Reference Design
September 2025

Intel® Gaudi® 3 AI Accelerator Cluster Reference Design

Accelerate your AI solutions with the latest Intel Gaudi 3 accelerator-based systems—built for scale and expandability with all-Ethernet-based fabrics and support for a wide range of industry AI models and frameworks

Contents

1. Introduction 1
2. Reference Design Overview 2
   - Compute Nodes 2
   - Cluster Networking Overview 2
3. Expandable 32-Node Cluster 3
   - Modular Cluster Design 3
   - Unified Cluster Design 5
   - Deployment Bill of Materials 7
4. Design Considerations 8
   - Network Considerations 8
   - Storage Considerations 9
   - Control Plane Considerations 9
   - PDU Considerations 10
5. Software 10
6. Summary 11

Appendix A: L3 Scale-Out Accelerator Fabric Configuration 12
   - Example Scale-Out Fabric Switch 12
   - Scale-Out IP Address Configuration 13
   - Manual Generation of the gaudinet.json Network Configuration File 13
   - Example gaudinet.json Generator Script 14

Appendix B: L2 Scale-Out Accelerator Fabric Configuration 15
   - Reference IP Address Assignment 15

1. Introduction

The rapid deployment of artificial intelligence (AI) is reshaping industries worldwide, driven by enhanced usability and a diverse array of vertical solutions tailored to sectors such as healthcare, legal, transportation, manufacturing, and energy. Despite AI's transformative potential, adoption can be hindered by the high costs associated with AI infrastructure and concerns over vendor lock-in. However, the emergence of open industry solutions, like those based on the Intel® Gaudi® AI accelerator product line, is paving the way for broader AI integration.

Intel Gaudi accelerators are specifically designed for deep learning (DL) and Generative AI (GenAI), excelling in large language model (LLM) and multi-modal model training and inferencing. These accelerators are purpose-built for running DL workloads of various sizes across multi-tenant data centers, offering a competitive edge in GenAI compute capability, pricing, energy efficiency, and market availability.[1]

Enterprise AI solutions often require multiple accelerators or GPUs interconnected across several chassis, utilizing extensive racks of compute, network, and storage equipment. While proprietary fabrics like Nvidia's NVLink or InfiniBand have dominated AI GPU cluster deployments, Ethernet-based solutions are gaining traction, offering a more flexible approach.

The integration of AI is prompting a reevaluation of data center designs. Traditional data centers, initially optimized for general-purpose computing, are being retrofitted with high-density GPU clusters to support complex AI models. This phased approach involves incorporating GPU accelerators into existing server racks, enhancing power and cooling systems, and expanding networking capabilities to manage increased computational loads.

Conversely, new data centers are being designed with a central focus on AI workloads. These next-generation facilities feature high power density, advanced thermal management systems, and architecture that is optimized for the scalability of GPUs. Purpose-built data centers offer enhanced operational efficiency and long-term scalability, addressing the growing demand for AI compute while circumventing the limitations of legacy infrastructure.

This document aims to guide enterprise IT operations, developers, and infrastructure leaders in designing and deploying multi-node AI infrastructure using Intel Gaudi 3 AI accelerator-based systems,[2] helping to ensure they are equipped to meet the evolving demands of AI integration.

2. Reference Design Overview

Figure 1 provides an overview of the essential components in the Intel Gaudi 3 AI accelerator cluster Reference Design. The Intel Gaudi 3 accelerators are compatible with a diverse array of industry AI models and frameworks, though detailing these options is beyond the scope of this document. This Reference Design outlines an "AI-ready" cluster infrastructure, encompassing necessary node-level software, system management software, and provisioning and management tools. Additionally, it suggests specific hardware for compute, network, storage, and control plane. It's important to note that the software stack is not tailored to any particular use case, such as training or inference. For a detailed summary of the software components, please refer to Section 5.

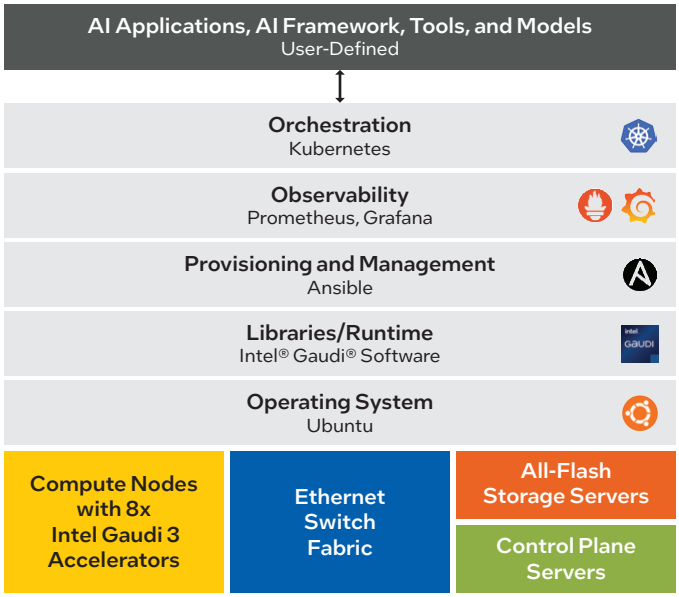

Figure 1. Overview of Intel® Gaudi® 3 AI accelerator Reference Design requirements.

