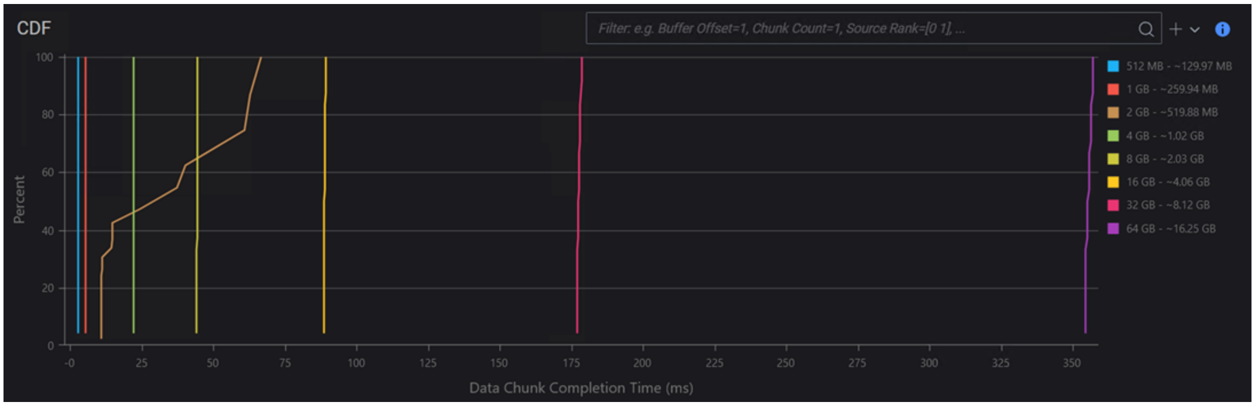
- Tenant 2



- Tenant3



- Tenant 4



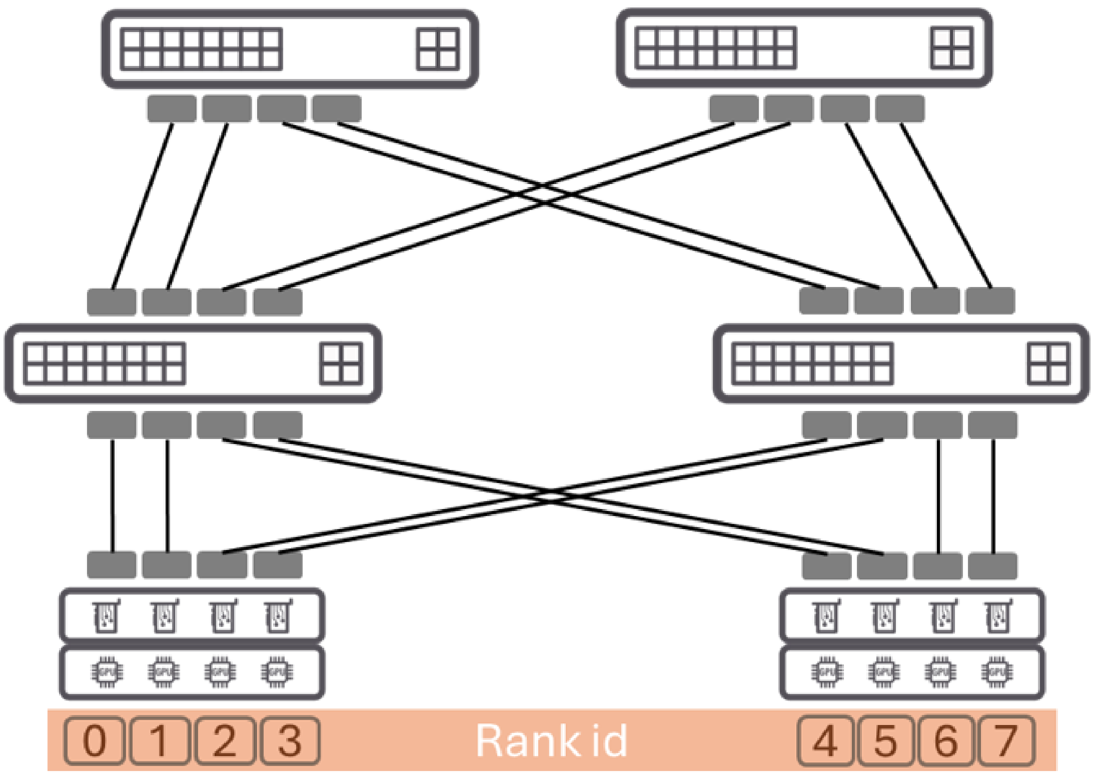
Conclusion

During the test, one trial (2 GB) from tenant 4 experienced a long-tail latency, indicating congestion mitigation had occurred.

Test Methodology for Load Balancing

In traditional Ethernet fabrics interconnecting NPUs, few flows often lead to flow collisions, which is a natural consequence of low entropy flows. This inherent limitation results in unavoidable collisions that can significantly impact the transfer time for affected flows. As more flows collide on the same link, the transfer time for each individual flow doubles, leading to an exponential deterioration in performance with increased collision rates. Consequently, this phenomenon has a direct impact on the overall Artificial Intelligence / Machine Learning (AI / ML) training job completion time, increasing the duration and reducing the efficiency of these critical processes.

In this experiment, we emulate a scenario featuring two NPU hosts, each equipped with four NPU and NICs. To fully leverage scale-out capabilities by utilizing available network resources, a best practice is to connect the first two Network Interface Cards (NICs) of the host to one switch, while connecting the remaining two NICs to another switch. We employ the all-to-all and ring algorithm, all-gather and all-reduce collective operations. Under these conditions, the network fabric has multiple egress queue pairs, may experience congestion, enabling us to validate the packet distribution, link usage, and packet loss.



Perform the following test procedure:

1. Set the peering test ports with IP and gateway.
2. Set the rank id as [0, 1, 4, 5, 2, 3, 6, 7].
3. Set the shared buffer size 32 MB.
4. Set the data size 200 MB.
5. Set the algorithm.
6. Run trials.

The following test cases demonstrate the test methodologies and detailed step-by-step configuration with automated test packages available in CB:

Test Collectives	Data Size	Write msg size	Parallel QPs	Congestion control
All-to-all	200 MB	128 KB	1	N
All-gather Ring	200 MB	128 KB	1	N
All-gather Ring	200 MB	128 KB	4	N
All-gather Ring spray	200 MB	128 KB	1	N

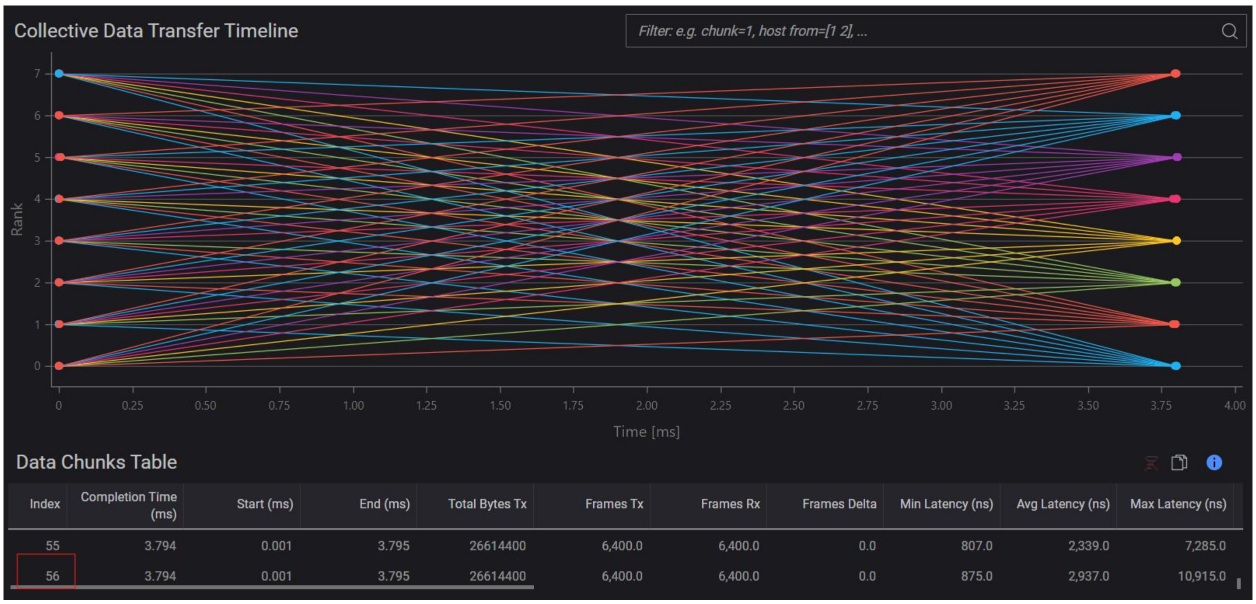
Test case 1: All-to-all with 5-tuple hash ECMP

Overview

This test case employs an all-to-all collective operation to evaluate the ECMP performance of the fabric. The all-to-all architecture is characterized by a significant amount of overhead, requiring 56 QP instances for a single operation and 186 MB of memory per data chunk. This translates to an approximately 26.6 MB data size for a single QP during transmission. Meanwhile, the SUT consists of ToR switches, interconnected with multiple pod switches, forming redundant paths by using the 5-tuple Hash-based ECMP mode.

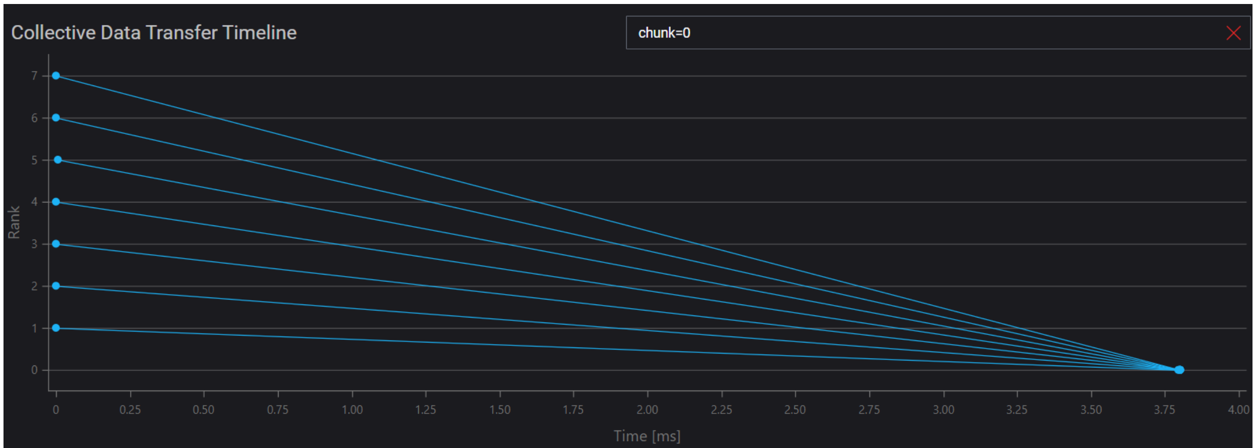
For All-to-all collective:

- Total QPs = 56



- Single QP = 26.6 MB

- Chunk size = 26.6 M * 7 = 186 MB



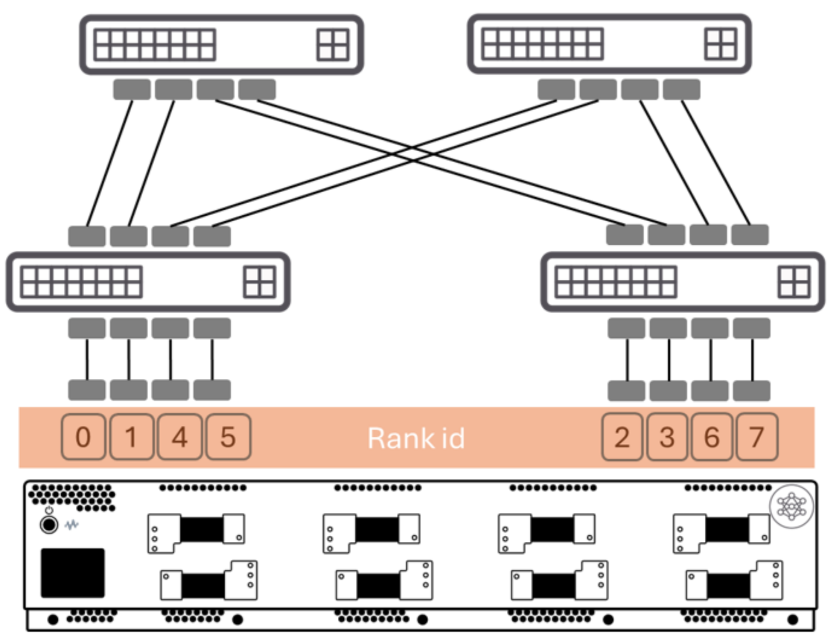
Data Chunks Table

Index	Completion Time (ms)	Start (ms)	End (ms)	Total Bytes Tx	Frames Tx	Frames Rx	Frames Delta	Min Latency (ns)	Avg Latency (ns)	Max Latency (ns)
1	3.796	0.0	3.796	26614400	6,400.0	6,400.0	0.0	847.0	2,951.0	9,625.0
2	3.8	0.0	3.8	26614400	6,400.0	6,400.0	0.0	2,112.0	4,996.0	12,307.0

Objectives

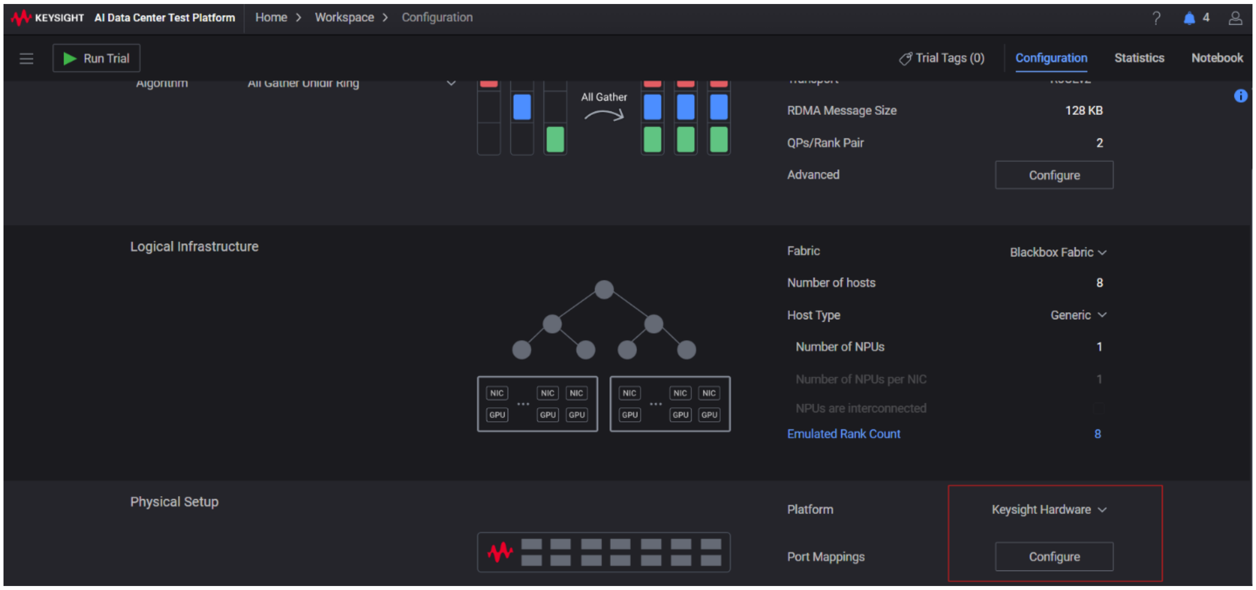
1. Distribute packets across multiple paths.
2. Observe links usage.
3. Measure packet loss / drop rates.

Setup

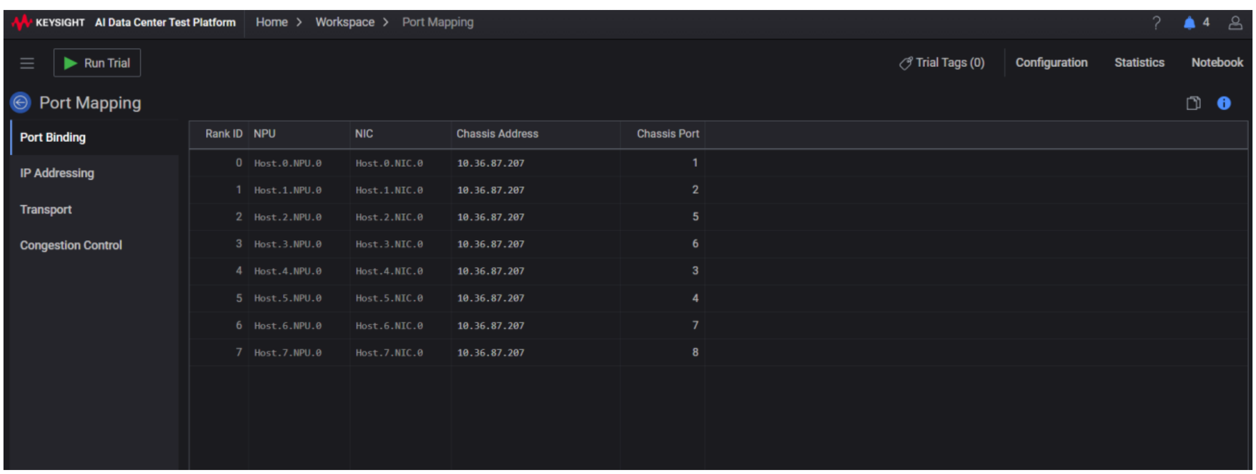


Step-by-step instructions

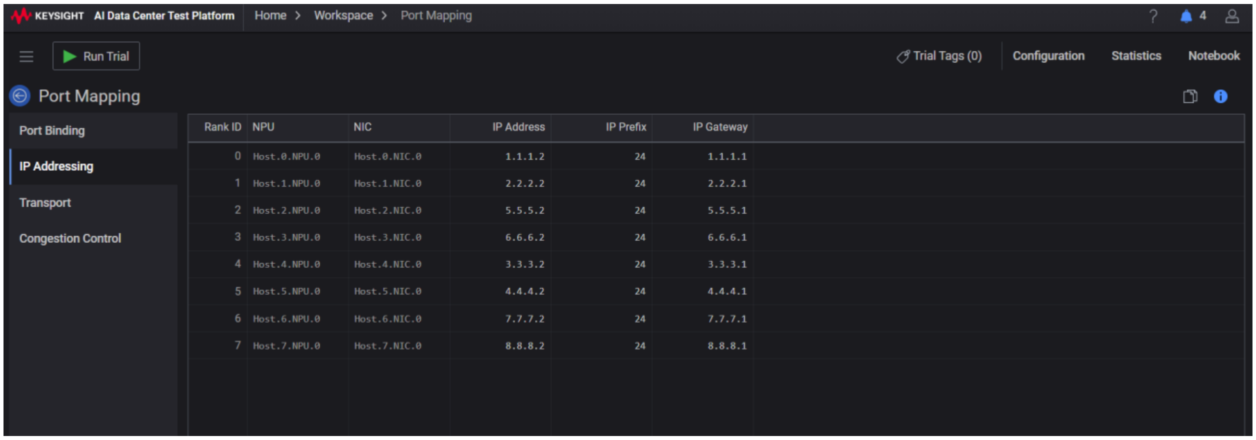
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



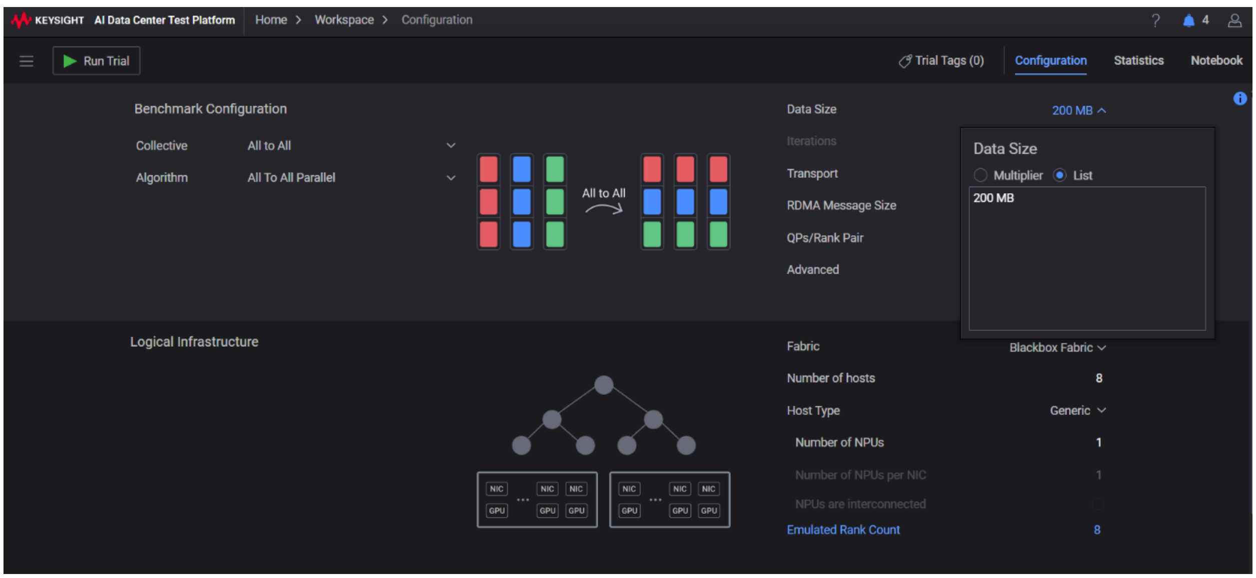
2. Set the port binding. Use rank ID in the order of [0, 1, 4, 5, 2, 3, 6, 7], which represents how the emulated NPUs are binded with physical AresONE test ports.



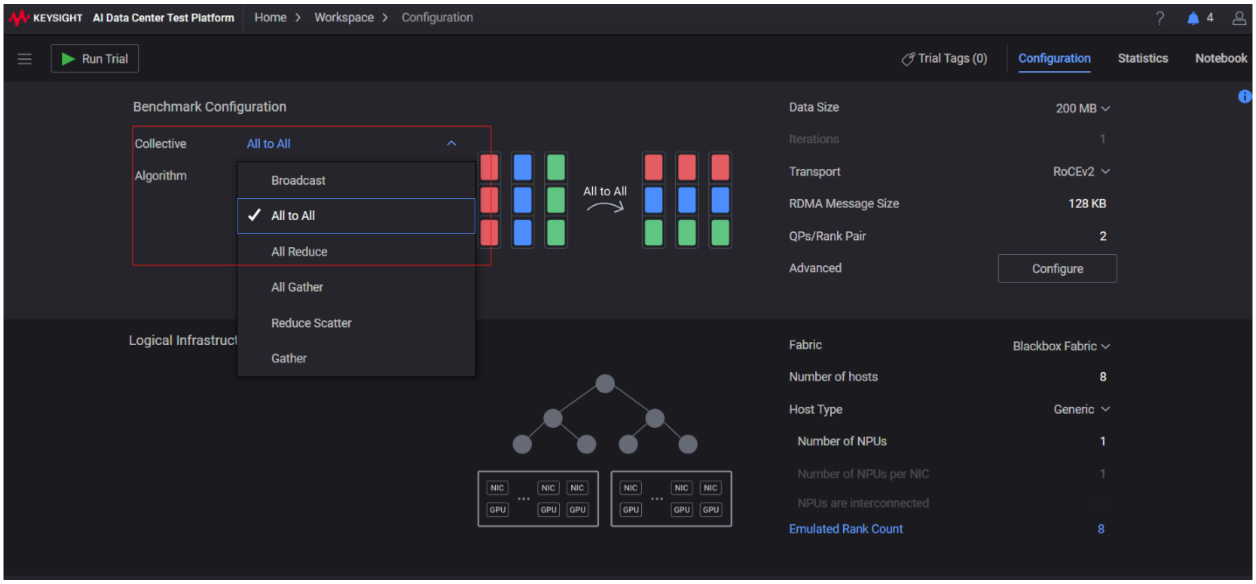
3. Set the IP address and gateway.



4. Set the data size.

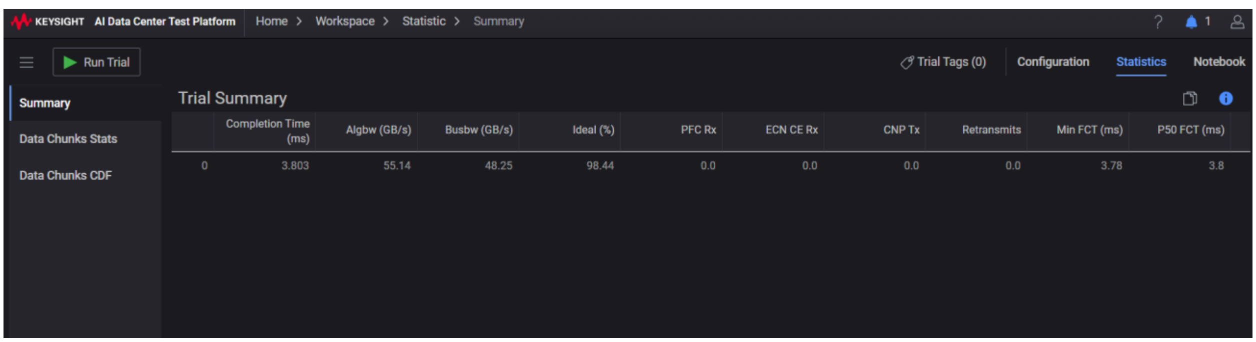


5. Set the collective and algorithms.



6. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.