Compute Nodes

Several leading OEMs offer AI computing systems featuring Intel Gaudi 3 accelerators. For this Reference Design, the compute element (referred to as a node) is an air-cooled server chassis featuring eight Intel Gaudi 3 accelerators. This compute node is purpose-built to enable rapid development, training, and deployment of large GenAI models. It can be configured into scalable clusters of many nodes to handle vast data sets for AI and is well-suited for LLMs, multi-modal models, recommendation engines, and neural network applications.

Intel Gaudi 3 accelerators have proven to be a viable alternative to the competition in GenAI compute capability, pricing, energy efficiency, and market availability.[3] One key advantage is in networking: Intel Gaudi accelerators are designed for excellent scalability. Each accelerator integrates 24x 200 GbE RDMA over Converged Ethernet (RoCE) ports, of which 21 are used for scale-up connectivity within an eight-card node, and three are used for scale-out connectivity to expand beyond a single node. These external ports are grouped into 6x 800 GbE ports on the node, which are then connected to 800 GbE Ethernet switches that make up the accelerator fabric. These switches directly interconnect all the AI accelerators in the cluster and deliver up to 25.6 Tbps throughput.[4]

Cluster Networking Overview

GenAI clusters typically have three primary high-speed networks that connect the compute, storage, and control plane servers and a fourth network for out-of-band management.

Compute

A group of Intel® Xeon® processor-based accelerator nodes connects to a dedicated accelerator fabric. The accelerator fabric is an Ethernet network used by the Intel Gaudi AI accelerators to directly communicate with other Intel Gaudi AI accelerators in other nodes. This is implemented as a three-ply full Clos Ethernet fabric using OSFP4x 200 Gbps links to provide 800 Gbps per connector.

Storage

A group of Intel Xeon processor-based server nodes with SSDs and a scalable storage application connects through a dedicated full Clos 100 Gbps Ethernet fabric, providing high-performance storage for the compute nodes.

Control Plane

The control plane consists of Intel Xeon processor-based server nodes that facilitate and streamline the deployment, management, and operation of a scalable AI cluster. Control plane functions include provisioning, configuration, network setup, upgrades, monitoring, system health, performance metrics, and user management. The control plane fabric is an Ethernet fabric that connects the control plane servers to the compute nodes and the rest of the cluster infrastructure. It is implemented as a full Clos 100 Gbps Ethernet fabric.

Management

The out-of-band management fabric connects the control plane servers, switch management ports, baseboard management controllers (BMCs), and power distribution units (PDUs). The management fabric also handles all switch management communication. It is implemented as a layer 2 fabric with 25 and 1 Gbps connections with 25 Gbps aggregation links.

Figure 2 depicts a cluster with independent storage and control plane fabrics.

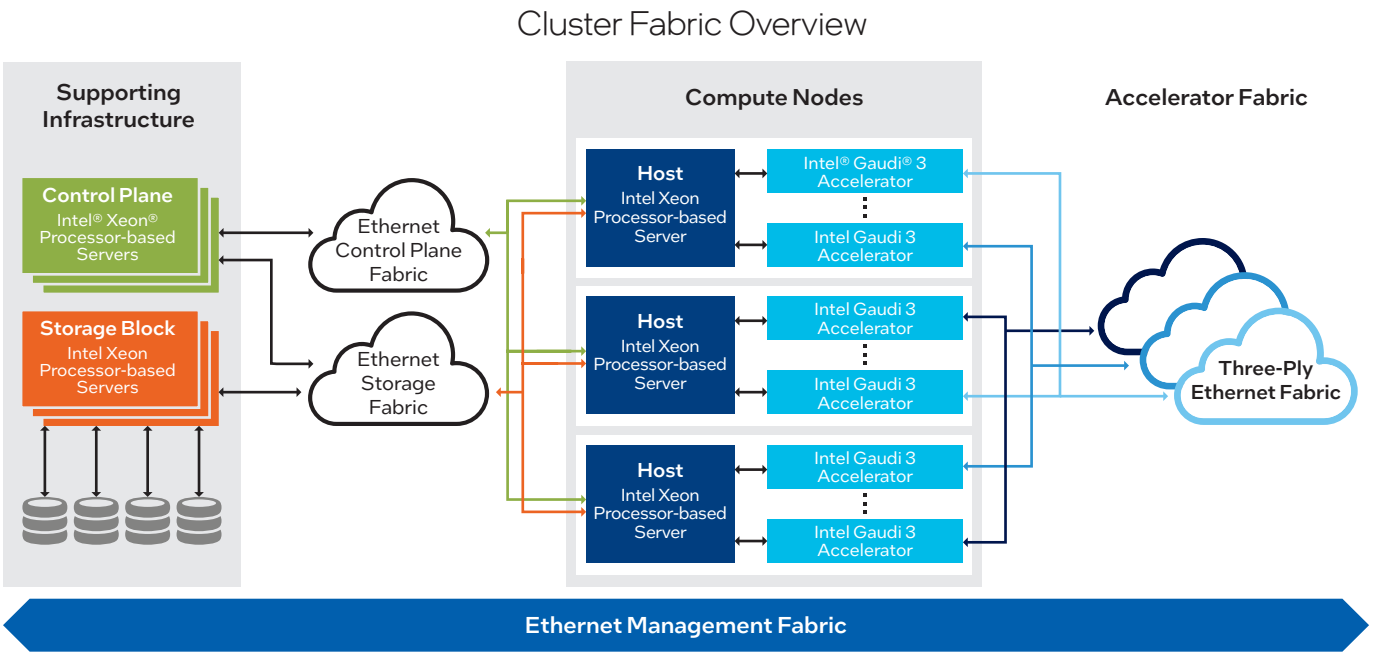

Figure 2. Cluster design using independent storage and control plane fabrics.

3. Expandable 32-Node Cluster

The architecture supports two deployment models:

- **Modular cluster.** This model is ideal in well-established and fully operational data centers that already have essential cluster and storage service functionalities.
- **Unified cluster.** This model integrates compute and accelerator racks with storage, cluster, and network racks into a single cohesive setup.

Modular Cluster Design

In this scenario, the architecture focuses on optimizing space utilization, simplifying cabling, and enhancing efficiency through a modular design. Figure 3 on the following page illustrates a typical modular Intel Gaudi 3 AI architecture configuration.

Figure 3. Typical modular Intel® Gaudi® 3 AI accelerator cluster with external control plane and storage services.

The architecture consists of eight compute racks, each housing four Intel Gaudi AI accelerator nodes (totaling 32 nodes across the eight racks), plus a network rack (for a total of nine racks). Compute racks are equipped with management switches to facilitate device management and monitoring. Depending on the number of ports required, one or more management switches may be necessary. The central rack is dedicated to accelerator networking equipment, including leaf and spine switches. This rack serves as the hub for isolated network connectivity between the compute nodes.

This configuration is a full Clos, non-blocking, three-ply network due to its three-stage structure, full mesh connectivity, and non-blocking design (see Table 1). It features leaf switches that are connected to servers, spine switches that are interconnected to all leaf switches, and redundancy with multiple paths to ensure high performance and reliability.

Table 1. Modular Cluster Characteristics

Clos Feature	Description
Three-stage (three-ply)	Leaf-spine-leaf
Full mesh leaf-spine	Every leaf to every spine
Non-blocking	Uplink=downlink bandwidth
Multipath (Equal Cost Multipath [ECMP])	Three-ply, multiple paths
Provisioning and Management	Yes

The modular cluster design does not include storage and control plane services. Connections from the network rack to storage and control plane services are made with uplinks to the data center's existing infrastructure using storage and cluster leaf switches.

The modular design approach offers flexibility and scalability, allowing for seamless integration of new systems into current operations, with the following benefits:

- **Space optimization.** The architecture is designed to maximize rack space by efficiently arranging nodes and switches, reducing the data center's overall footprint.
- **Cost efficiency.** The architecture uses active optical cables (AOCs) and direct attach cables (DACs) to minimize cabling costs and deliver high-performance connectivity.
- **Scalability.** The modular design allows for easy expansion by adding racks or nodes as needed, without significant reconfiguration.
- **Reduced latency.** The centralized network rack helps ensure that data transfer between nodes and services is quick and efficient, reducing latency and improving performance.
- **Flexibility.** The architecture can be adapted to existing data center setups, making it suitable for both new installations and upgrades to current infrastructure.

Unified Cluster Design

The unified deployment model integrates compute and accelerator racks with storage, cluster, and network racks into a single cohesive setup. This approach is designed to provide a comprehensive solution that includes all necessary components for data processing, storage, and networking within a single deployment. Figure 4 on the following page depicts a typical unified Intel Gaudi 3 AI accelerator architecture configuration.