Test variables

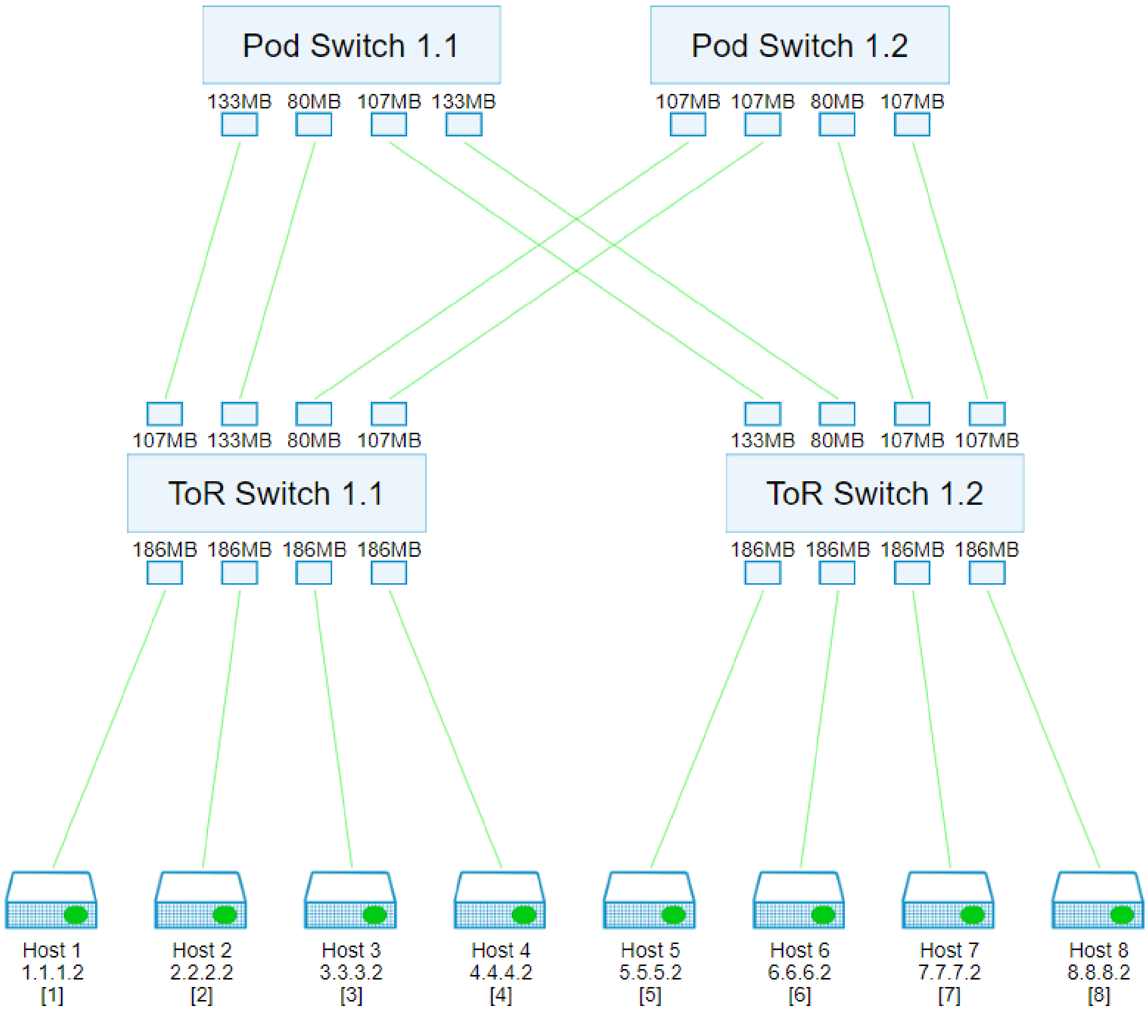
Parallel-QPs

When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

Results analysis

As per the test results, the transmit bytes on the egress port are evenly distributed across all links, indicating full usage.

Bytes count (TX) on SUT

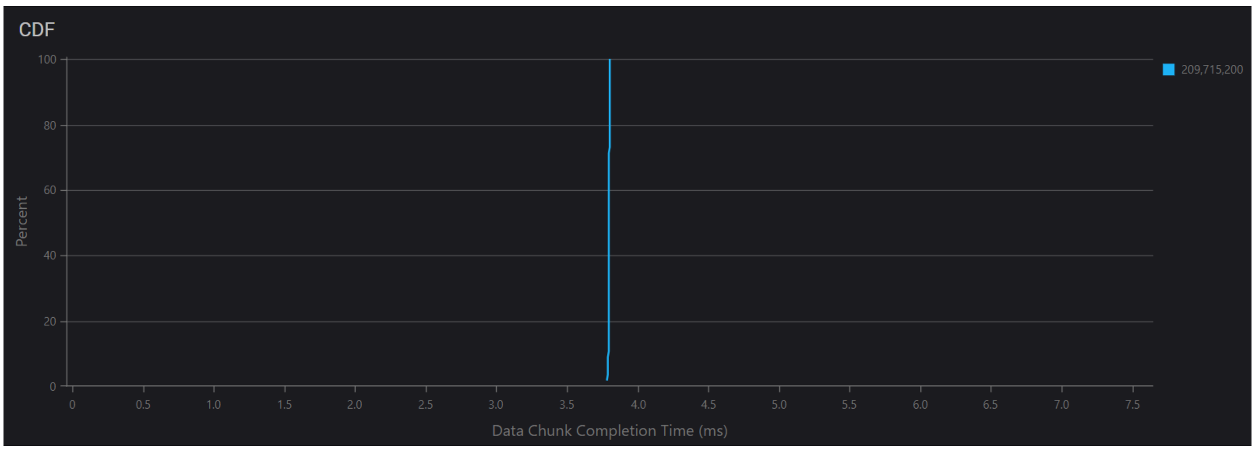


Egress QPs for switches

Switch	Egress QPs	Link usage (%)	Distribution	Frames delta
ToR Switch 1.1	16	100	4 + 5 + 3 + 4	0
ToR Switch 1.2	16	100	5 + 3 + 4 + 4	0

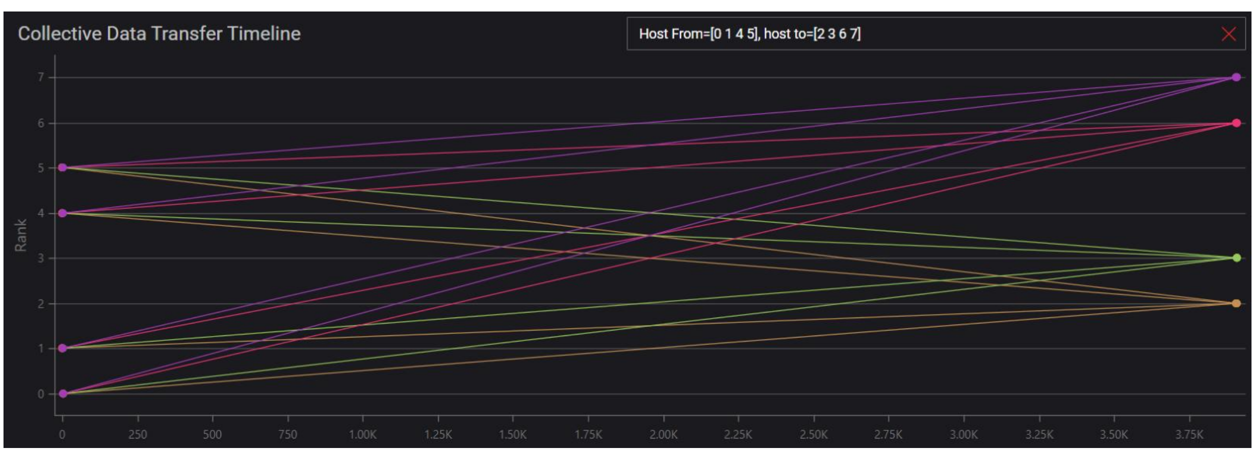
The ideal percentage against bus bandwidth is high, indicating no congestion and high throughput during forwarding.

There is no long tail latency shown in cumulative distribution function (CDF).



Conclusion

Each ToR switch egresses with 16 QPs, which are strategically distributed throughout the fabric to ensure optimal usage. This even distribution enables effective load balancing, effectively eliminating congestion and ensuring a seamless flow of data.



Test case 2: All-gather-ring with 5-tuple hash ECMP

Overview

This test case employs an all-gather-ring collective operation to evaluate the ECMP performance of the fabric. The all-gather ring with traffic flowing in a circular pattern from node 0 to 7, requiring 56 QP instances for a single operation and 186 MB of memory per data chunk. This translates to an approximately 26.6 MB data size for a single QP during transmission. Meanwhile, the SUT consists of ToR switches, interconnected with multiple pod switches, forming redundant paths by using the 5-tuple Hash-based ECMP mode.

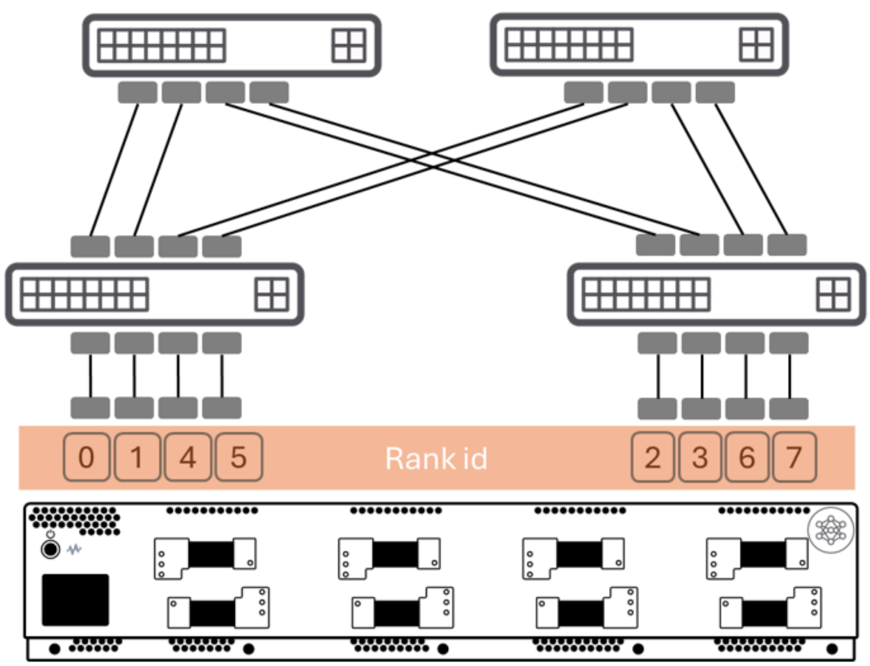
For All-gather-ring collective:

- Total QPs = 56
- single QP = 26.6 MB
- Chunk size = 26.6M * 7 = 186 MB

Objectives

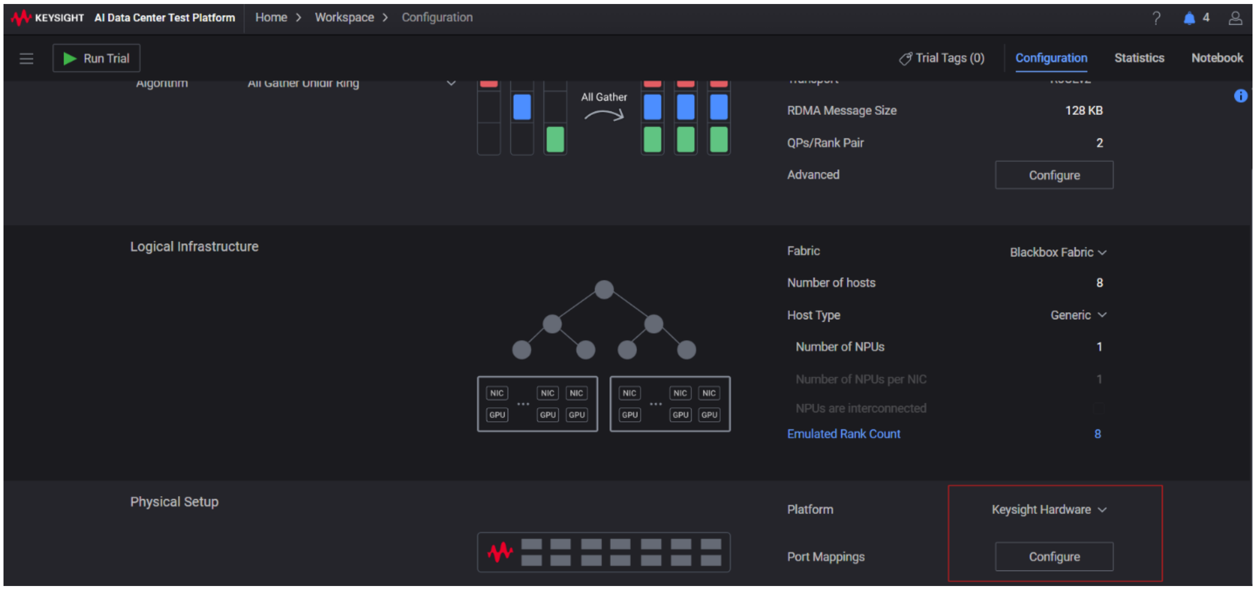
1. Distribute packets across multiple paths.
2. Observe links usage.
3. Measure packet loss / drop rates.

Setup

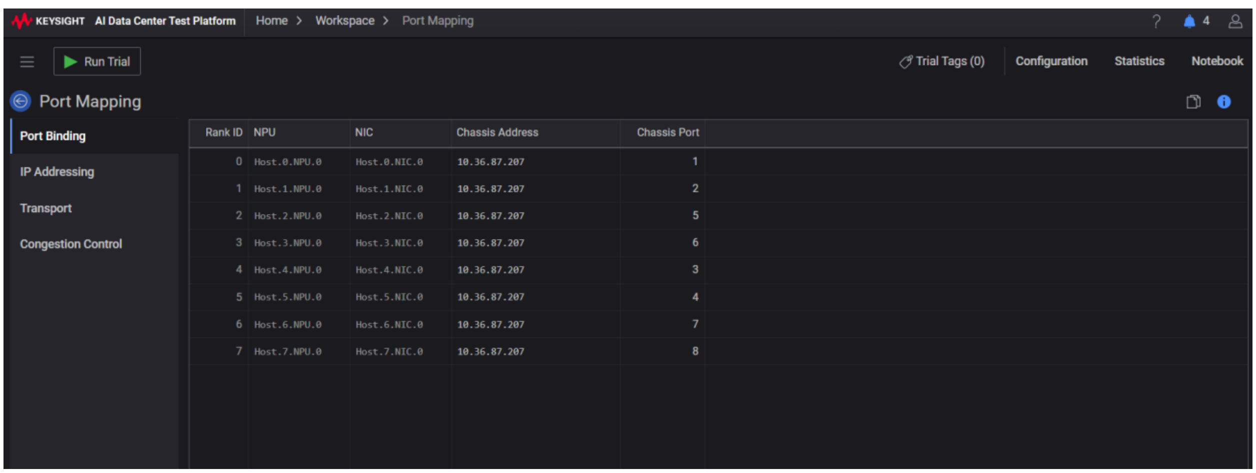


Step-by-step instructions

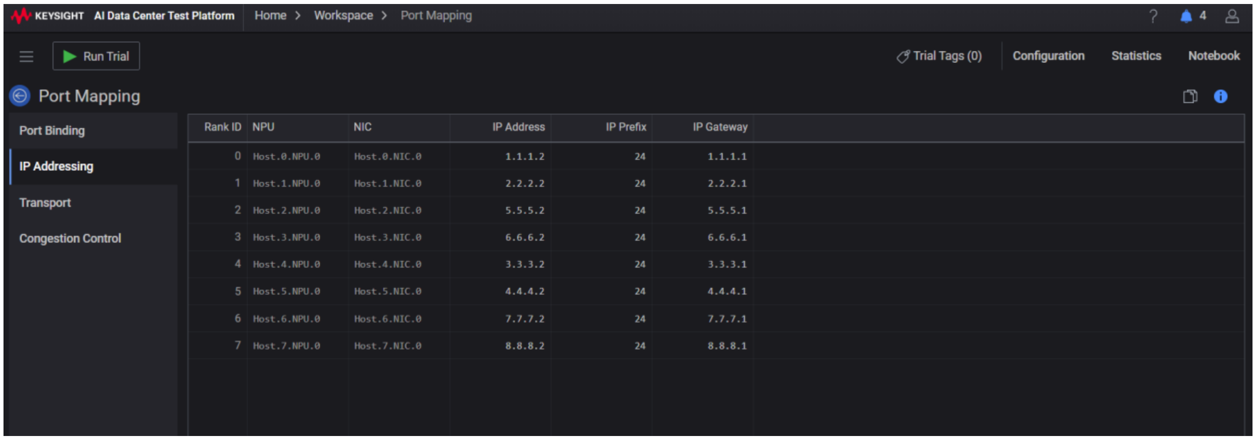
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



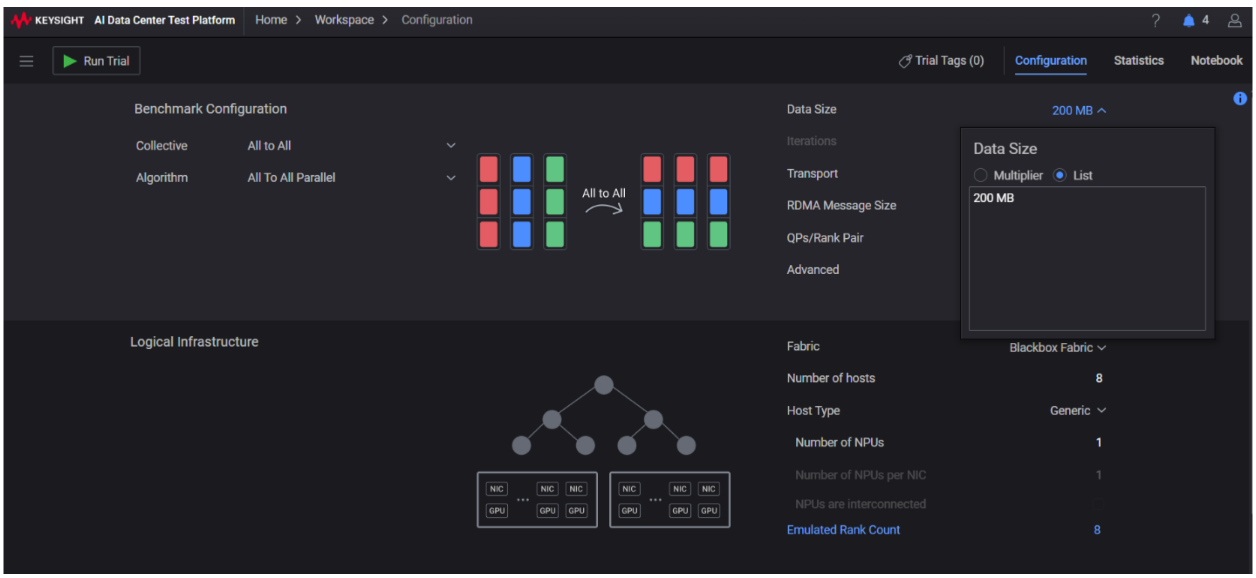
2. Set the port binding.



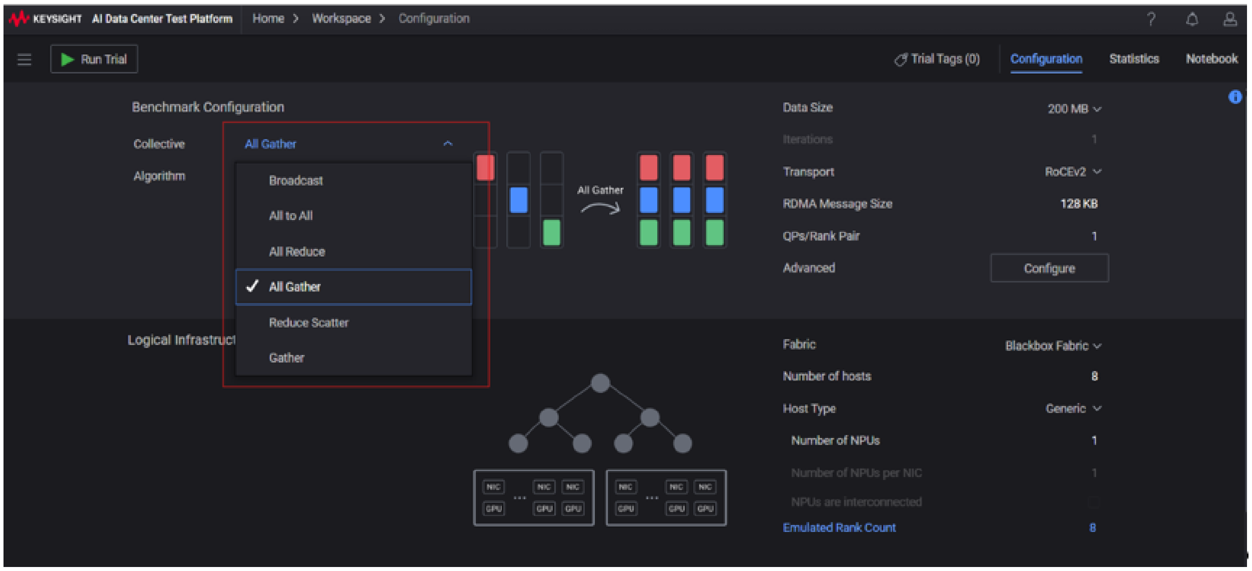
3. Set the IP address and gateway.



4. Set the data size.

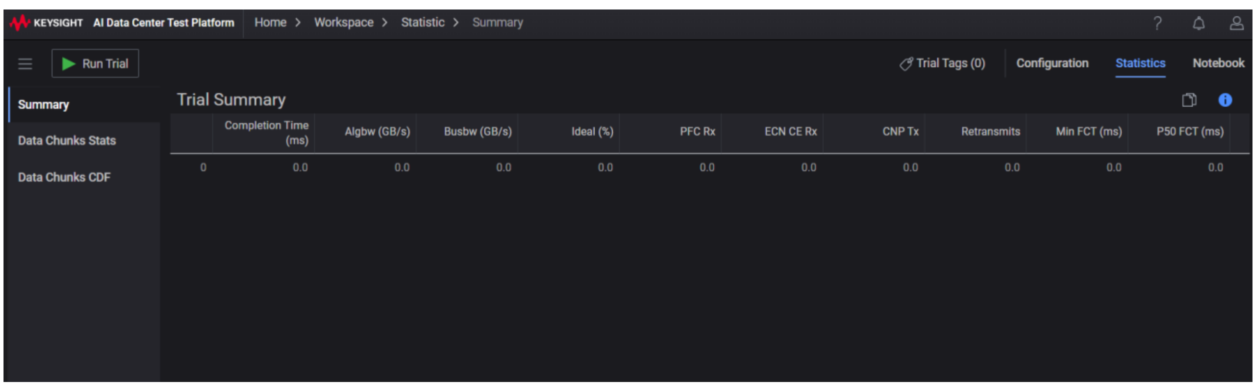


5. Set the collective and algorithms.



6. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.



Test variables

- Parallel-QPs

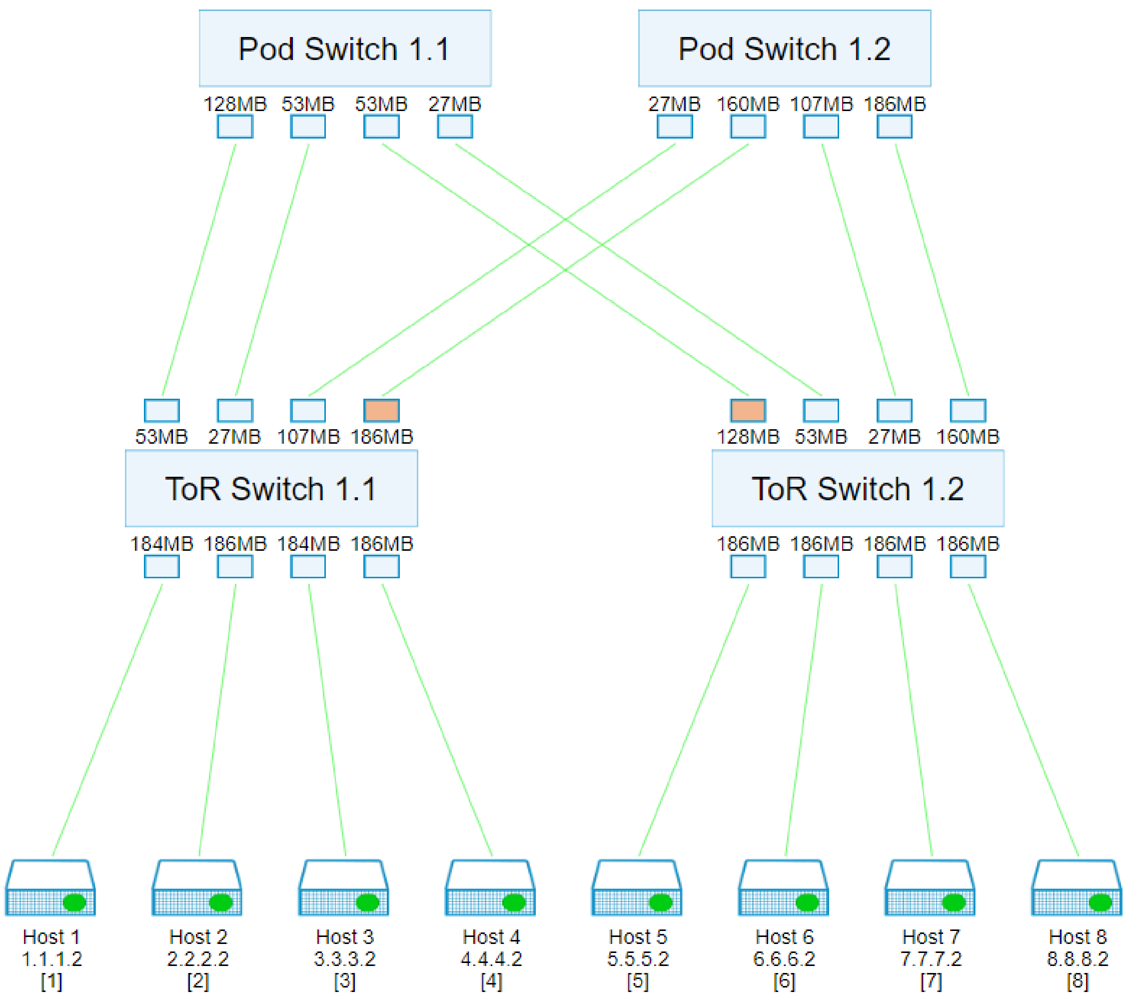
  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

- All-reduce-ring

Results analysis

The training job is likely to fail because of packet drops caused by the fabric's performance issues.

Bytes count (TX)



Egress QPs for switches

Switch	Egress QPs	Link usage (%)	Distribution	Frames delta
ToR Switch 1.1	14	100	2 + 1 + 4 + 7	666
ToR Switch 1.2	14	100	5 + 2 + 1 + 6	607

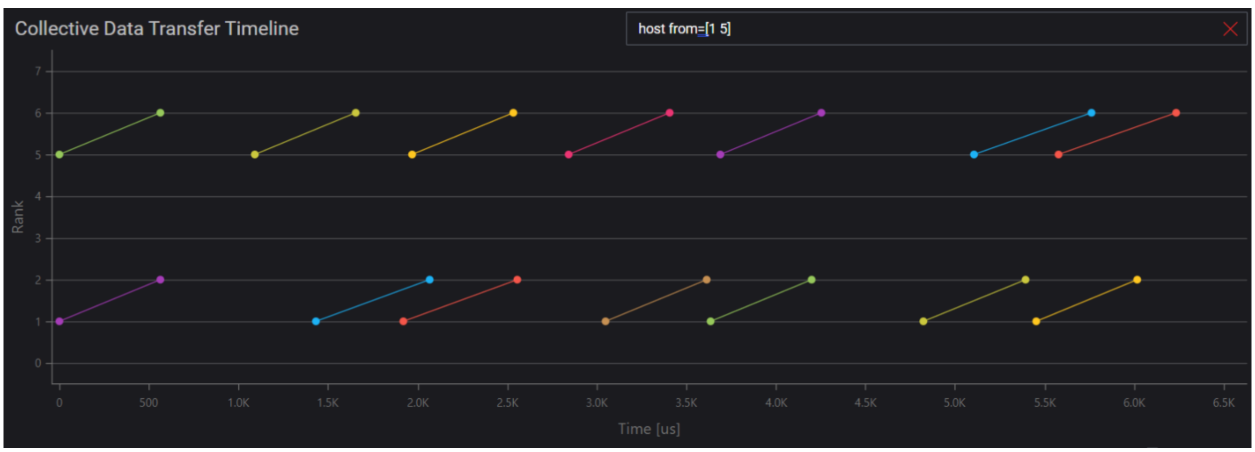
Packet loss

Data Chunks Table

es Delta	Min Latency (ns)	Avg Latency (ns)	Max Latency (ns)	Ecn-CE Rx	Cnp Tx	Cnp Rx	Total Bytes Rx	Busbw (GB/s)	Source Rank	Destination Rank
666.0	2,155.0	240,300.0	485,152.0	0.0	0.0	0.0	23844836	0.0	3	4
607.0	2,110.0	235,580.0	422,252.0	0.0	0.0	0.0	24090222	0.0	7	0

Conclusion

Each ToR switch has 14 QPs, which are not well balanced throughout the fabric, leading to congestion. As a result, we can observe frame delta statistics in the Data Chunks table. The Collective Data Transfer Timeline graph reveals that two QPs consistently arrive at the same egress port at the same time, causing congestion.



Test case 3: All-gather-ring with multi-QP

Overview

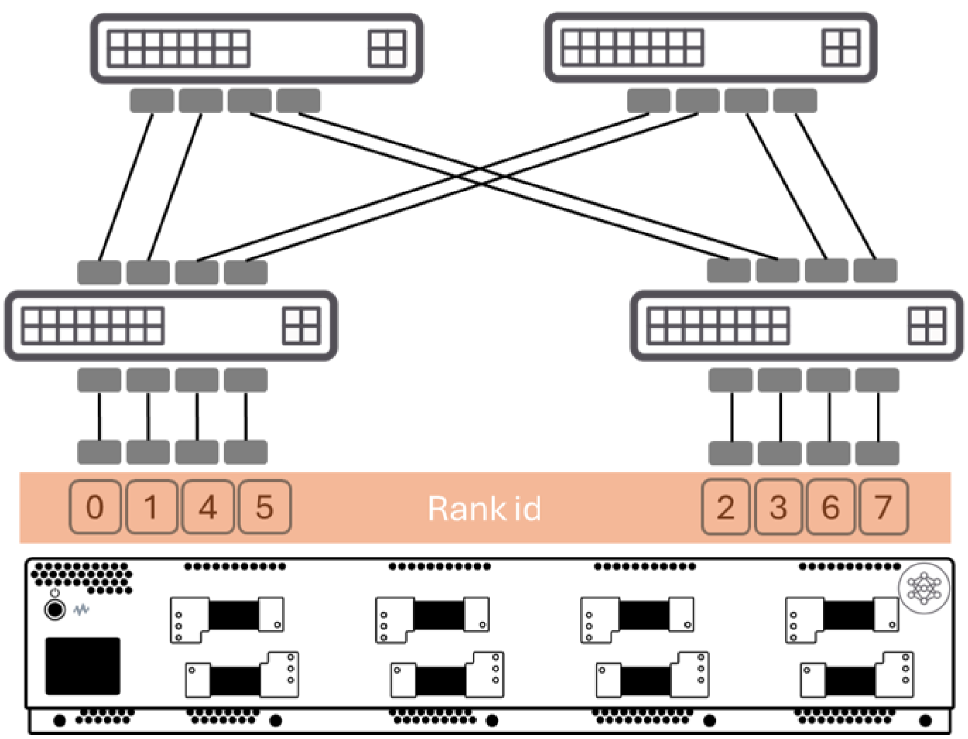
This test case uses parallel QPs to address the unbalanced hashing issue in the fabric, thereby achieving improved performance through an all-gather-ring collective operation. The all-gather-ring with traffic flowing in a circular pattern from node 0 to 7, requiring 56 * 4 (M-QPs) QP instances for a single operation and 186 MB of memory per data chunk. This translates to an approximately 6.6 MB data size for a single QP during transmission. Meanwhile, the SUT consists of ToR switches, interconnected with multiple pod switches, forming redundant paths by using the 5-tuple Hash-based ECMP mode.

- QPs = 56 * 4
- 186 MB for every chunk, 186 / (7*4) = 6.6 M for single QP

Objectives

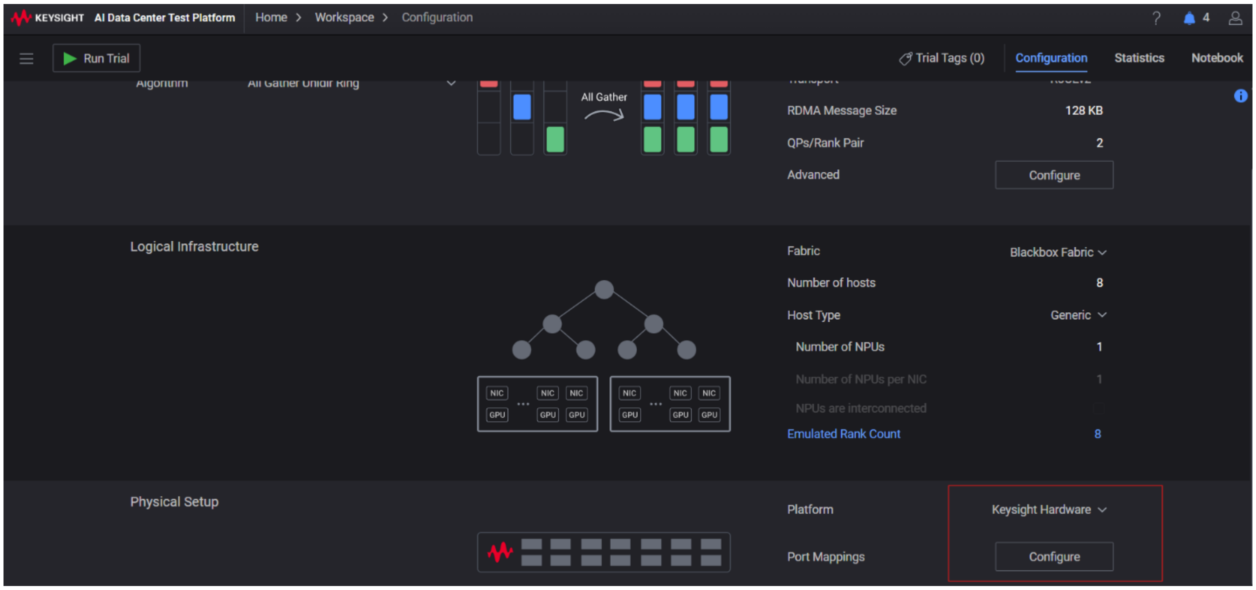
1. Distribute packets across multiple paths.
2. Observe links usage.
3. Measure packet loss / drop rates.

Setup

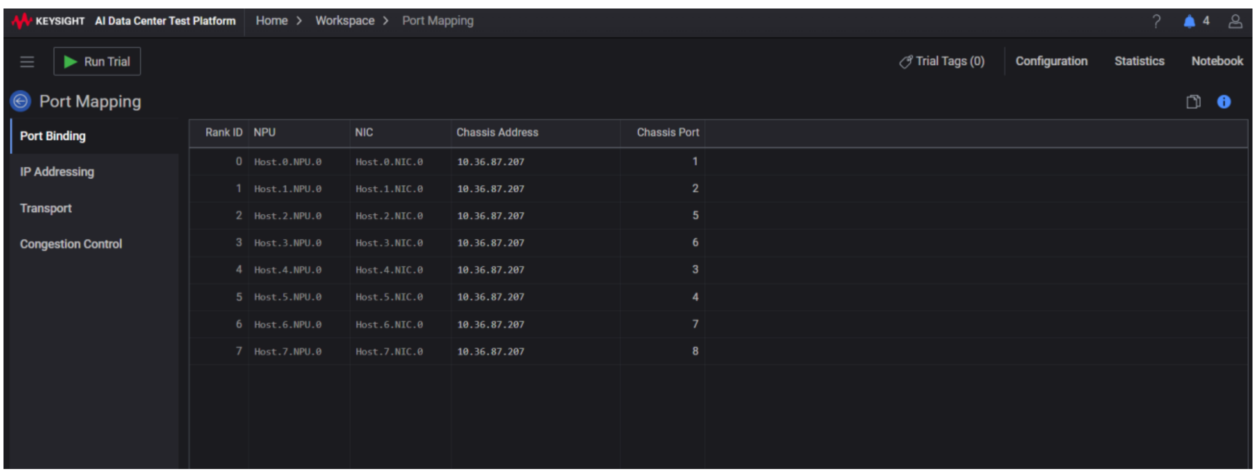


Step-by-step instructions

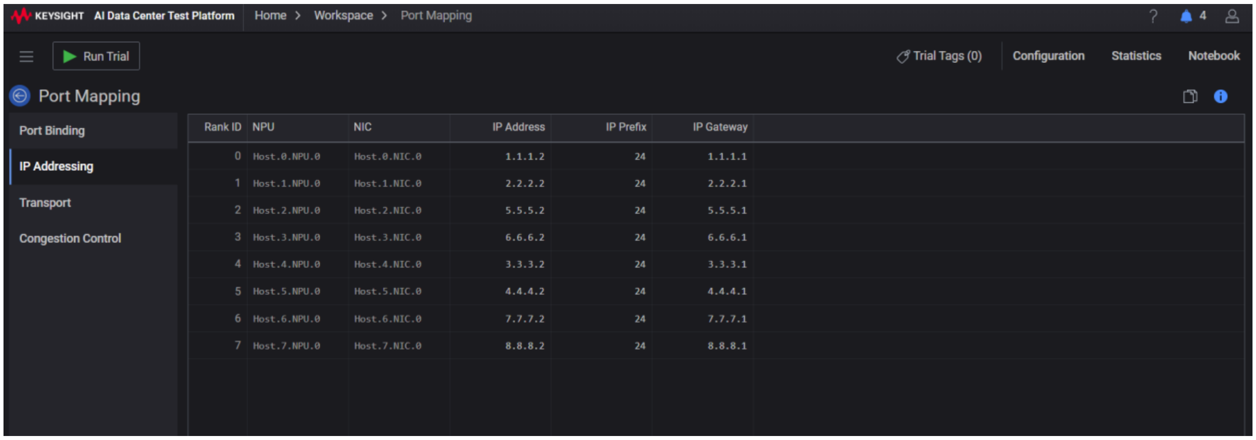
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



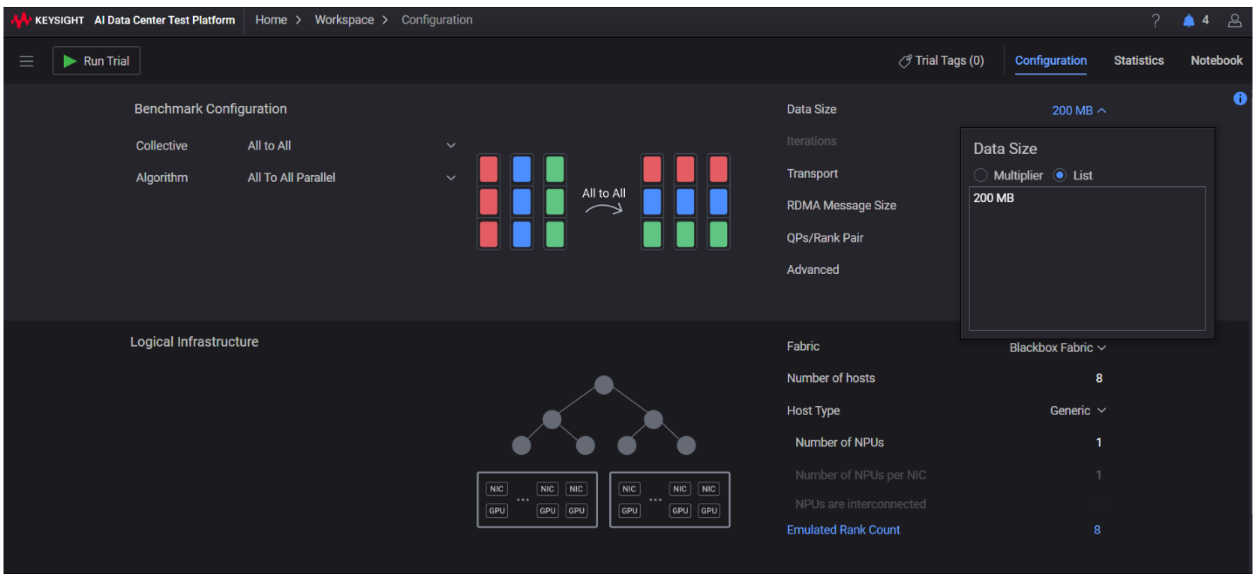
2. Set the port binding.



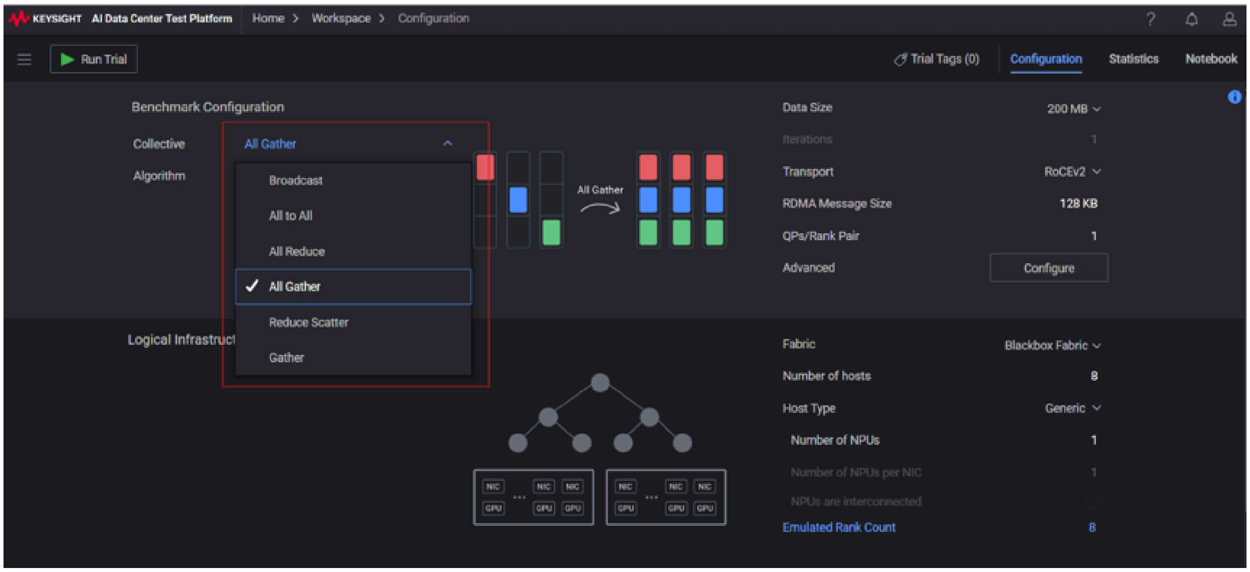
3. Set the IP address and gateway.



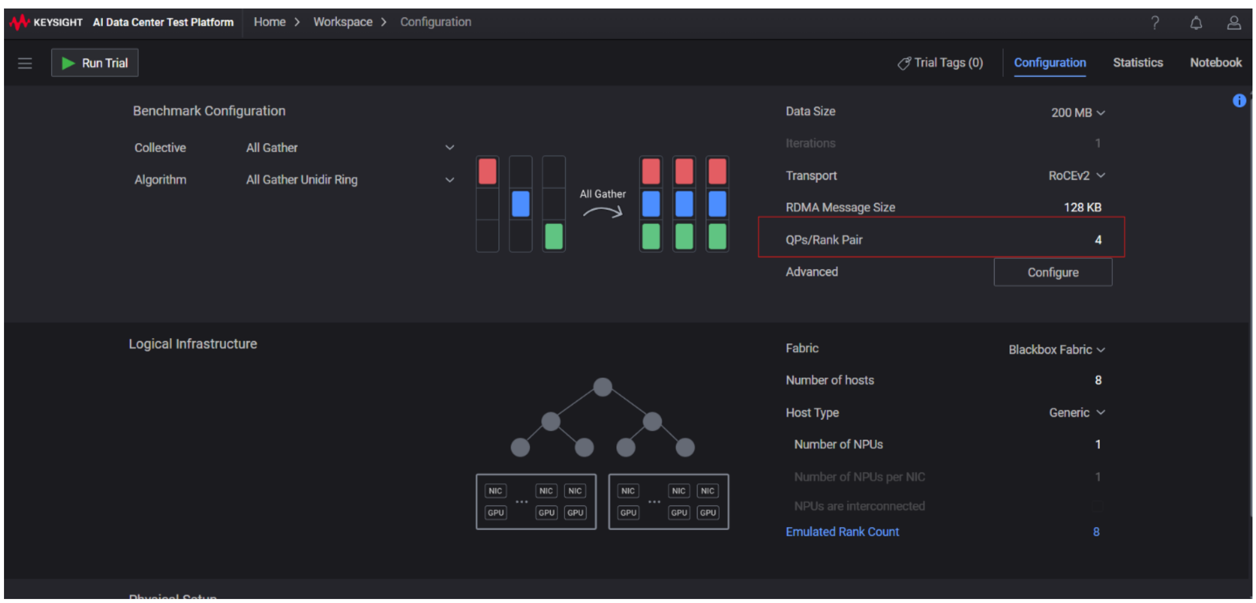
4. Set the data size.



5. Set the collective and algorithms.

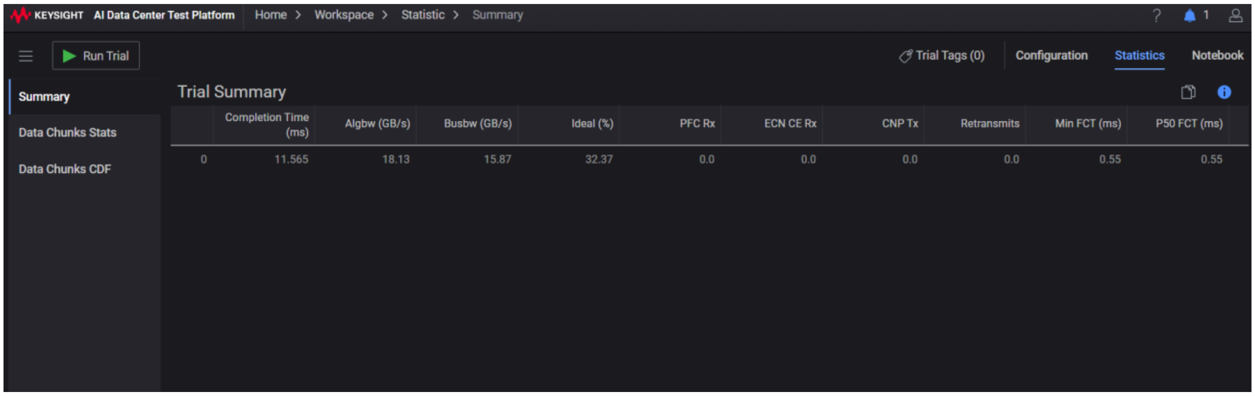


6. Set the QPs / Rank Pair.



7. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.



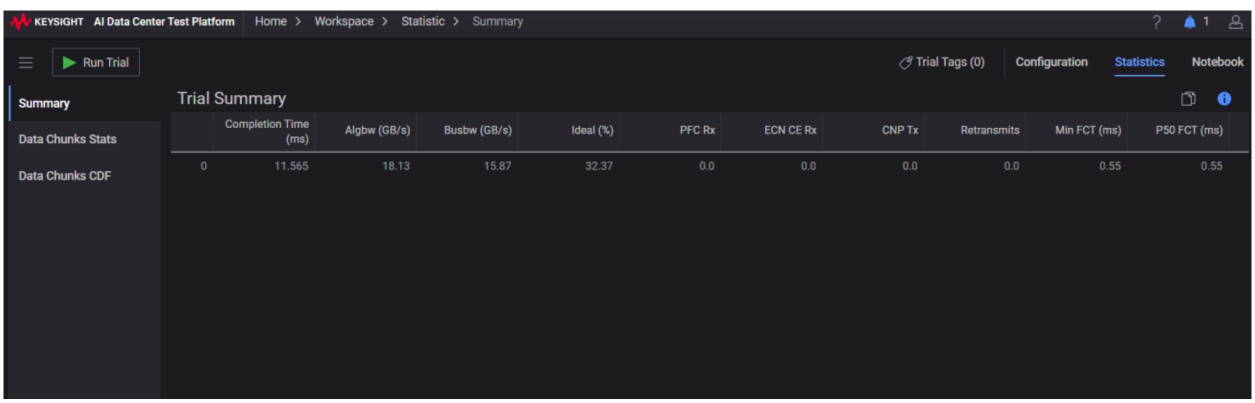
Test variables

- All-reduce-ring
- Parallel-QPs

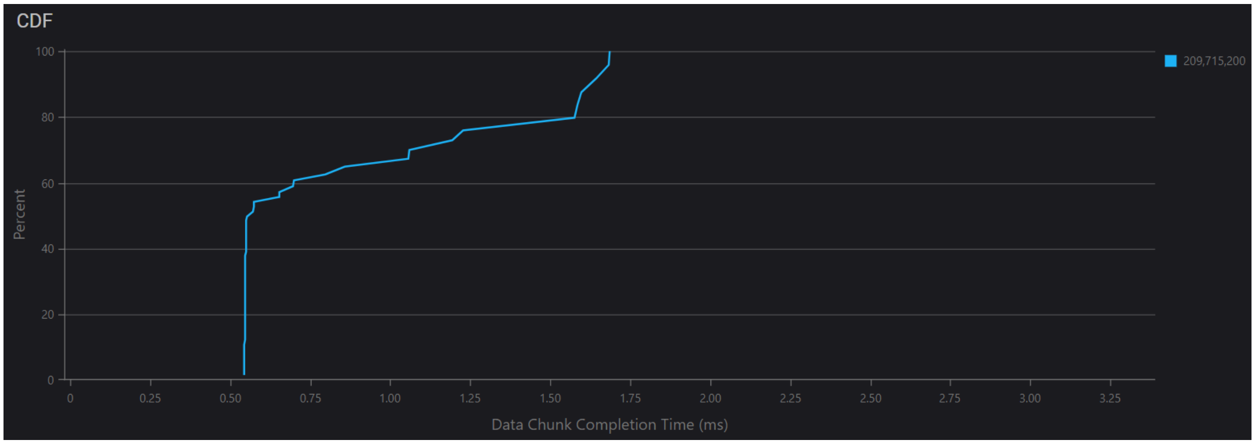
  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

Results analysis

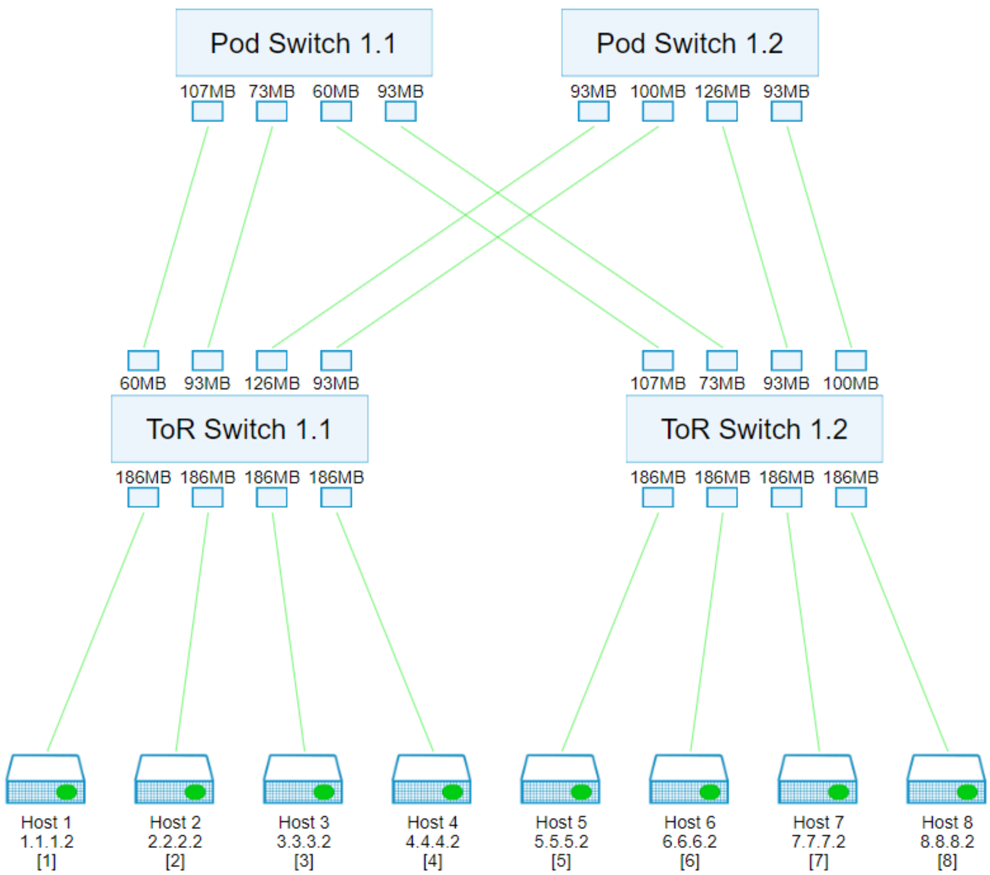
The training job was completed with a suboptimal algorithm bandwidth, suggesting long tail latency.



To mitigate congestion, the fabric requires additional QPs to rebalance hashing at the expense of long tail latency.



Bytes count



Egress QPs for switches

Switch	Egress QPs	Link usage (%)	Distribution	Frames delta
ToR Switch 1.1	14 * 4	100	9 + 14 + 19 + 14	0
ToR Switch 1.2	14 * 4	100	16 + 11 + 14 + 15	0

Conclusion

Collective operations are being run by using parallel QPs to ensure adequate routing entropy, at the expense of increased QP creation.

Test case 4: All-gather-ring with spray

Overview

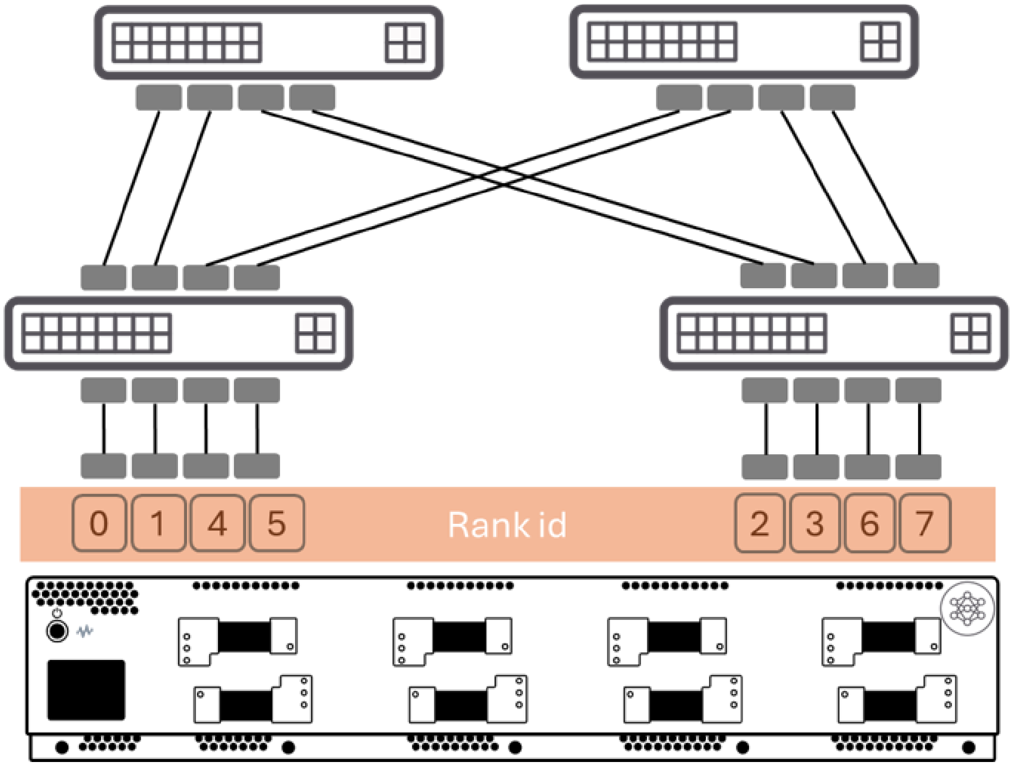
This test case employs an all-gather-ring collective operation to evaluate the ECMP performance of the fabric. The all-gather ring with traffic flowing in a circular pattern from node 0 to 7, requiring 56 QP instances for a single operation and 186 MB of memory per data chunk. This translates to an approximately 26.6 MB data size for a single QP during transmission. Meanwhile, the SUT consists of ToR switches, interconnected with multiple pod switches, forming redundant paths by using the Random Spray ECMP mode.

- QPs = 56
- 186 MB for every chunk, 186 / 7 = 26.6 M for single QP

Objectives

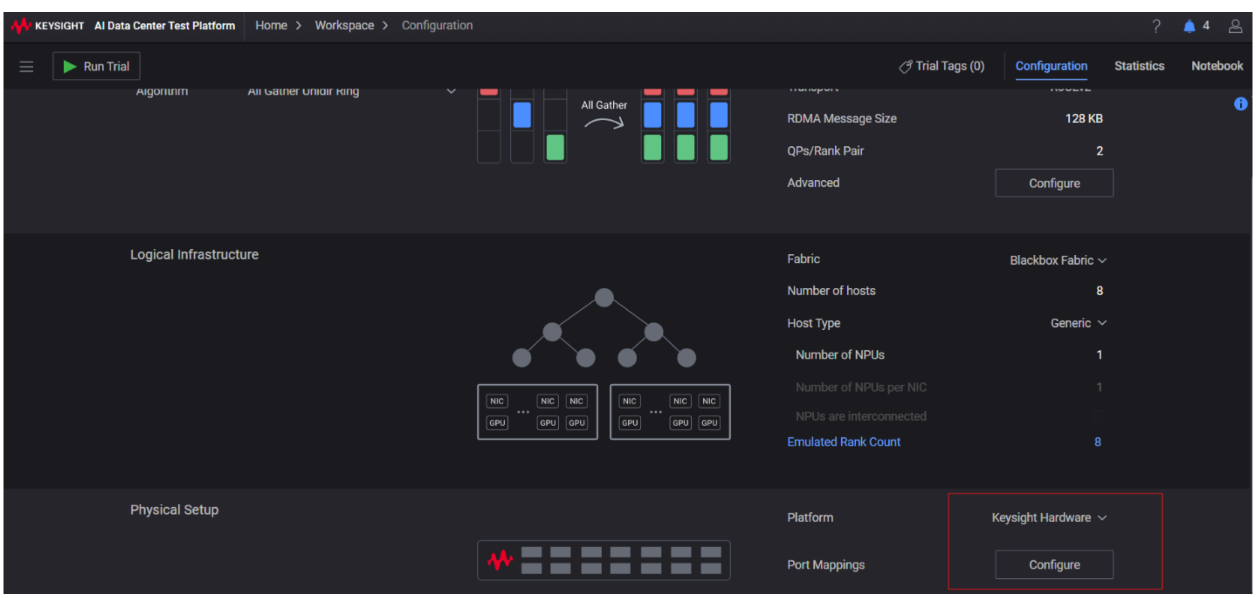
1. Distribute packets across multiple paths.
2. Observe links usage.
3. Measure packet loss / drop rates.

Setup

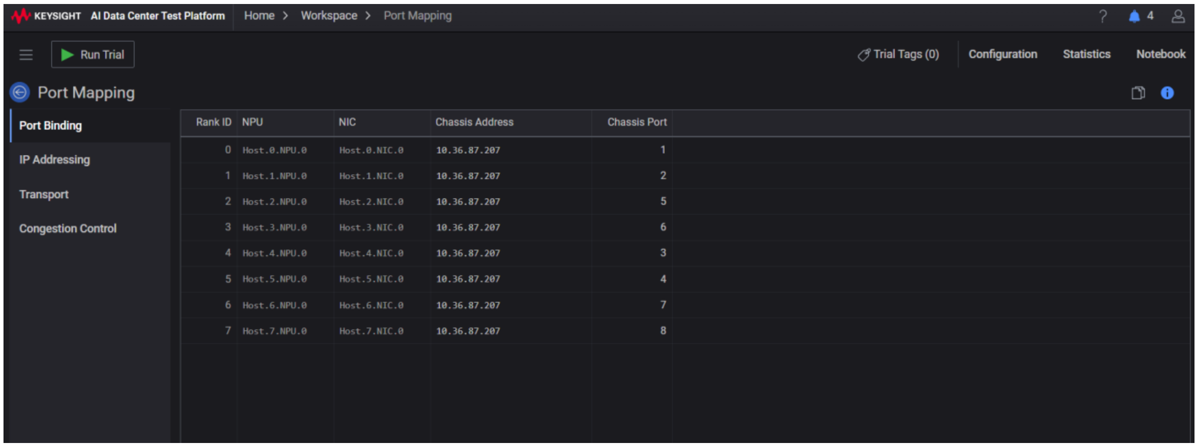


Step-by-step instructions

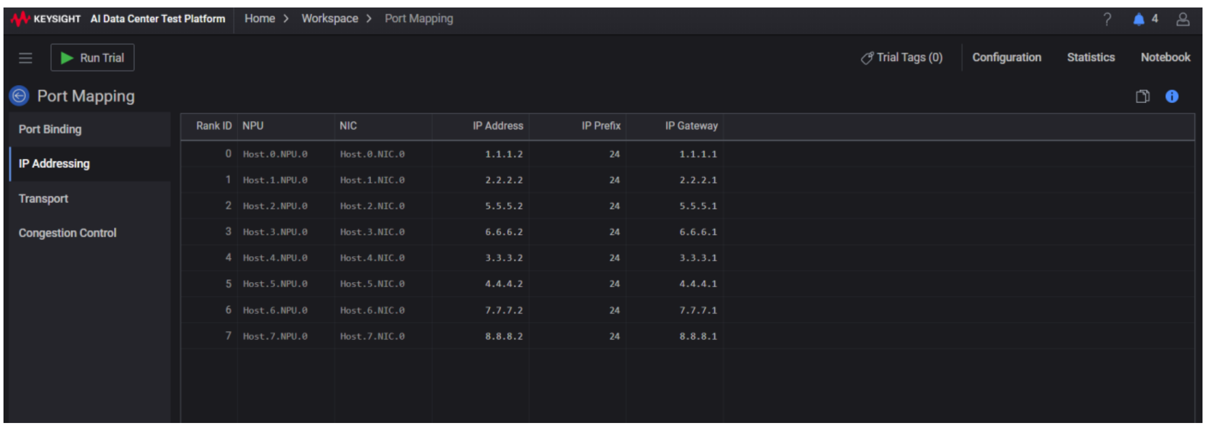
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



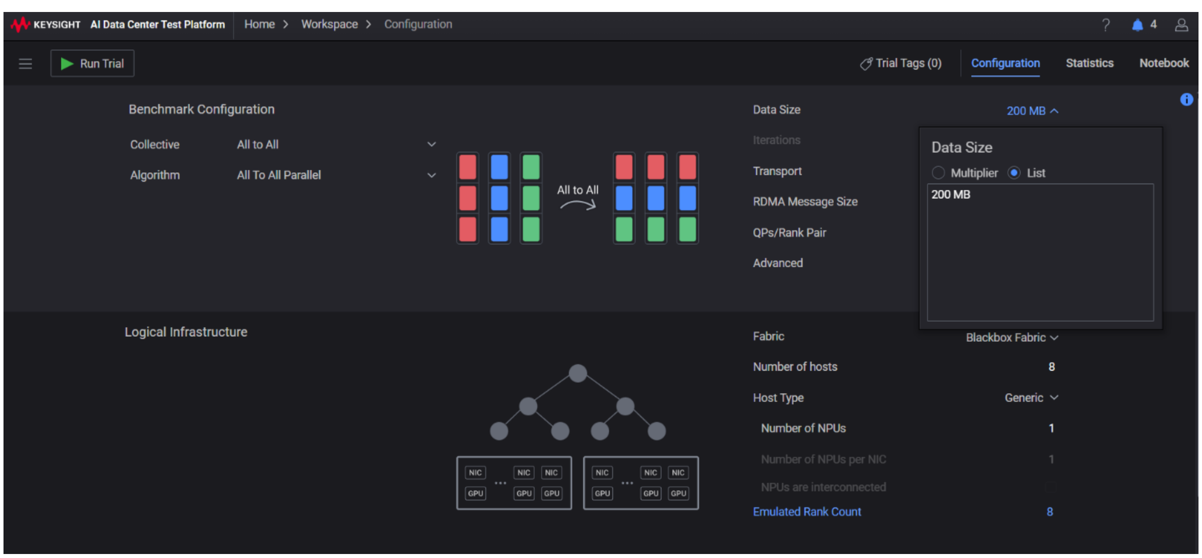
2. Set the port binding.



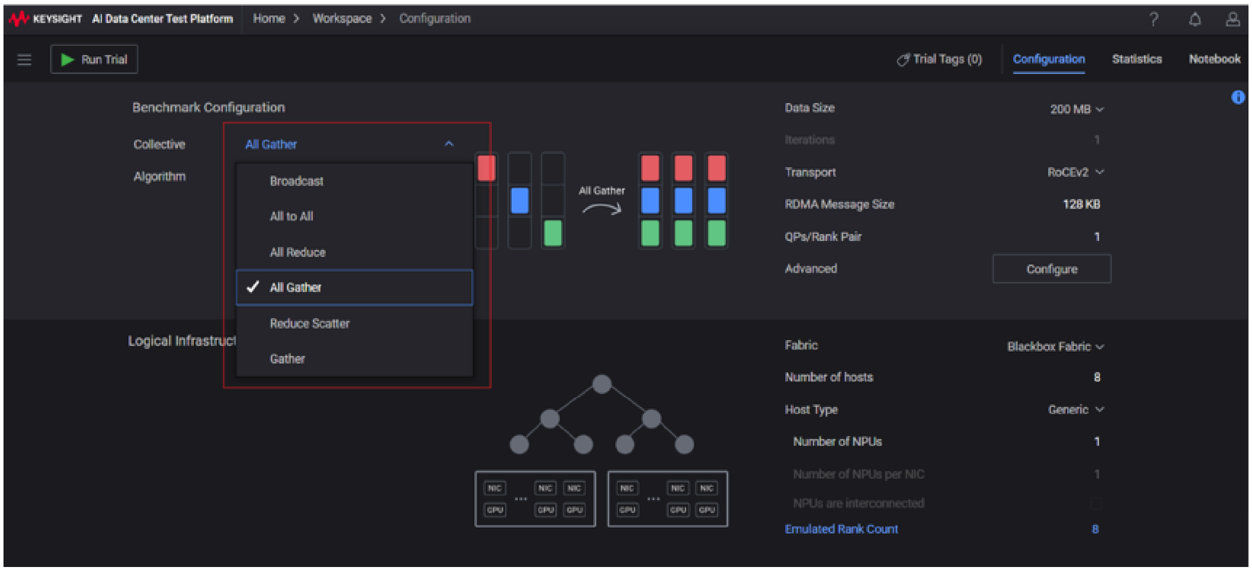
3. Set the IP address and gateway.



4. Set the data size.

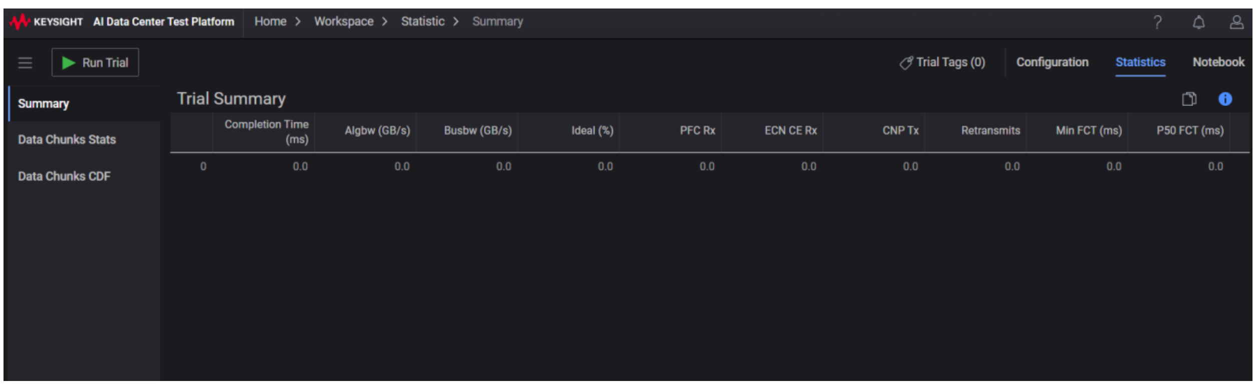


5. Set the collective and algorithms.



6. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.



Test variables

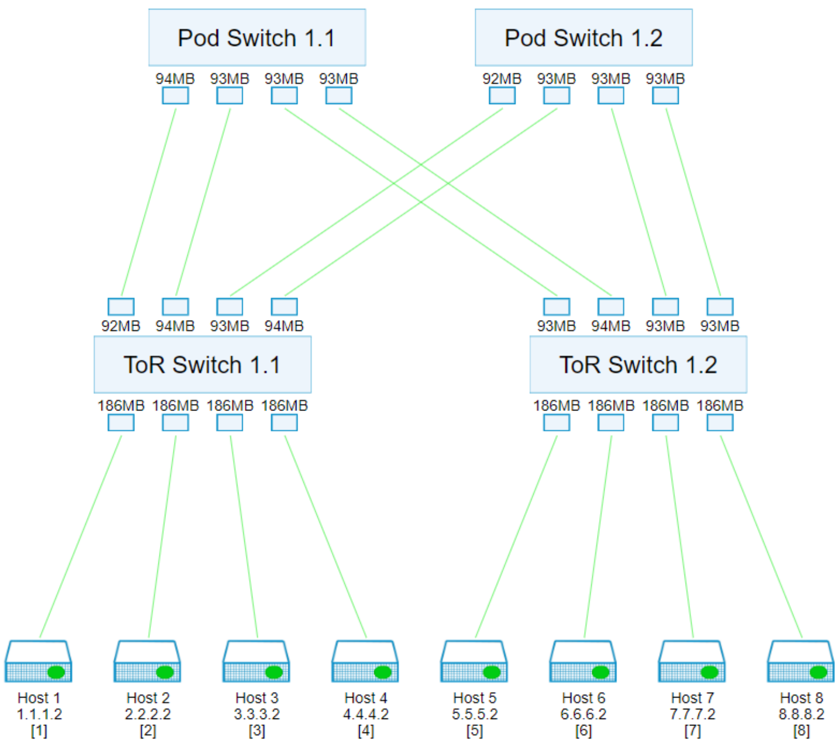
- All-reduce-ring
- Adaptive routing

  Enable the use of adaptive routing capable data transfers for the IB verbs transport. Adaptive routing can improve the performance of communications at scale.

Results analysis

The training job is likely to fail because of reordered packet caused by the fabric's random spray ECMP mode.

Bytes count



The egress transmits packets use approximately 3.5 queue pairs, indicating that the hashing is not flow-based but sprays packets evenly across the egress port.

Conclusion

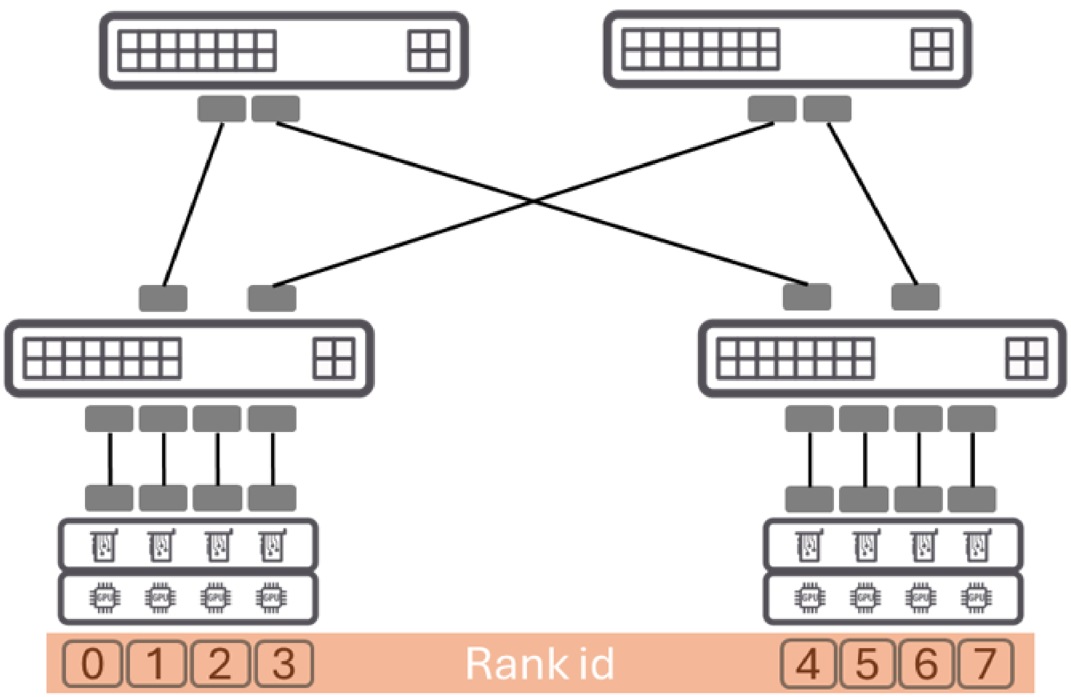
Spray ECMP mode is effective in resolving congestion issues but requires the reordering capability to be enabled at the NIC level.

Test Methodology for Congestion Control

AI fabric is a high-performance interconnect technology designed for accelerating AI, machine learning, and deep learning workloads. To achieve its performance goals, AI fabric relies heavily on lossless RDMA.

As AI workloads become increasingly complex, they require more processing power and memory. AI systems require high reliability to ensure accurate model training and deployment. Lossless RDMA guarantees that data is reliably delivered, reducing the risk of errors or data loss during transmission.

In this experiment, we emulate a scenario featuring two NPU hosts, each equipped with four NPU and NICs. We employ an oversubscription fabric with congestion, enabling us to validate the congestion control enablement and benchmark the maximum performance of the collective operations under congestion.



Perform the following test procedure:

1. Set the peering test ports with IP and gateway.
2. Set the rank id as [0, 1, 2, 3, 4, 5, 6, 7].
3. Set the data size.
4. Set the algorithm.
5. Run trials.

The following test cases demonstrate the test methodologies and detailed step-by-step configuration with automated test packages available in CB.

Test collectives	Data size	Write msg size	Parallel QPs	Congestion control	PFC headroom / ECN min threshold
All to all	16 MB-8 GB	128 KB	1	PFC	100000
All to all	16 MB-8 GB	128KB	1	ECN	10000
All to all	16 MB-8 GB	128KB	1	PFC+ECN	100000
Gather	16 MB-8 GB	128KB	1	PFC	100000

Test case 1: JCT with PFC only

Overview

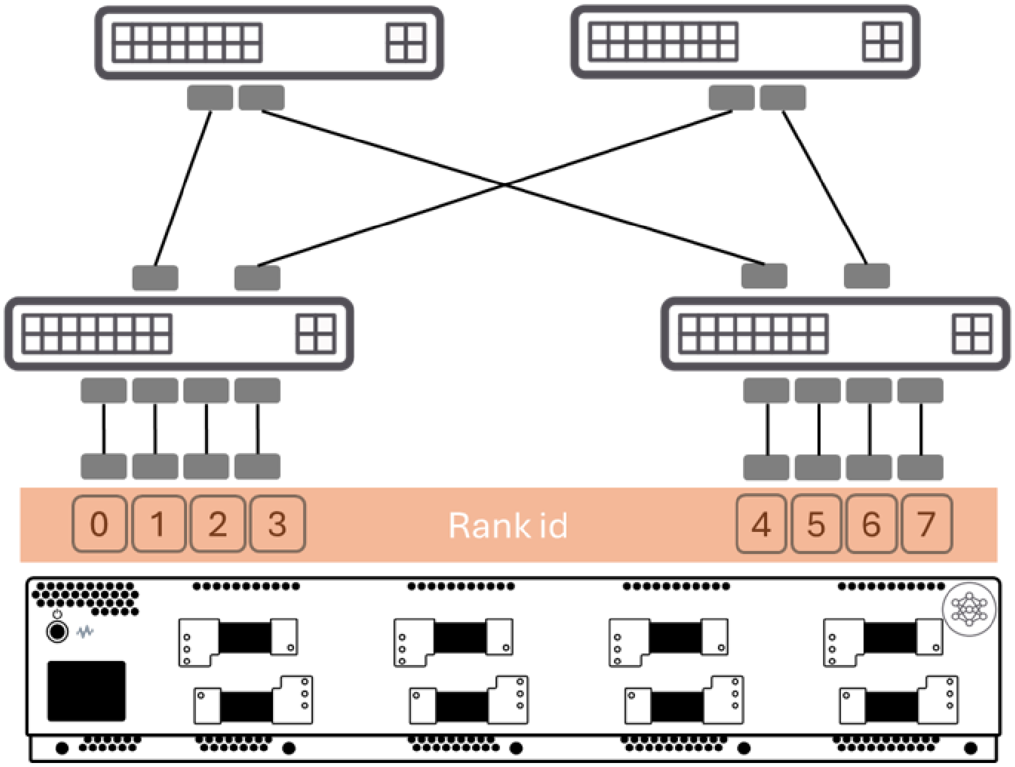
PFC is a flow control algorithm designed to manage network bandwidth and prevent congestion. It prioritizes traffic based on importance, ensuring that critical packets (like RDMA requests) are delivered efficiently while less important packets (like Acknowledgment (ACKs) or Negative Acknowledgment (NAKs)) can be delayed or dropped, if necessary. PFC categorizes incoming traffic into different priority levels based on its importance.

This test case employs an all-to-all collective operation to evaluate the performance of the fabric with PFC enabled.

Objectives

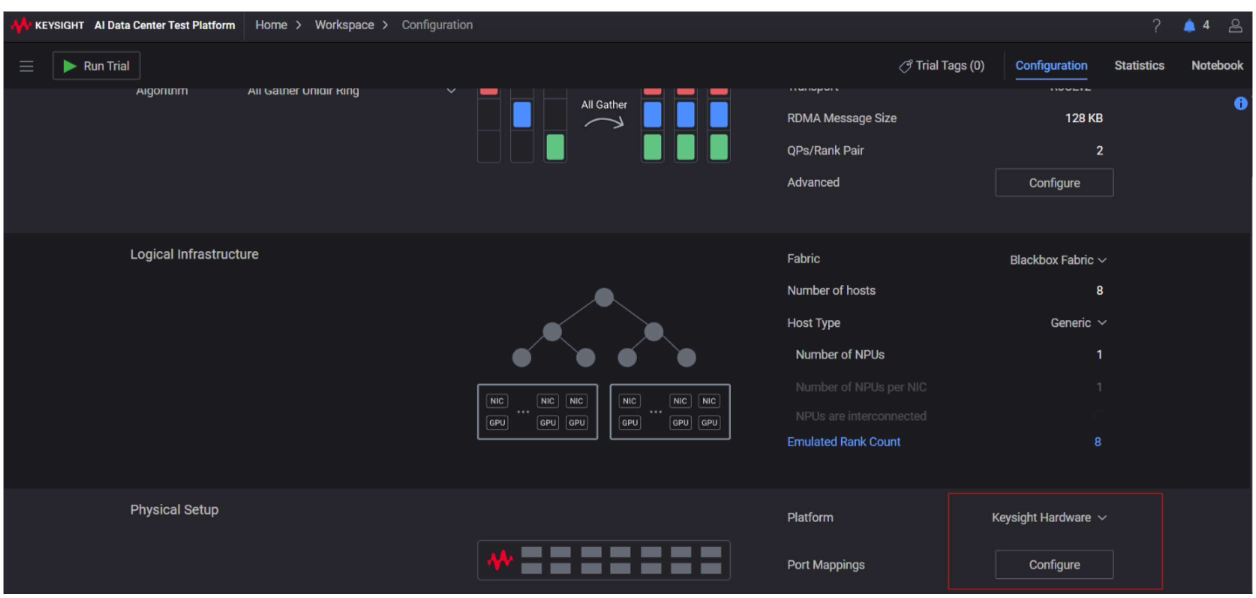
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Verify PFC dynamically adjusts the available network bandwidth to accommodate the priority traffic, preventing congestion.

Setup

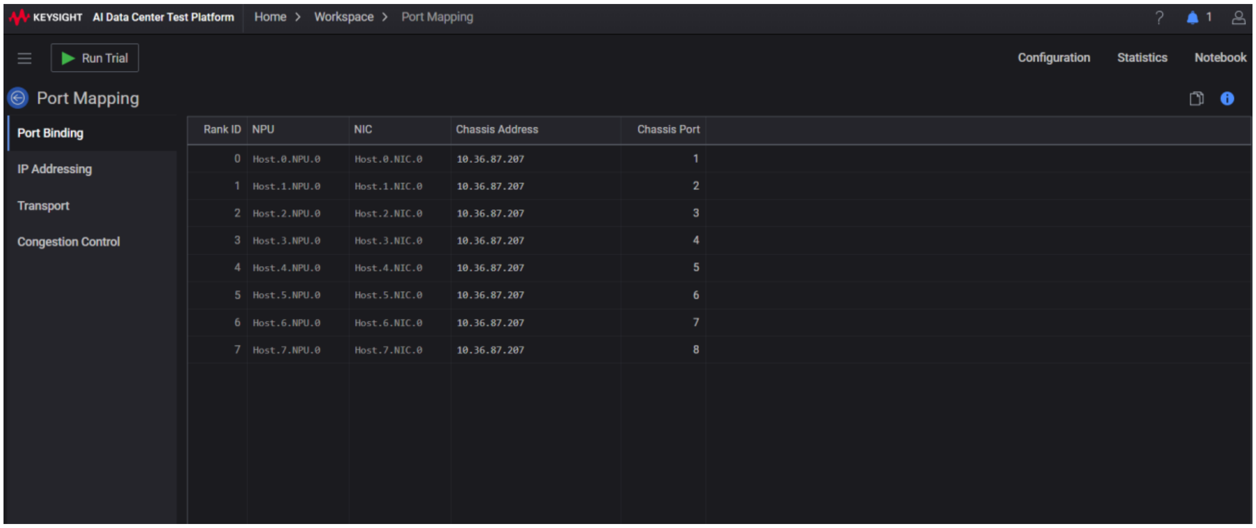


Step-by-step instructions

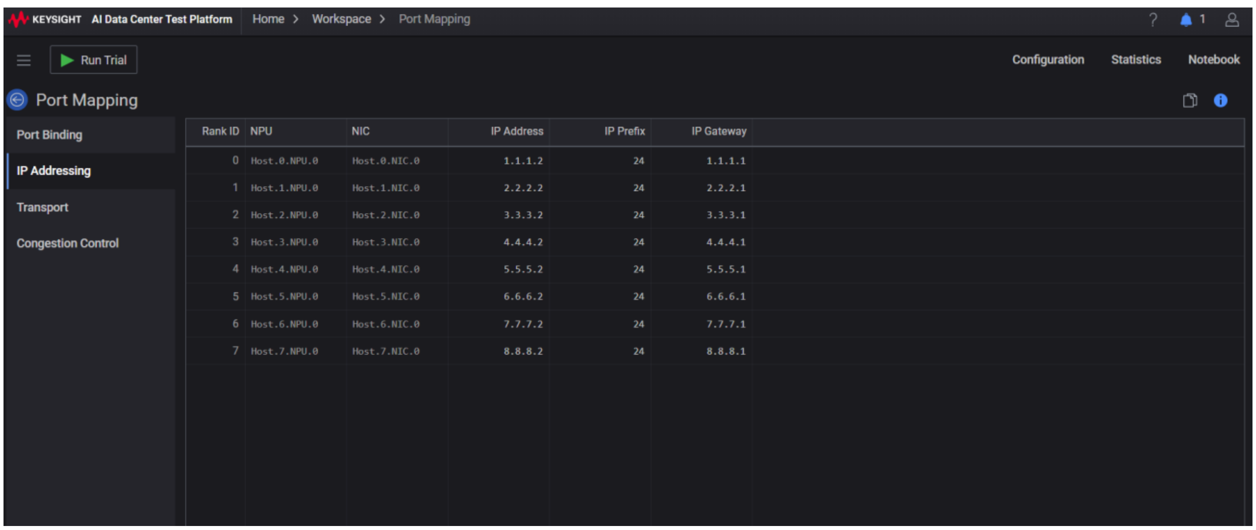
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



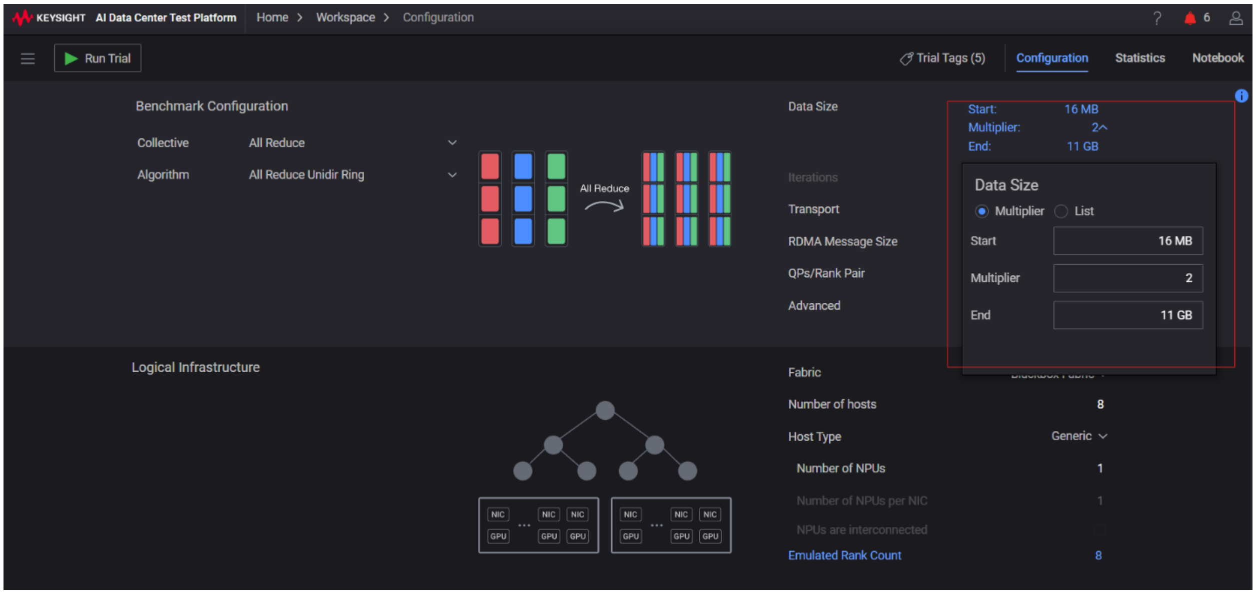
2. Set the port binding.



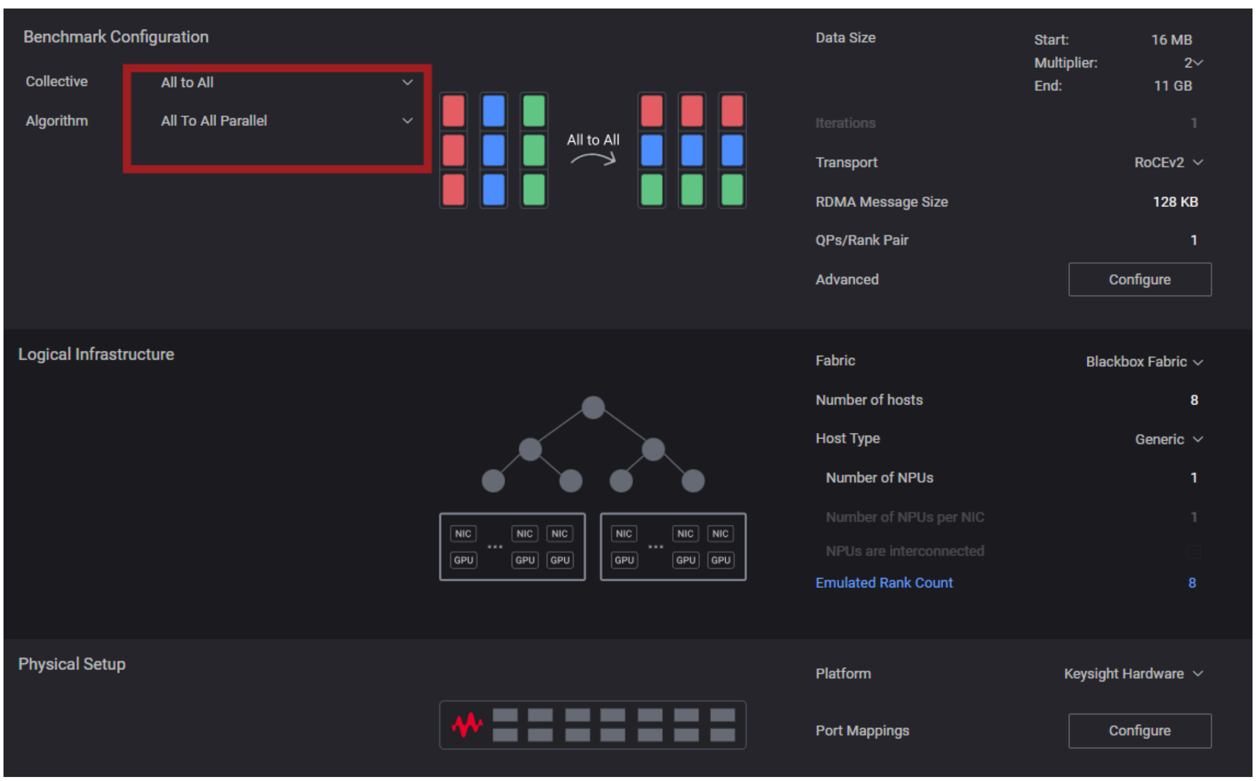
3. Set the IP address and gateway.



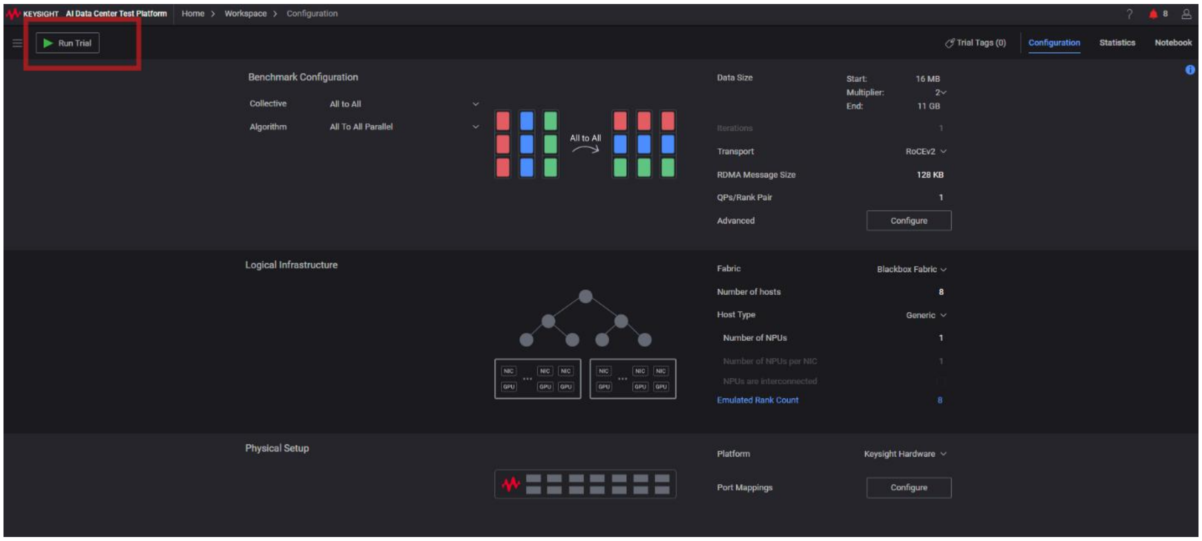
4. Set the data size from 16MB to 11GB with a multiplier of 2.



5. Set the collective and algorithms.



6. Select **Run Trial**.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other ring algorithms to test: All-Gather-Ring and Reduce-Scatter-Ring.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

- RDMA Write message size

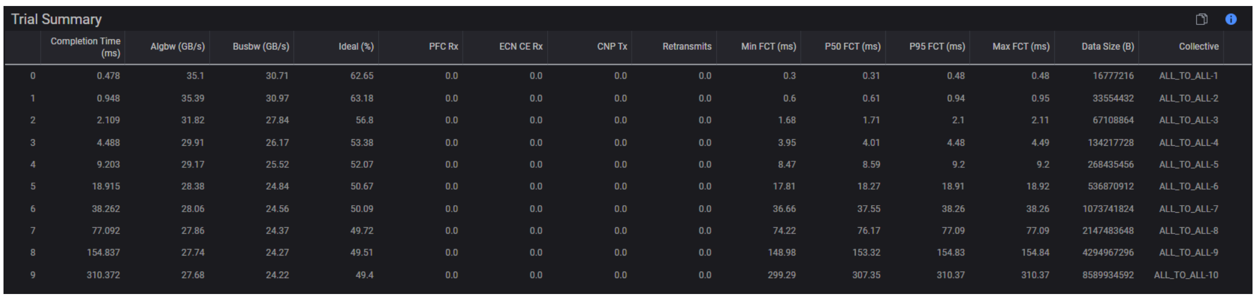
  Within a single QP in collectives, multiple RDMA Write operations are performed, with each operation split into separate chunks based on the RDMA write message size for the data chunk. Within these RDMA Write operations, each packet is defined by its corresponding RDMA MTU.

- Parallel-QPs

  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

Results analysis

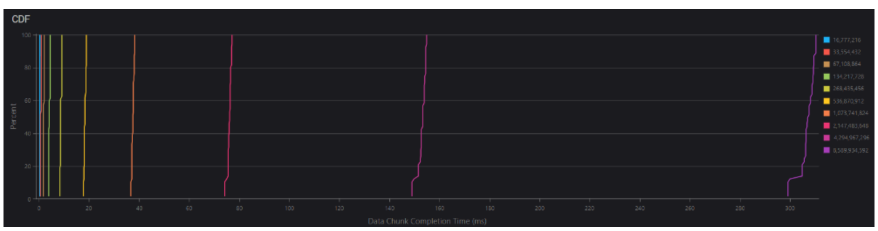
- Completion time



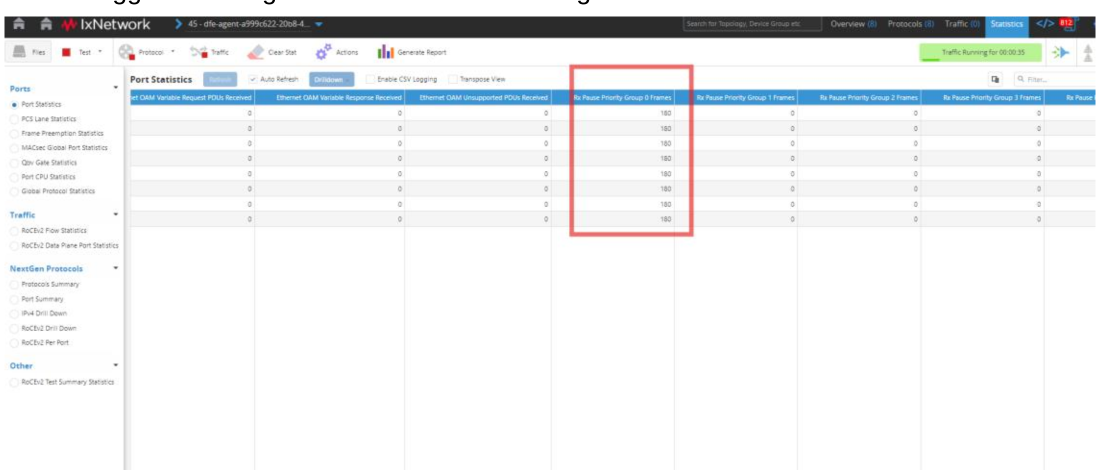
Trial Summary

	Completion Time (ms)	Algbw (GB/s)	Busbw (GB/s)	Ideal (%)	PFC Rx	ECN CE Rx	CNP Tx	Retransmits	Min FCT (ms)	P50 FCT (ms)	P95 FCT (ms)	Max FCT (ms)	Data Size (B)	Collective
0	0.478	35.1	30.71	62.65	0.0	0.0	0.0	0.0	0.3	0.31	0.48	0.48	16777216	ALL_TO_ALL-1
1	0.948	35.39	30.97	63.18	0.0	0.0	0.0	0.0	0.6	0.61	0.94	0.95	33554432	ALL_TO_ALL-2
2	2.109	31.82	27.84	56.8	0.0	0.0	0.0	0.0	1.68	1.71	2.1	2.11	67108864	ALL_TO_ALL-3
3	4.488	29.91	26.17	53.38	0.0	0.0	0.0	0.0	3.95	4.01	4.48	4.49	134217728	ALL_TO_ALL-4
4	9.203	29.17	25.52	52.07	0.0	0.0	0.0	0.0	8.47	8.59	9.2	9.2	268435456	ALL_TO_ALL-5
5	18.915	28.38	24.84	50.67	0.0	0.0	0.0	0.0	17.81	18.27	18.91	18.92	536870912	ALL_TO_ALL-6
6	38.262	28.06	24.56	50.09	0.0	0.0	0.0	0.0	36.66	37.55	38.26	38.26	1073741824	ALL_TO_ALL-7
7	77.092	27.86	24.37	49.72	0.0	0.0	0.0	0.0	74.22	76.17	77.09	77.09	2147483648	ALL_TO_ALL-8
8	154.837	27.74	24.27	49.51	0.0	0.0	0.0	0.0	148.98	153.32	154.83	154.84	4294967296	ALL_TO_ALL-9
9	310.372	27.68	24.22	49.4	0.0	0.0	0.0	0.0	299.29	307.35	310.37	310.37	8589934592	ALL_TO_ALL-10

- Long-tail latency is observed



- PFC is triggered during tests with a sense of congestion



Conclusion

PFC effectively manages the congestion. Operations run smoothly across various data sizes. Large data sizes result in lower bus bandwidth and long tail latency.

Test case 2: JCT with ECN only

Overview

Data Center Quantized Congestion Notification (DCQCN) is a congestion control algorithm designed specifically for AI fabric, which is optimized for AI workloads. Its dynamic adjustment mechanism helps prevent network congestion, reduces latency, and improves overall network usage. By leveraging DCQCN, AI fabric provides a more efficient and scalable infrastructure for AI applications.

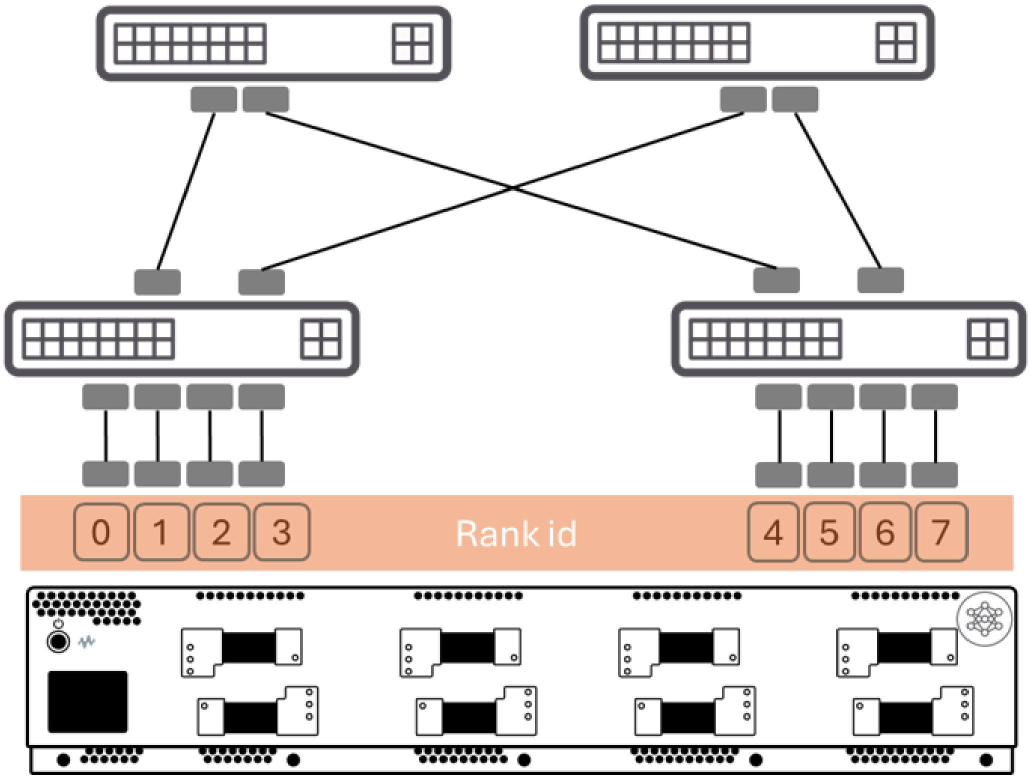
When a network device detects congestion, it sets the ECN field in the IP header of packets being sent through the congested link. The receiver (for example, a server) examines the ECN field and notifies the sender that congestion has occurred. Based on the notification, the sender adjusts its sending rate to prevent further congestion.

This test case employs an all-to-all collective operation to evaluate the performance of the fabric with ECN enabled.

Objectives

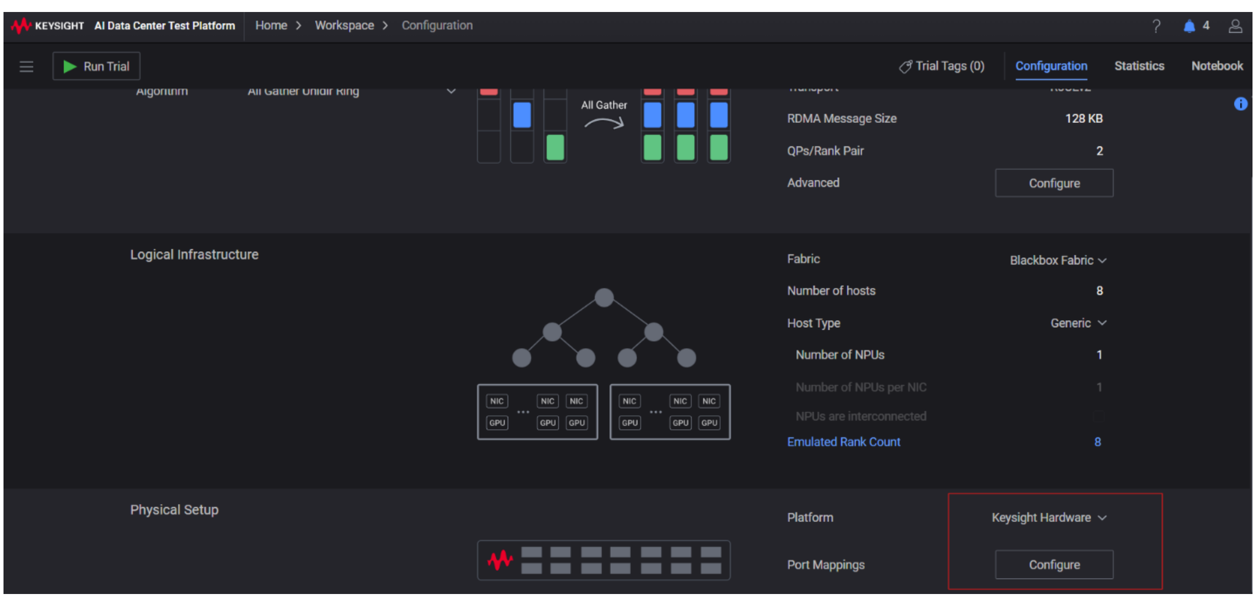
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Verify ECN dynamically adjusts the available network bandwidth to accommodate the priority traffic, preventing congestion.

Setup

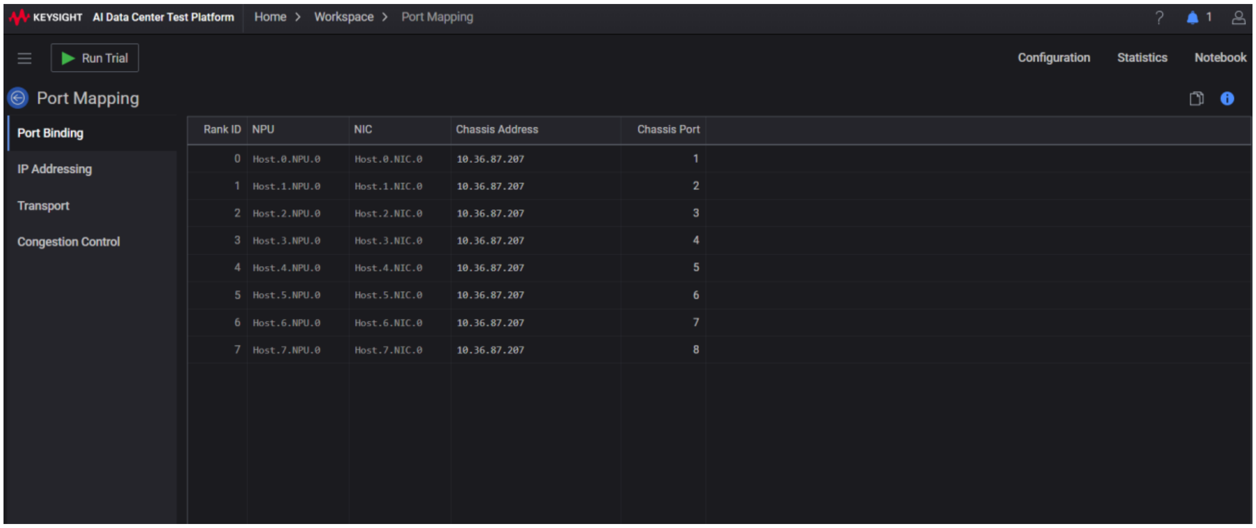


Step-by-step instructions

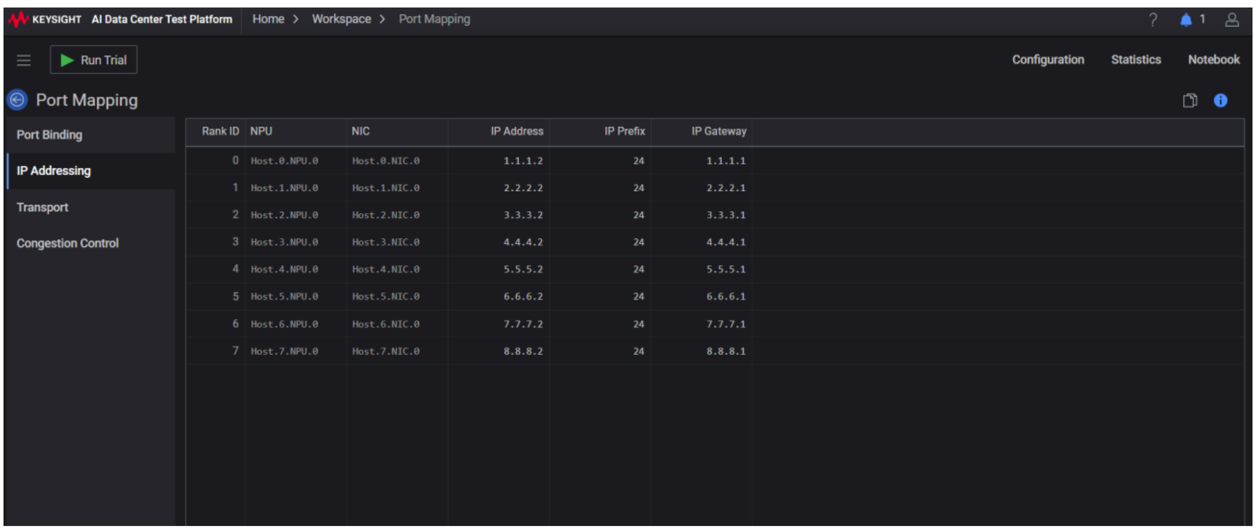
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



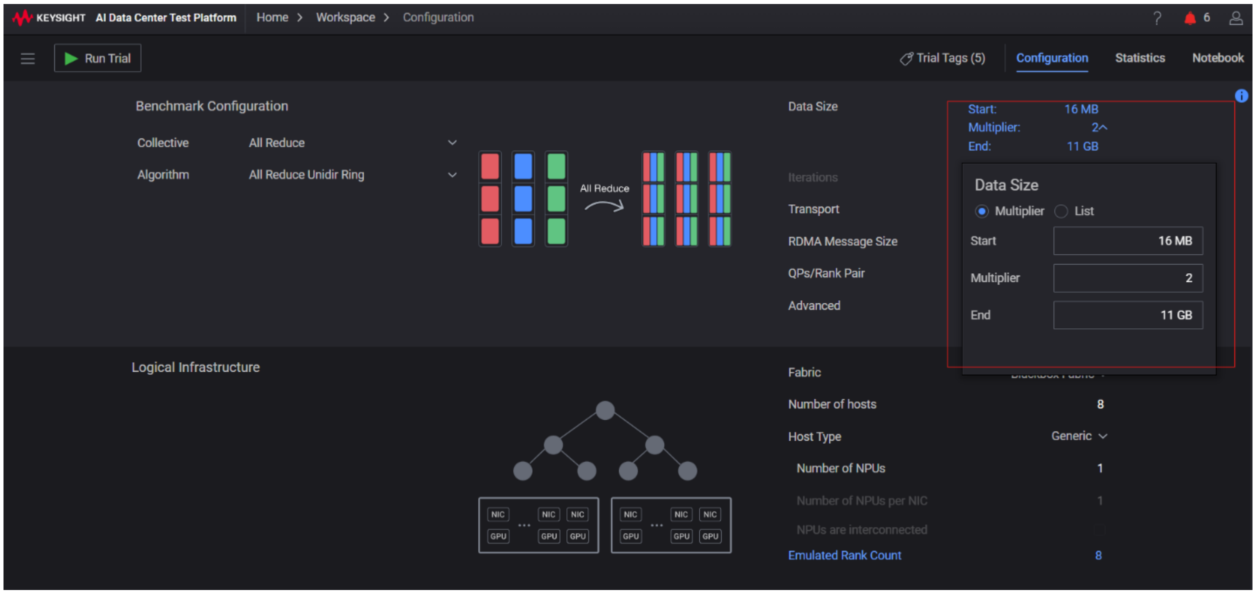
2. Set the port binding.



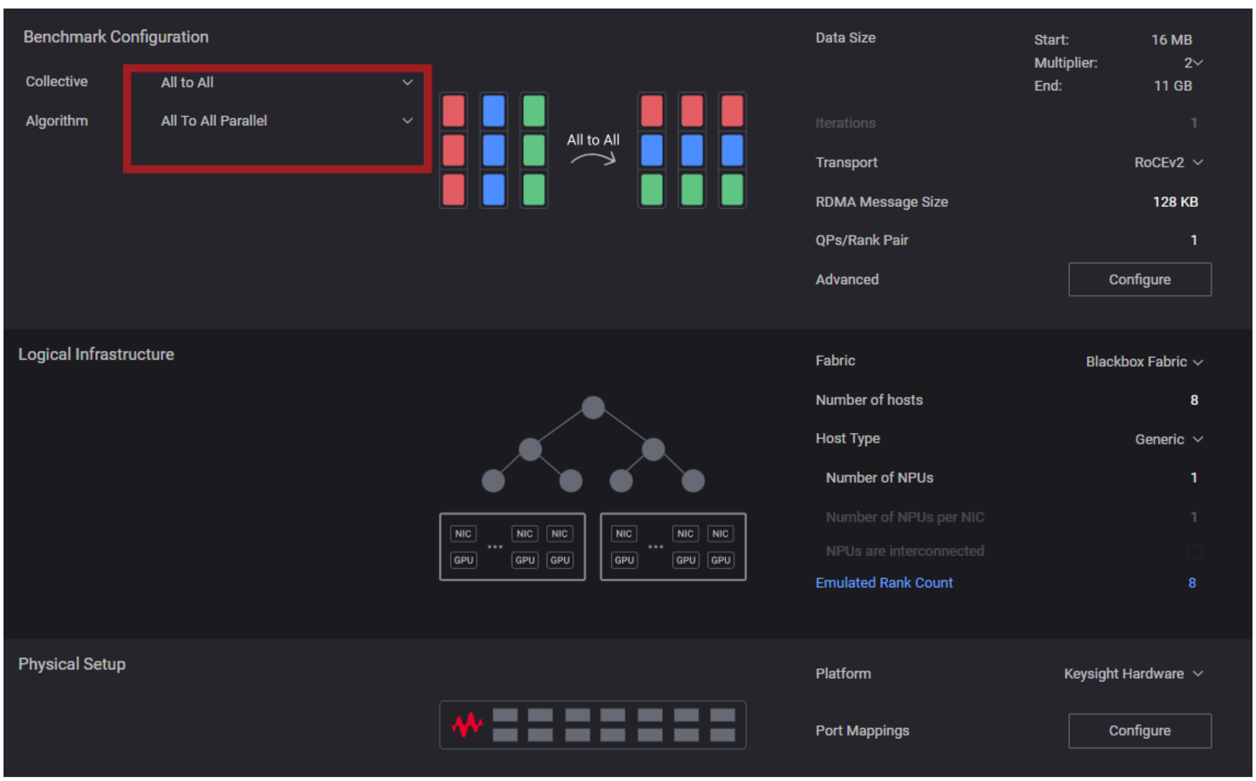
3. Set the IP address and gateway.



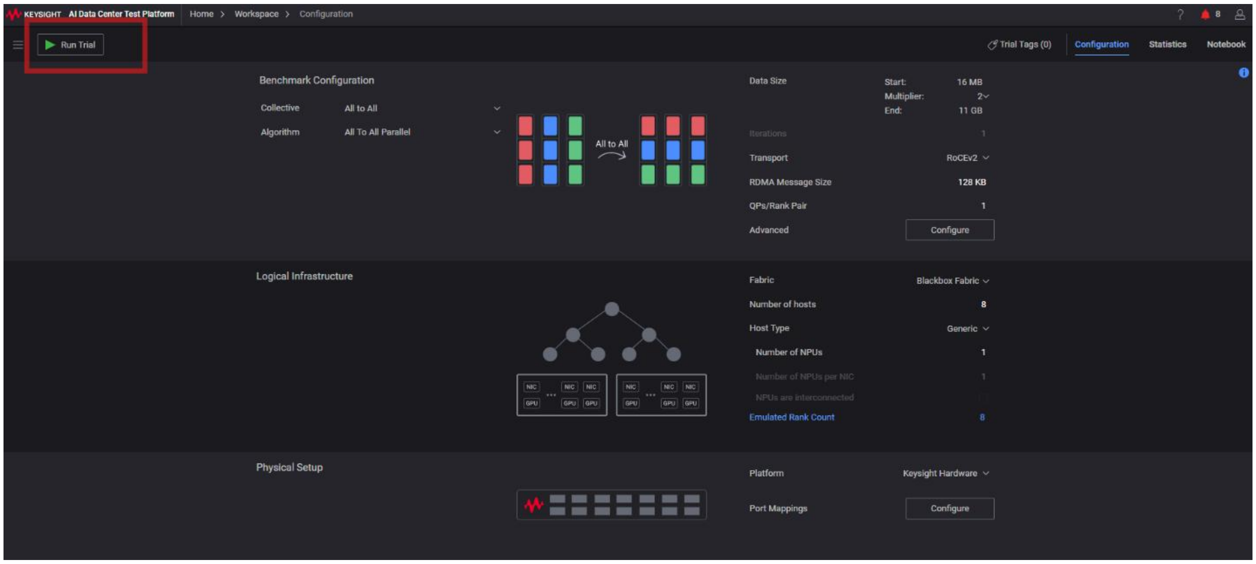
4. Set the data size from 16 MB to 11 GB with a multiplier of 2.



5. Set the collective and algorithms.



6. Select **Run Trial**.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an 'active' MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an 'active MTU.'

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other ring algorithms to test: All-Gather-Ring and Reduce-Scatter-Ring.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

- RDMA Write message size

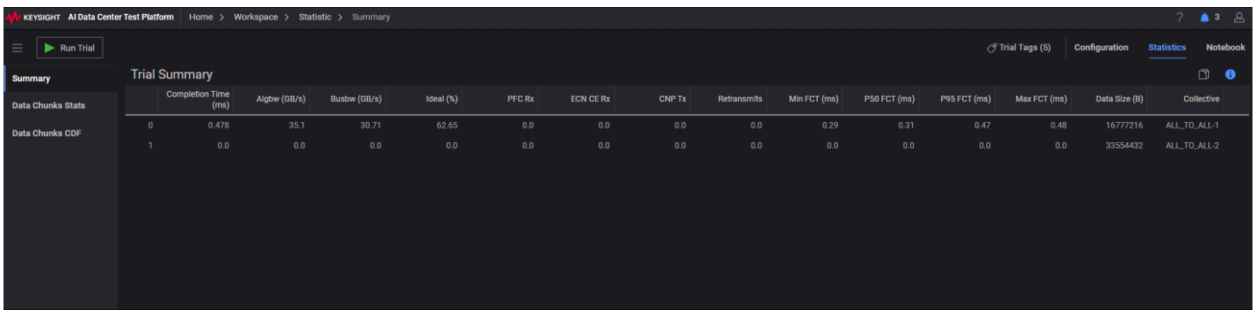
  Within a single QP in collectives, multiple RDMA Write operations are performed, with each operation split into separate chunks based on the RDMA write message size for the data chunk. Within these RDMA Write operations, each packet is defined by its corresponding RDMA MTU.

- Parallel-QPs

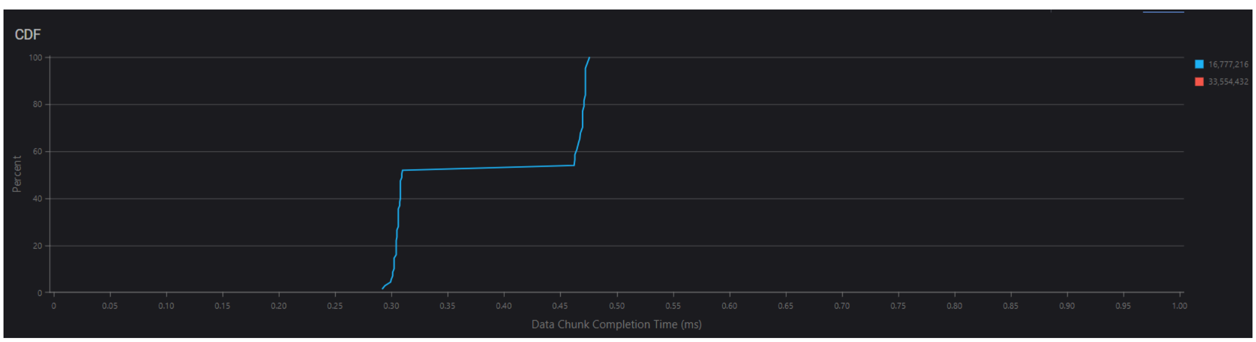
  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

Results analysis

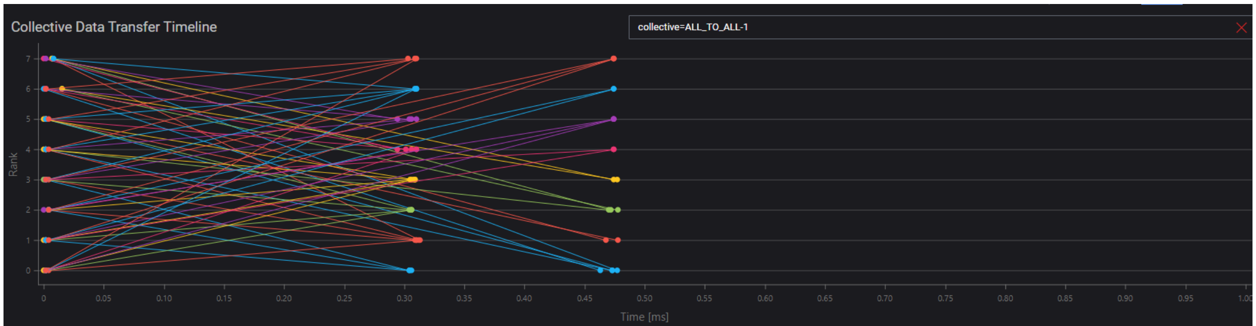
- Completion **time**



- Long-tail latency is observed



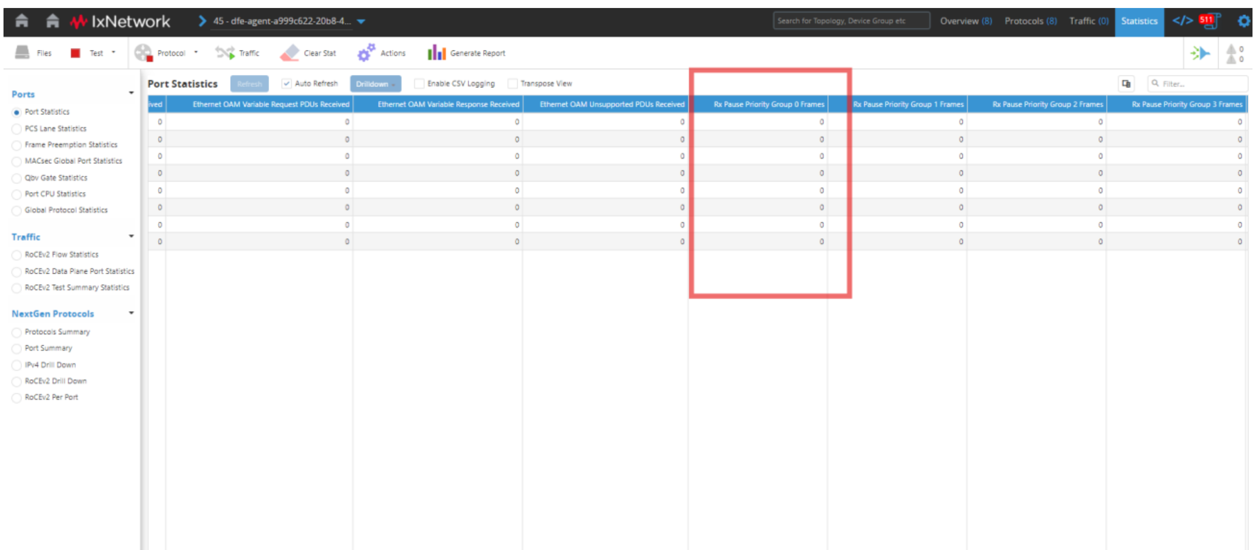
- ECN is marking, and CNP is transmitted between the receiver and sender



Data Chunks Table

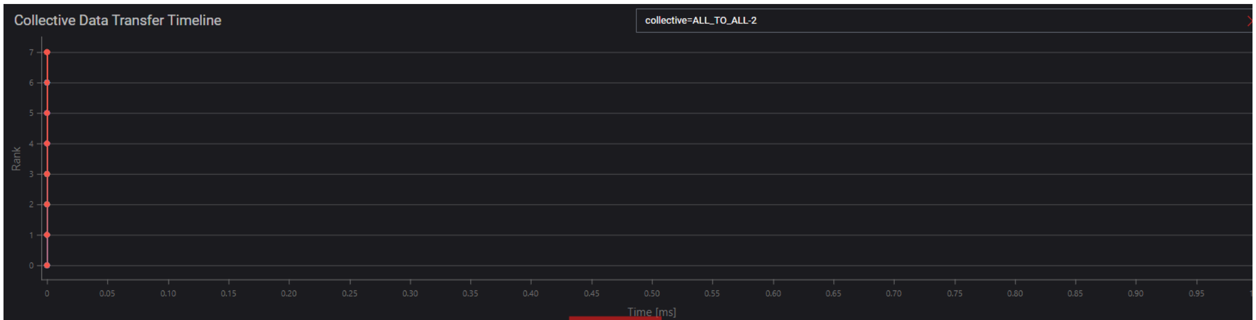
Index	Completion Time (ms)	Start (ms)	End (ms)	Total Bytes Tx	Frames Tx	Frames Rx	Frames Delta	Min Latency (ns)	Avg Latency (ns)	Max Latency (ns)	Ecn-CE Rx	Cnp Tx	Cnp Rx	Total Bytes Rx
1	0.304	0.002	0.306	2129152	512.0	512.0	0.0	792.0	1,040.0	3,622.0	6.0	1.0	1.0	2129152
2	0.3	0.004	0.304	2129152	512.0	512.0	0.0	790.0	1,084.0	3,845.0	12.0	1.0	1.0	2129152
56	0.301	0.002	0.303	2129152	512.0	512.0	0.0	807.0	1,250.0	4,595.0	35.0	4.0	4.0	2129152
15	0.305	0.004	0.309	2129152	512.0	512.0	0.0	790.0	1,555.0	4,352.0	54.0	5.0	5.0	2129152
26	0.47	0.004	0.474	2129152	512.0	512.0	0.0	3,985.0	65,998.0	126,645.0	508.0	8.0	8.0	2129152
25	0.472	0.002	0.474	2129152	512.0	512.0	0.0	2,410.0	66,070.0	126,970.0	506.0	8.0	8.0	2129152
24	0.472	0.002	0.474	2129152	512.0	512.0	0.0	2,850.0	66,477.0	125,722.0	508.0	8.0	8.0	2129152
23	0.472	0.002	0.474	2129152	512.0	512.0	0.0	2,947.0	66,075.0	126,802.0	507.0	8.0	8.0	2129152
22	0.308	0.0	0.308	2129152	512.0	512.0	0.0	797.0	1,754.0	6,692.0	86.0	4.0	4.0	2129152

- No PFC is triggered



ived	Ethernet OAM Variable Request PDUs Received	Ethernet OAM Variable Response Received	Ethernet OAM Unsupported PDUs Received	Rx Pause Priority Group 0 Frames	Rx Pause Priority Group 1 Frames	Rx Pause Priority Group 2 Frames	Rx Pause Priority Group 3 Frames
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- There are packet losses for a big data size



Data Chunks Table

Index	Completion Time (ms)	Start (ms)	End (ms)	Total Bytes Tx	Frames Tx	Frames Rx	Frames Delta ↓	Min Latency (ns)	Avg Latency (ns)	Max Latency (ns)	Ecn-CE Rx	Cnp Tx	Cnp Rx	Total Bytes Rx
86	0.0	0.0	0.0	4258304	1,024.0	916.0	108.0	5,580.0	152,809.0	280,407.0	915.0	16.0	16.0	3809176
60	0.0	0.0	0.0	4258304	1,024.0	916.0	108.0	2,730.0	149,232.0	281,015.0	910.0	16.0	16.0	3809176
62	0.0	0.0	0.0	4258304	1,024.0	921.0	103.0	5,687.0	153,676.0	281,175.0	921.0	16.0	16.0	3829982
61	0.0	0.0	0.0	4258304	1,024.0	932.0	92.0	2,485.0	151,220.0	281,100.0	926.0	16.0	16.0	3875688
84	0.0	0.0	0.0	4258304	1,024.0	934.0	90.0	2,910.0	151,399.0	281,340.0	922.0	16.0	16.0	3884052
63	0.0	0.0	0.0	4258304	1,024.0	935.0	89.0	2,657.0	151,552.0	280,942.0	928.0	16.0	16.0	3888162
95	0.0	0.0	0.0	4258304	1,024.0	938.0	86.0	5,287.0	155,452.0	281,257.0	938.0	16.0	16.0	3900700
74	0.0	0.0	0.0	4258304	1,024.0	944.0	80.0	2,845.0	153,016.0	281,185.0	937.0	16.0	16.0	3925616
87	0.0	0.0	0.0	4258304	1,024.0	944.0	80.0	2,755.0	152,522.0	281,270.0	932.0	16.0	16.0	3925648

Conclusion

ECN employs a marking mechanism with congested queues, signaling to downstream nodes that the network link is experiencing congestion. CNP facilitates communication between the receivers and senders, prompting the latter to reduce their sending rates and avoid exacerbating the congestion.

ECN effectively manages the congestion with a small data size but fails with a big data size. It could take a while to receive notification, and the sender does not stop before getting the notification. ECN alone may not be able to control congestions for a big data size.

Test case 3: JCT with PFC and ECN

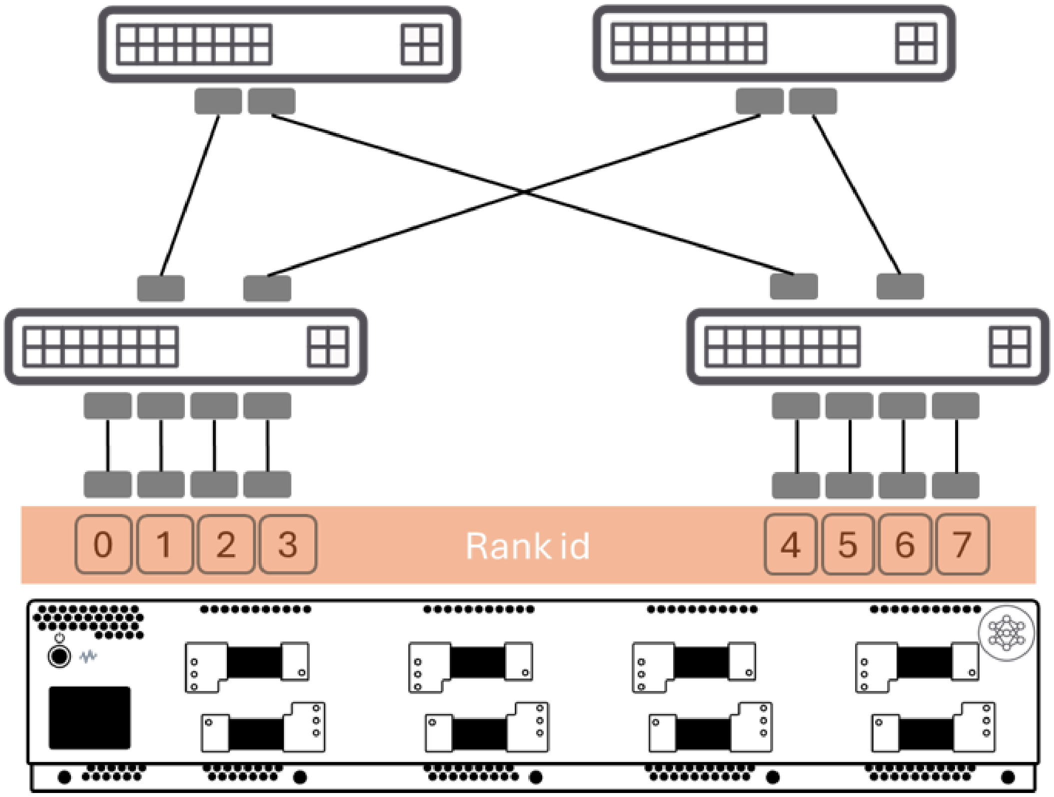
Overview

This test case employs an all-to-all collective operation to evaluate the performance of the fabric with both PFC and ECN enabled.

Objectives

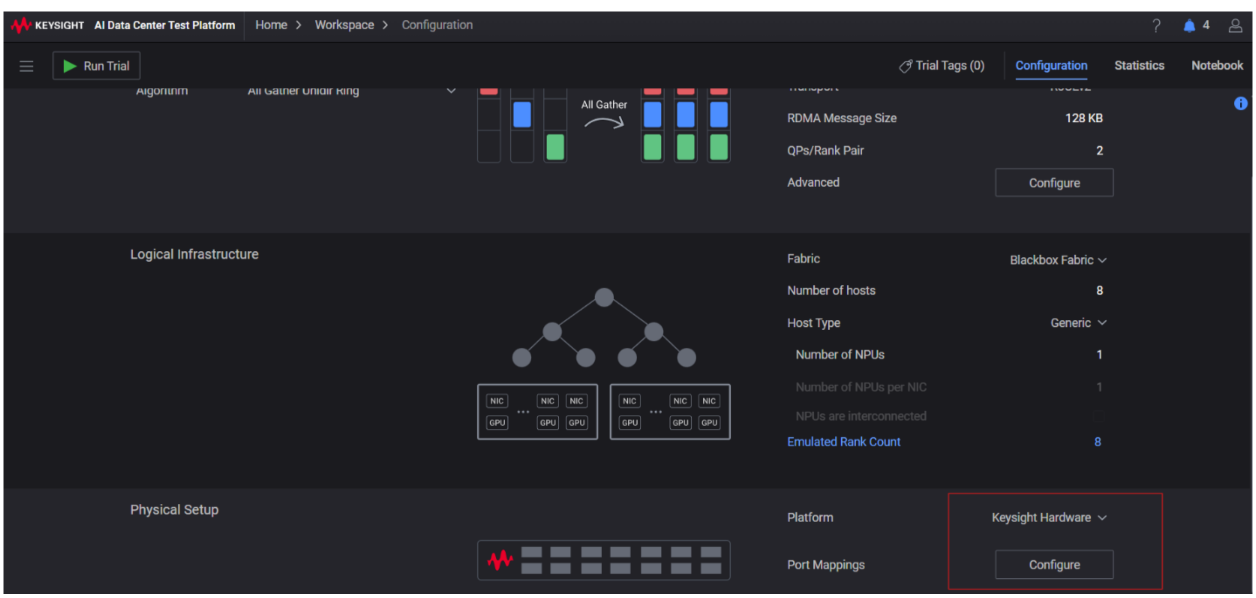
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Verify PFC / ECN dynamically adjusts the available network bandwidth to accommodate the priority traffic, preventing congestion.

Setup

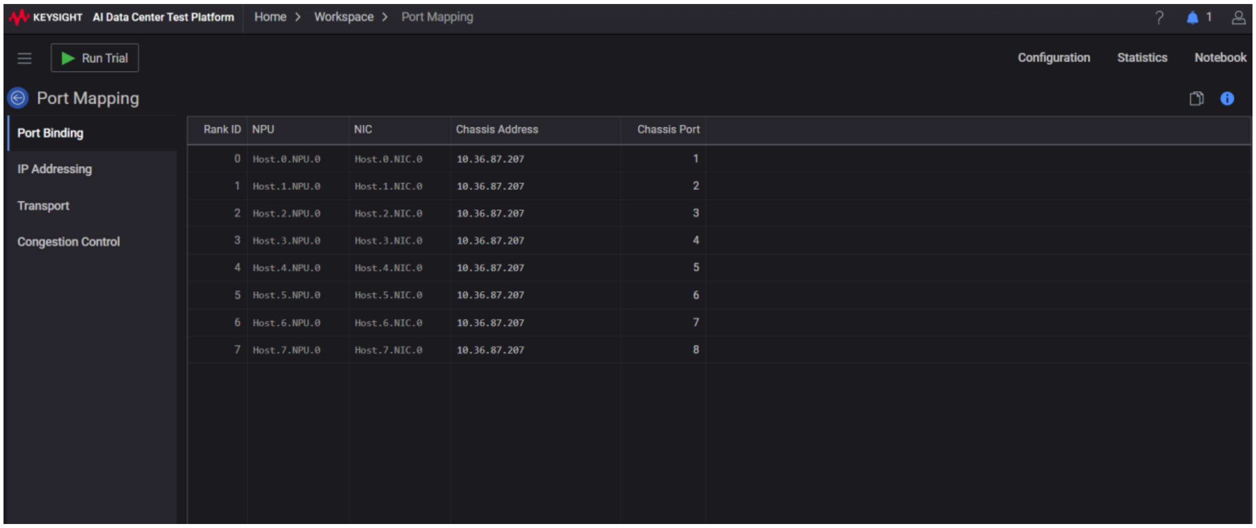


Step-by-step instructions

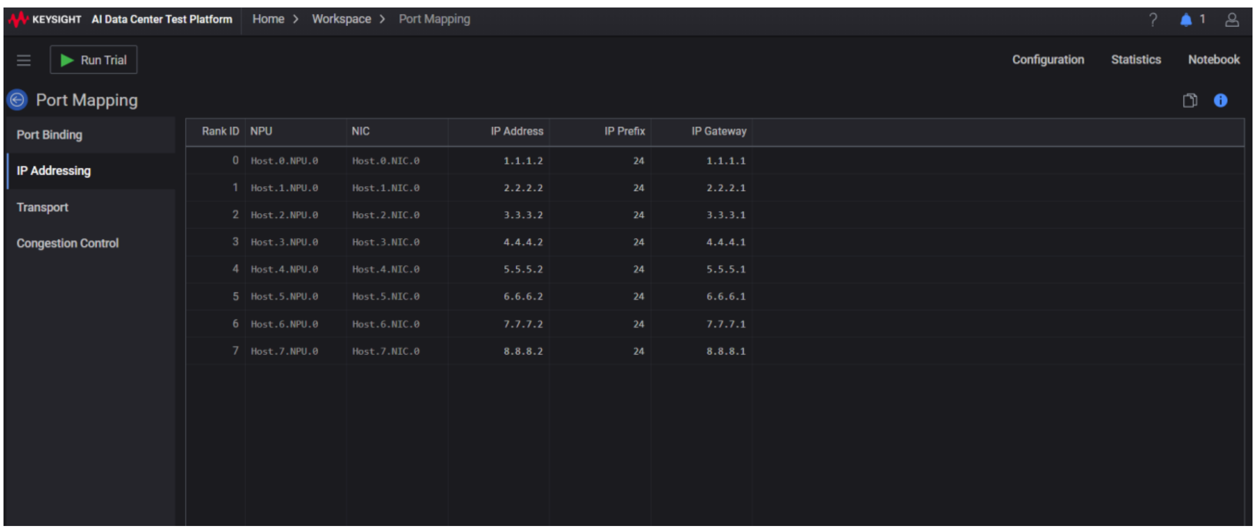
1. Under **Platform**, select **Keysight Hardware.** Select **Configure**.



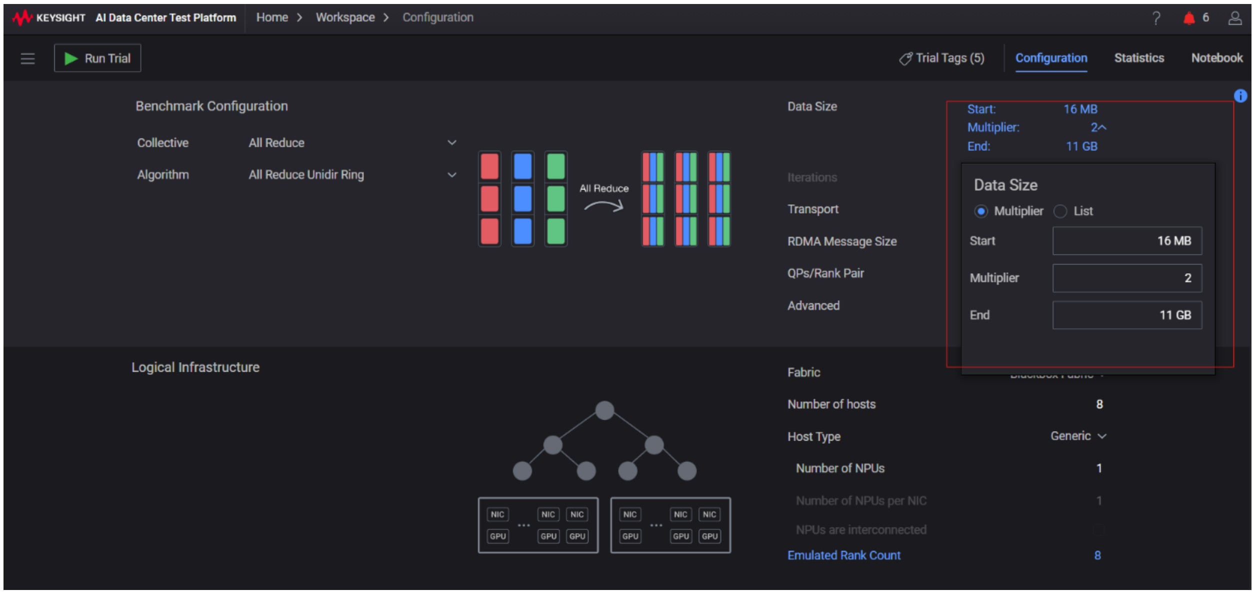
2. Set the port binding.



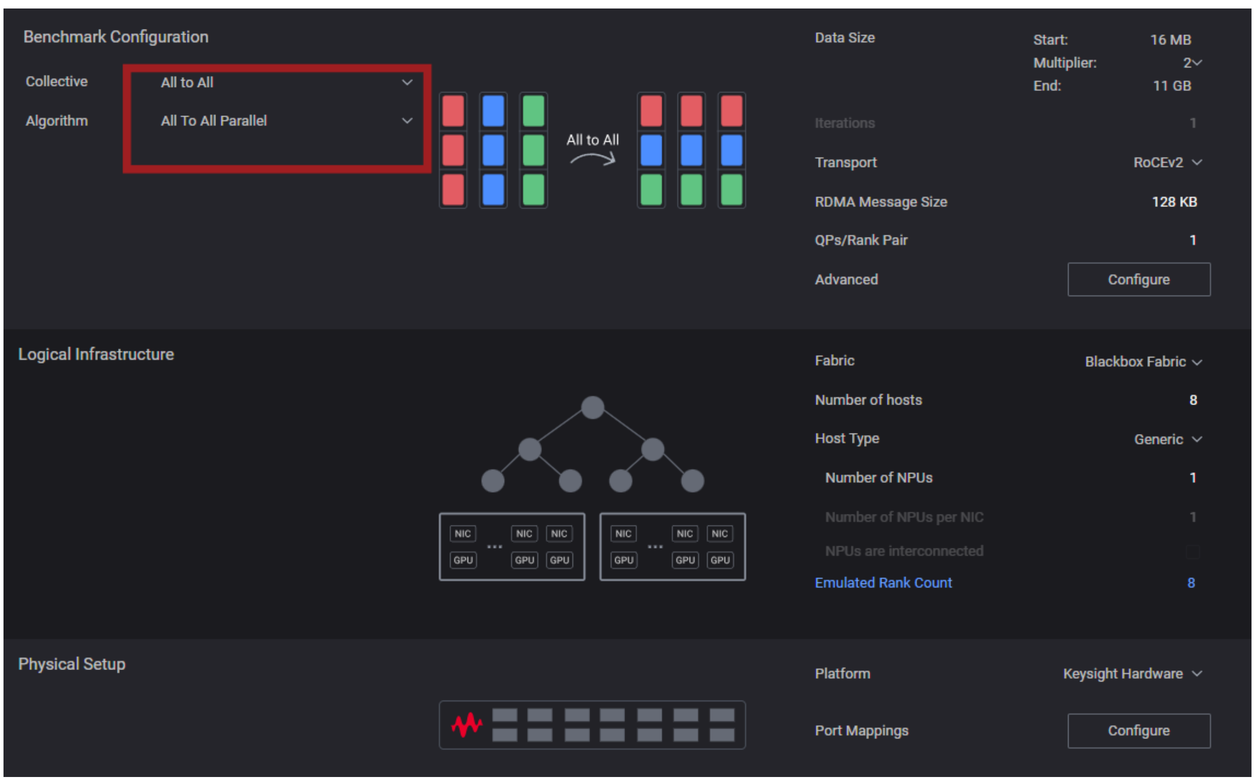
3. Set the IP address and gateway.



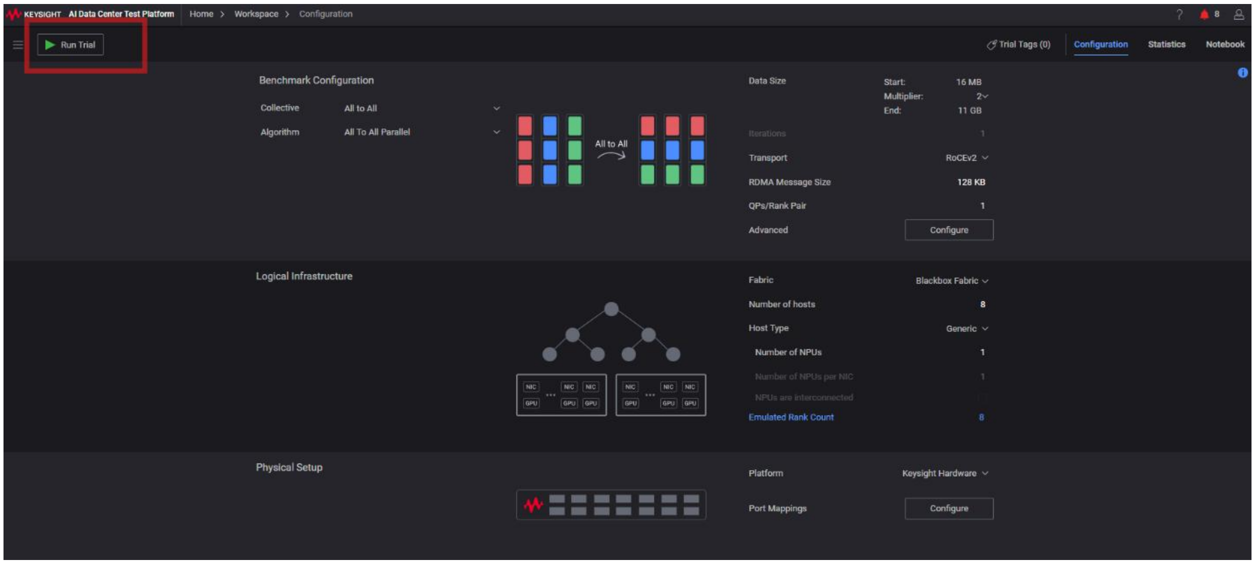
4. Set the data size from 16MB to 11GB with a multiplier of 2.



5. Set the collective and algorithms.



6. Select **Run Trial**.



During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Collectives

  Other ring algorithms to test: All-Gather-Ring and Reduce-Scatter-Ring.

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

- RDMA write message size

  Within a single QP in collectives, multiple RDMA Write operations are performed, with each operation split into separate chunks based on the RDMA write message size for the data chunk. Within these RDMA Write operations, each packet is defined by its corresponding RDMA MTU.

- Parallel-QPs

  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

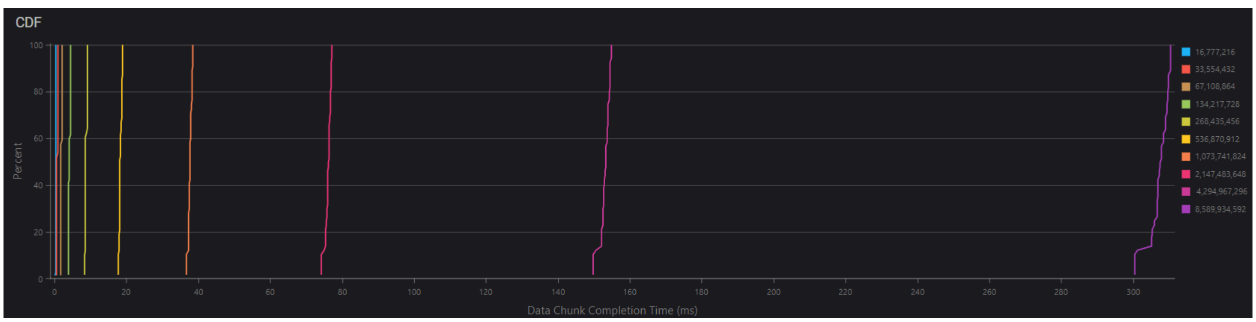
Results analysis

- Completion time

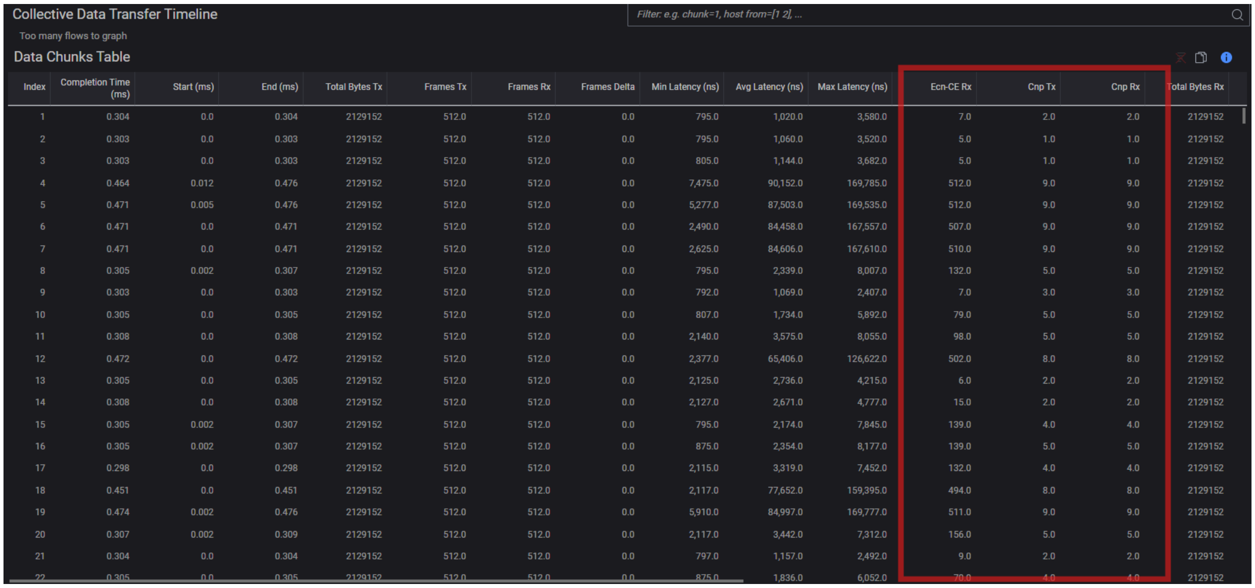
Trial Summary

	Completion Time (ms)	Algbw (GB/s)	Busbw (GB/s)	Ideal (%)	PFC Rx	ECN CE Rx	CNP Tx	Retransmits	Min FCT (ms)	P50 FCT (ms)	P95 FCT (ms)	Max FCT (ms)	Data Size (B)	Collective
0	0.476	35.25	30.84	62.92	0.0	0.0	0.0	0.0	0.29	0.31	0.47	0.47	16777216	ALL_TO_ALL-1
1	0.949	35.36	30.94	63.11	0.0	0.0	0.0	0.0	0.6	0.61	0.95	0.95	33554432	ALL_TO_ALL-2
2	2.124	31.6	27.65	56.4	0.0	0.0	0.0	0.0	1.69	1.73	2.12	2.12	67108864	ALL_TO_ALL-3
3	4.491	29.89	26.15	53.35	0.0	0.0	0.0	0.0	3.94	4.01	4.49	4.49	134217728	ALL_TO_ALL-4
4	9.219	29.12	25.48	51.98	0.0	0.0	0.0	0.0	8.38	8.59	9.21	9.22	268435456	ALL_TO_ALL-5
5	18.926	28.37	24.82	50.64	0.0	0.0	0.0	0.0	17.84	18.24	18.92	18.93	536870912	ALL_TO_ALL-6
6	38.428	27.94	24.45	49.88	0.0	0.0	0.0	0.0	36.64	37.75	38.42	38.42	1073741824	ALL_TO_ALL-7
7	77.195	27.82	24.34	49.66	0.0	0.0	0.0	0.0	74.19	76.22	77.19	77.19	2147483648	ALL_TO_ALL-8
8	154.842	27.74	24.27	49.51	0.0	0.0	0.0	0.0	149.81	153.28	154.84	154.84	4294967296	ALL_TO_ALL-9
9	310.403	27.67	24.21	49.4	0.0	0.0	0.0	0.0	300.48	307.67	310.39	310.4	8589934592	ALL_TO_ALL-10

- Long-tail latency is observed



- ECN is marking, and CNP is transmitted between the receiver and sender

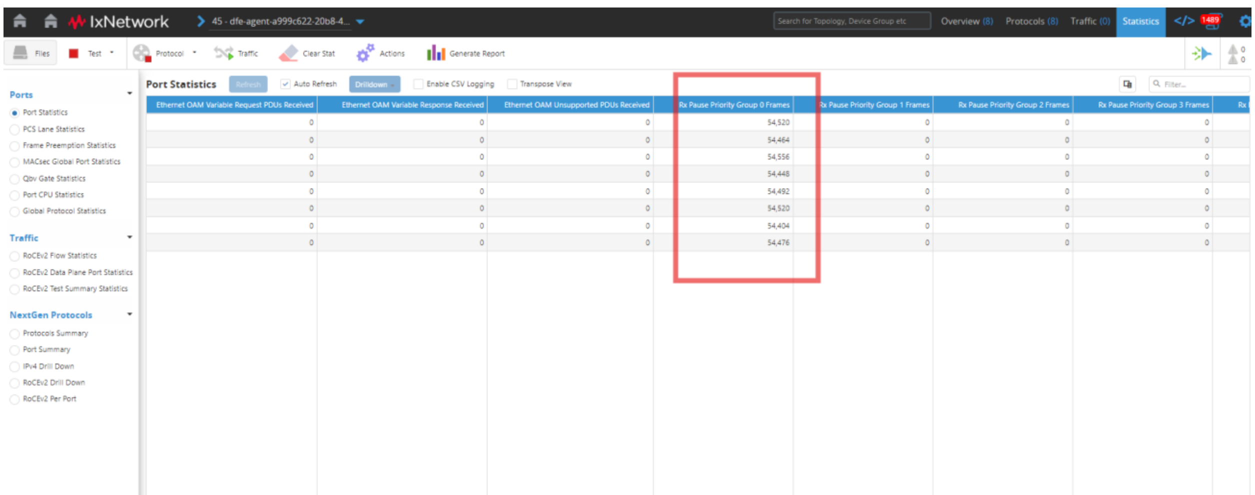

Collective Data Transfer Timeline

Too many flows to graph

Data Chunks Table

Index	Completion Time (ms)	Start (ms)	End (ms)	Total Bytes Tx	Frames Tx	Frames Rx	Frames Delta	Min Latency (ns)	Avg Latency (ns)	Max Latency (ns)	Ecn-CE Rx	Cnp Tx	Cnp Rx	Total Bytes Rx
1	0.304	0.0	0.304	2129152	512.0	512.0	0.0	795.0	1,020.0	3,580.0	7.0	2.0	2.0	2129152
2	0.303	0.0	0.303	2129152	512.0	512.0	0.0	795.0	1,060.0	3,520.0	5.0	1.0	1.0	2129152
3	0.303	0.0	0.303	2129152	512.0	512.0	0.0	805.0	1,144.0	3,682.0	5.0	1.0	1.0	2129152
4	0.464	0.012	0.476	2129152	512.0	512.0	0.0	7,475.0	90,152.0	169,785.0	512.0	9.0	9.0	2129152
5	0.471	0.005	0.476	2129152	512.0	512.0	0.0	5,277.0	87,503.0	169,535.0	512.0	9.0	9.0	2129152
6	0.471	0.0	0.471	2129152	512.0	512.0	0.0	2,490.0	84,458.0	167,557.0	507.0	9.0	9.0	2129152
7	0.471	0.0	0.471	2129152	512.0	512.0	0.0	2,625.0	84,606.0	167,610.0	510.0	9.0	9.0	2129152
8	0.305	0.002	0.307	2129152	512.0	512.0	0.0	795.0	2,339.0	8,007.0	132.0	5.0	5.0	2129152
9	0.303	0.0	0.303	2129152	512.0	512.0	0.0	792.0	1,069.0	2,407.0	7.0	3.0	3.0	2129152
10	0.305	0.0	0.305	2129152	512.0	512.0	0.0	807.0	1,734.0	5,892.0	79.0	5.0	5.0	2129152
11	0.308	0.0	0.308	2129152	512.0	512.0	0.0	2,140.0	3,575.0	8,055.0	98.0	5.0	5.0	2129152
12	0.472	0.0	0.472	2129152	512.0	512.0	0.0	2,377.0	65,406.0	126,622.0	502.0	8.0	8.0	2129152
13	0.305	0.0	0.305	2129152	512.0	512.0	0.0	2,125.0	2,736.0	4,215.0	6.0	2.0	2.0	2129152
14	0.308	0.0	0.308	2129152	512.0	512.0	0.0	2,127.0	2,671.0	4,777.0	15.0	2.0	2.0	2129152
15	0.305	0.002	0.307	2129152	512.0	512.0	0.0	795.0	2,174.0	7,845.0	139.0	4.0	4.0	2129152
16	0.305	0.002	0.307	2129152	512.0	512.0	0.0	875.0	2,354.0	8,177.0	139.0	5.0	5.0	2129152
17	0.298	0.0	0.298	2129152	512.0	512.0	0.0	2,115.0	3,319.0	7,452.0	132.0	4.0	4.0	2129152
18	0.451	0.0	0.451	2129152	512.0	512.0	0.0	2,117.0	77,652.0	159,395.0	494.0	8.0	8.0	2129152
19	0.474	0.002	0.476	2129152	512.0	512.0	0.0	5,910.0	84,997.0	169,777.0	511.0	9.0	9.0	2129152
20	0.307	0.002	0.309	2129152	512.0	512.0	0.0	2,117.0	3,442.0	7,312.0	156.0	5.0	5.0	2129152
21	0.304	0.0	0.304	2129152	512.0	512.0	0.0	797.0	1,157.0	2,492.0	9.0	2.0	2.0	2129152
22	0.305	0.0	0.305	2129152	512.0	512.0	0.0	875.0	1,836.0	6,052.0	70.0	4.0	4.0	2129152

- PFC is triggered during tests with a sense of congestion



Ethernet OAM Variable Request PDUs Received	Ethernet OAM Variable Response Received	Ethernet OAM Unsupported PDUs Received	Rx Pause Priority Group 0 Frames	Rx Pause Priority Group 1 Frames	Rx Pause Priority Group 2 Frames	Rx Pause Priority Group 3 Frames
0	0	0	54,520	0	0	0
0	0	0	54,464	0	0	0
0	0	0	54,556	0	0	0
0	0	0	54,448	0	0	0
0	0	0	54,492	0	0	0
0	0	0	54,520	0	0	0
0	0	0	54,404	0	0	0
0	0	0	54,476	0	0	0

Conclusion

ECN employs a marking mechanism with congested queues, signaling to downstream nodes that the network link is experiencing congestion. CNP facilitates communication between the receivers and senders, prompting the latter to reduce their sending rates and avoid exacerbating the congestion.

PFC and ECN effectively manage congestion. Operations run smoothly across various data sizes. Large data sizes result in lower bus bandwidth and long tail latency.

Test case 4: Gather collective with PFC only

Overview

The Gather collective represents a scenario where multiple NPUs transmit data to a single NPU, resulting in N:1 congestion. This type of collective is particularly challenging for the fabric as it requires efficient handling of aggregated traffic and minimizing contention at the destination node. In this case, the fabric must be able to effectively manage the influx of data from the other nodes, ensuring that packets are delivered reliably and efficiently while avoiding congestion hotspots.

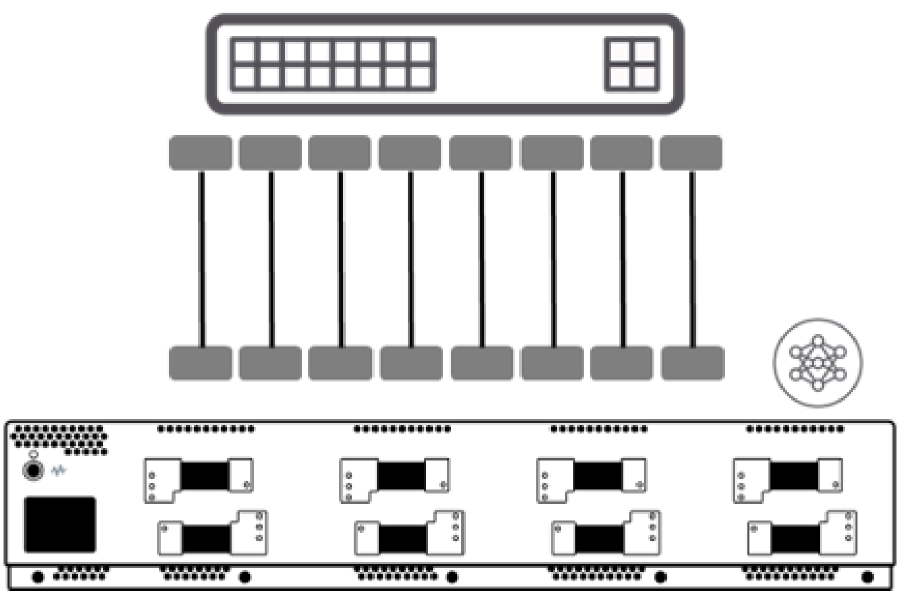
PFC is a flow control algorithm designed to manage network bandwidth and prevent congestion. It prioritizes traffic based on importance, ensuring that critical packets (like RDMA requests) are delivered efficiently while less important packets (like Acknowledgment (ACKs) or Negative Acknowledgment (NAKs)) can be delayed or dropped, if necessary. PFC categorizes incoming traffic into different priority levels based on its importance.

This test case employs the gather collective operation to evaluate the performance of the fabric with PFC enabled.

Objectives

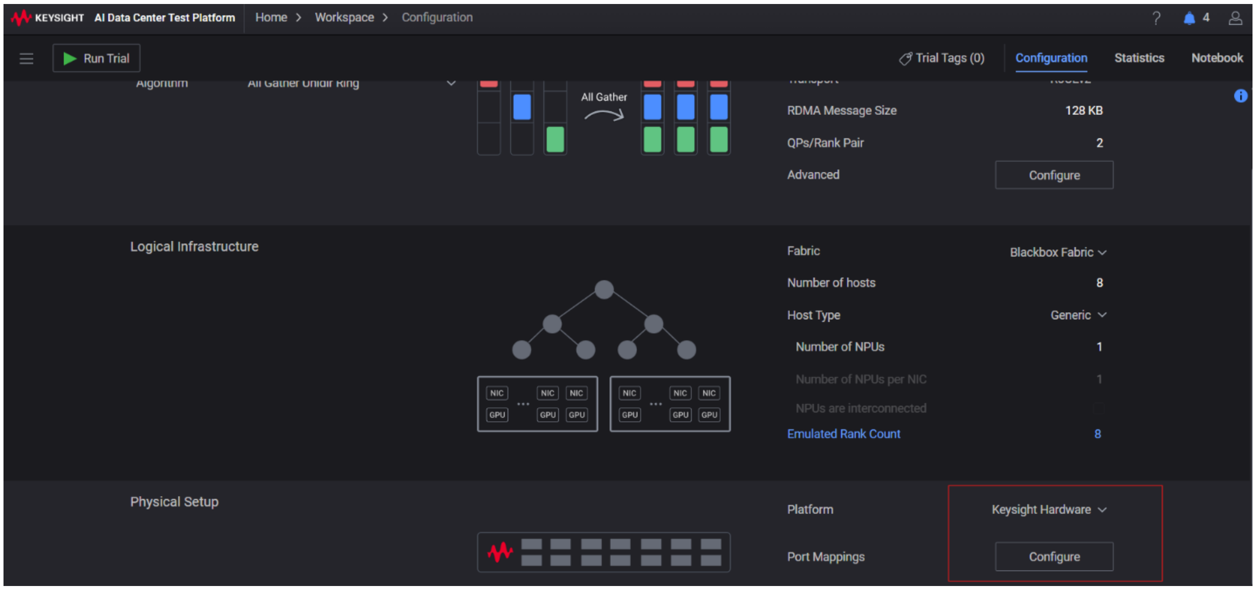
1. Get algorithm bandwidth and long tail latency across fabric in data center.
2. Verify PFC dynamically adjusts the available network bandwidth to accommodate the priority traffic, preventing congestion.

Setup

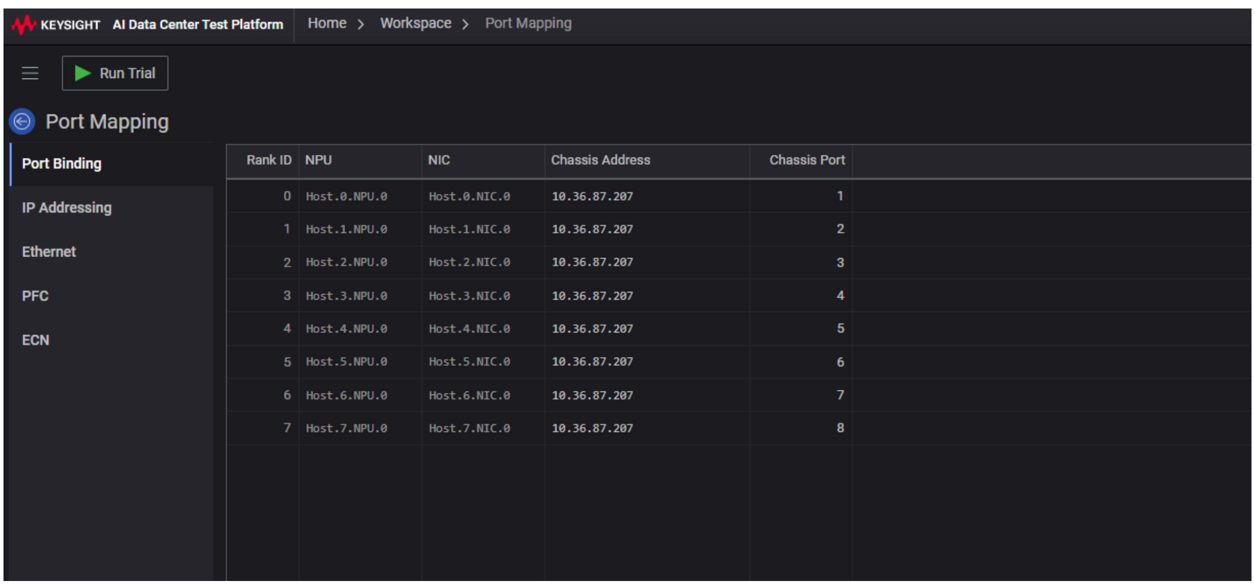


Step-by-step instructions

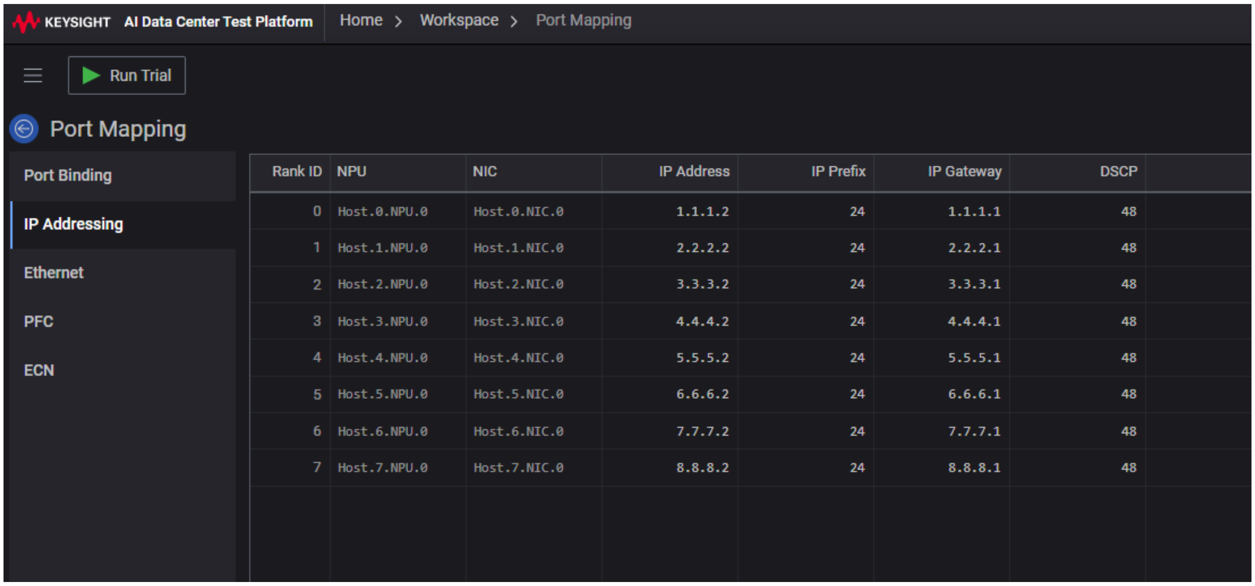
1. Under **Platform**, select **Keysight Hardware**. Select **Configure**.



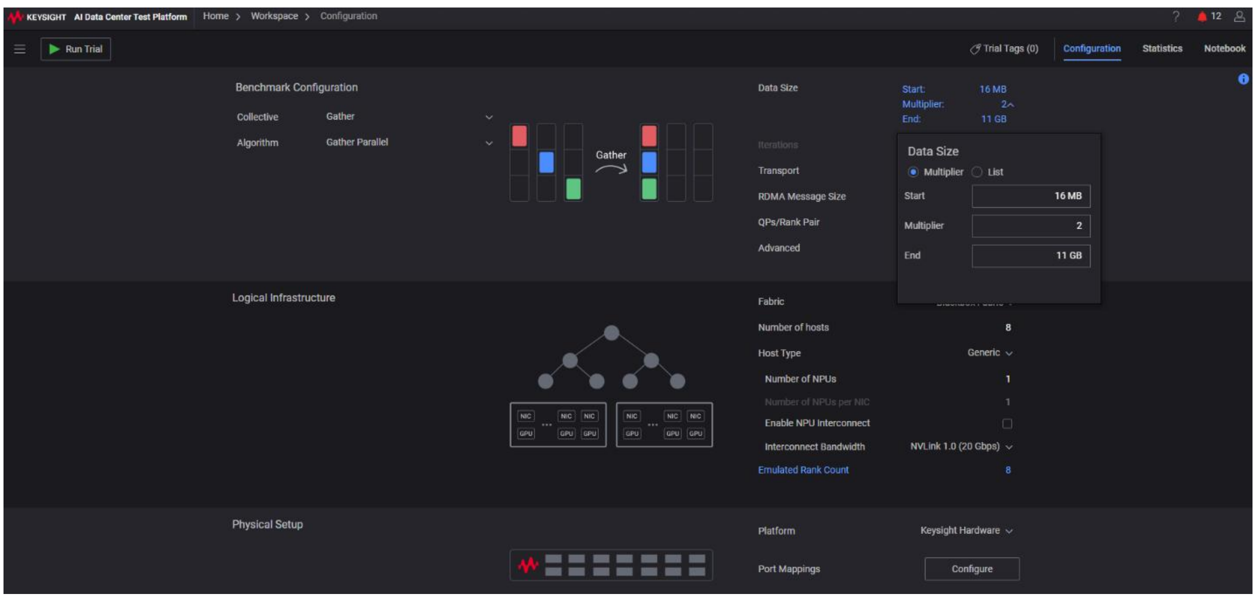
2. Set the port binding.



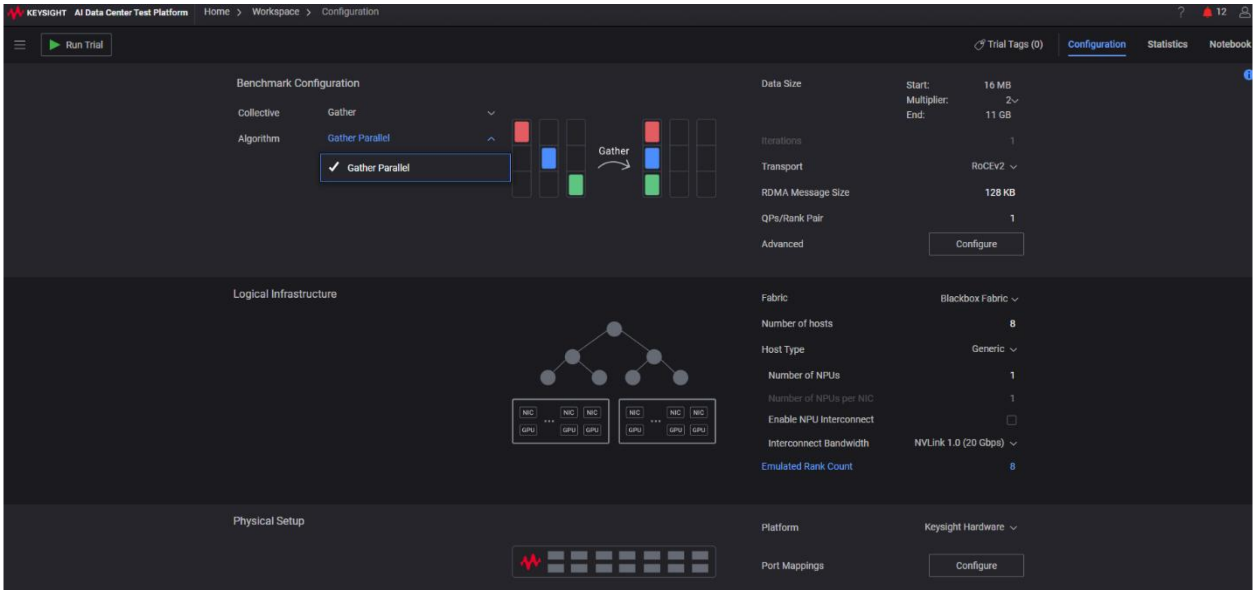
3. Set the IP address and gateway.



4. Set the data size from 16MB to 11GB with a multiplier of 2.



5. Set the collective and algorithms.



6. Select **Run Trial**.

During runtime, the view automatically switches to the statistic view.

Test variables

- RoCE MTU

  Select an ‘active’ MTU, the largest value from the preceding list, smaller than Eth MTU in the system (and considers RoCE transport headers and CRC fields). For example, with default Ethernet MTU = 1500 bytes, RoCE uses 1024; and with Ethernet MTU = 4200 bytes, it uses 4096 as an ‘active MTU.’

  Fix size MTU: 256, 512, 1024, 2048, or 4096 bytes

  Port MTU 8192 — IB MTU 4096

  Port MTU 2200 — IB MTU 2048

  Port MTU 1500 — IB MTU 1024

- Data size

  Collectives move memory data across each rank. Data size may refer to data or gradient tensor size in bytes.

  Data parallel: Gradients tensor size in bytes.

  Tensor parallel: Training data tensor size in bytes.

- RDMA Write message size

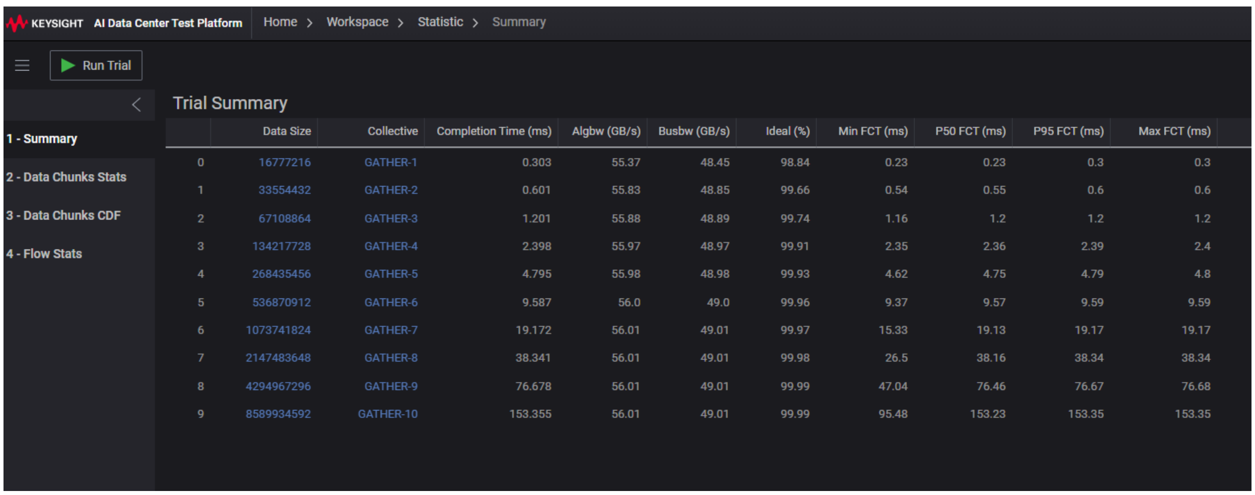
  Within a single QP in collectives, multiple RDMA Write operations are performed, with each operation split into separate chunks based on the RDMA write message size for the data chunk. Within these RDMA Write operations, each packet is defined by its corresponding RDMA MTU.

- Parallel-QPs

  When sending or receiving data between two ranks, instead of using single q-pairs, multiple parallel q-pairs are created to proceed with the data transfer. This can be useful on multi-level fabrics, which require multiple queue pairs to have good routing entropy.

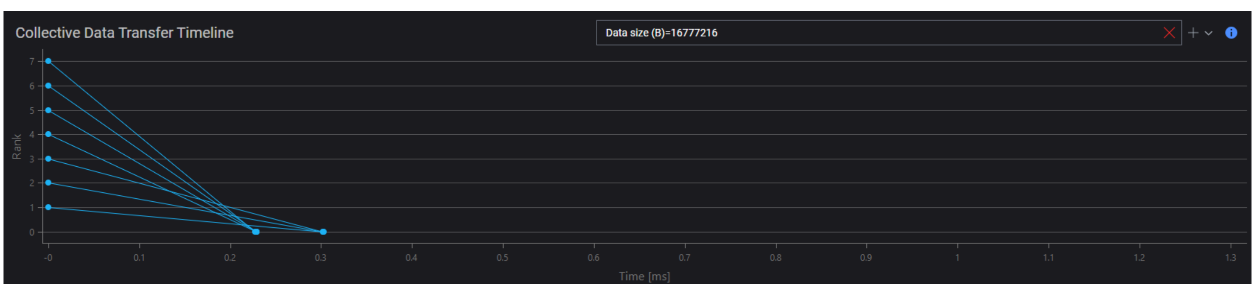
Results analysis

- Completion time



	Data Size	Collective	Completion Time (ms)	Algbw (GB/s)	Busbw (GB/s)	Ideal (%)	Min FCT (ms)	P50 FCT (ms)	P95 FCT (ms)	Max FCT (ms)
0	16777216	GATHER-1	0.303	55.37	48.45	98.84	0.23	0.23	0.3	0.3
1	33554432	GATHER-2	0.601	55.83	48.85	99.66	0.54	0.55	0.6	0.6
2	67108864	GATHER-3	1.201	55.88	48.89	99.74	1.16	1.2	1.2	1.2
3	134217728	GATHER-4	2.398	55.97	48.97	99.91	2.35	2.36	2.39	2.4
4	268435456	GATHER-5	4.795	55.98	48.98	99.93	4.62	4.75	4.79	4.8
5	536870912	GATHER-6	9.587	56.0	49.0	99.96	9.37	9.57	9.59	9.59
6	1073741824	GATHER-7	19.172	56.01	49.01	99.97	15.33	19.13	19.17	19.17
7	2147483648	GATHER-8	38.341	56.01	49.01	99.98	26.5	38.16	38.34	38.34
8	4294967296	GATHER-9	76.678	56.01	49.01	99.99	47.04	76.46	76.67	76.68
9	8589934592	GATHER-10	153.355	56.01	49.01	99.99	95.48	153.23	153.35	153.35

- Long-tail latency is observed



- PFC is triggered during tests with a sense of congestion

Conclusion

PFC effectively manages the congestion. Operations run smoothly across various data sizes. Gather collective will achieve nearly 99% of the ideal ratio because it’s oversubscribed.

References

- MTU Considerations for RoCE based Applications (nvidia.com)
- Environment Variables — NCCL 2.21.5 documentation (nvidia.com)
- (24) To spray or not to spray: Solving the low entropy problem of the AI/ML training workloads in the Ethernet Fabrics | LinkedIn
- Doubling all2all Performance with NVIDIA Collective Communication Library 2.12 | NVIDIA Technical Blog
- sonic-mgmt/docs/testplan/pfc/PFC_PAUSE_RESPONSE_HEADROOM_README.md at master sonic-net/sonic-mgmt (github.com)