	Storage Rack	Cluster Rack	Network Rack	Compute Racks				Accelerator Network Rack	Compute Racks			
U	Rack 12	Rack 11	Rack 10	Rack 9	Rack 8	Rack 7	Rack 6	Rack 5	Rack 4	Rack 3	Rack 2	Rack 1
42	Management PDU	Management BMC			Management PDU	Management BMC		Management BMC		Management BMC	Management PDU	
41												
40	Storage Leaf 3		Management Spine 2			Storage Leaf 2				Storage Leaf 2		
39	Cluster Leaf 3		Management Spine 1			Cluster Leaf 2				Cluster Leaf 2		
38			Cluster Gateway 2									
37			Cluster Gateway 1									
36			Cluster Spine 2									
35			Cluster Spine 1									
34			Storage Spine 4									
33	WEKA Server		Storage Spine 3									
32			Storage Spine 2					Accelerator Leaf				
31	WEKA Server		Storage Spine 1	Intel® Gaudi® 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Accelerator Leaf	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node
30								Accelerator Spine				
29	WEKA Server							Accelerator Leaf				
28								Accelerator Leaf				
27	WEKA Server							Accelerator Spine				
26								Accelerator Leaf				
25	WEKA Server	Infrastructure Server						Accelerator Leaf				
24								Accelerator Spine				
23	WEKA Server	Infrastructure Server		Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Accelerator Leaf	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node
22								Accelerator Leaf				
21	WEKA Server	Infrastructure Server						Accelerator Spine				
20								Accelerator Leaf				
19	WEKA Server	Infrastructure Server						Accelerator Leaf				
18								Accelerator Spine				
17	WEKA Server	Infrastructure Server						Accelerator Leaf				
16								Accelerator Leaf				
15	WEKA Server	Infrastructure Server		Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Accelerator Spine	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node
14												
13	WEKA Server	Infrastructure Server										
12												
11	WEKA Server	Infrastructure Server										
10												
9	WEKA Server	Infrastructure Server										
8												
7	WEKA Server	Infrastructure Server		Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node		Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node	Intel Gaudi 3 Accelerator Node
6												
5	WEKA Server	Infrastructure Server										
4												
3	WEKA Server	Infrastructure Server										
2												
1												
	Left and Right PDUs	Left and Right PDUs	PDU	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs	Left and Right PDUs

Figure 4. Typical unified Intel® Gaudi® 3 AI accelerator cluster with control plane and storage services.

The unified architecture's compute and accelerator network racks mirror the components and configuration of the modular architecture. It comprises eight compute racks, each containing four Intel Gaudi AI accelerator nodes, totaling 32 nodes. These racks are equipped with management switches for device oversight. The central accelerator network rack acts as the connectivity hub, featuring a full Clos, non-blocking, three-ply network structure. Leaf switches connect to servers, while spine switches link all leaf switches, delivering high performance and reliability through redundancy and multiple paths.

Dedicated racks for storage solutions, such as Weka or other storage systems, make data readily accessible and easy to manage. Control plane racks contain infrastructure servers for control plane services, enabling distributed computing and resource management.

Network racks comprise storage spine, control plane spine, management spine, and control plane gateway switches. Storage spine switches are typically used to connect storage devices, providing high-speed data transfer and redundancy. Control plane spine switches, on the other hand, are designed to interconnect multiple servers within a cluster, enabling scalability and efficient communication between nodes. Management spine switches are dedicated to handling the management traffic, helping to ensure that administrative tasks and monitoring are conducted smoothly without interfering with the primary data flow. Control plane gateway switches serve as the entry and exit points for data traffic, managing the flow between the internal network and external connections.

Network racks are indispensable for achieving high levels of performance, reliability, and scalability. The structured arrangement of switches within the racks allows for streamlined cabling and reduced latency, which are critical for maintaining optimal network speeds and minimizing downtime. Furthermore, network racks facilitate the integration of advanced technologies such as virtualization and cloud computing, which require robust and flexible network infrastructures.

The unified deployment model is ideal for facilities starting AI infrastructure from scratch, thanks to the following advantages:

- **Simplified deployment.** The unified architecture is ideal for new data centers because it provides a complete solution in one package, reducing the complexity of integrating separate systems.
- **Optimized space utilization.** By consolidating all components into a single setup, the architecture maximizes space efficiency, which is crucial for new data centers with limited space.
- **Cost efficiency.** The unified approach can be more cost-effective as it minimizes the need for additional infrastructure and cabling, reducing overall deployment costs.
- **Ease of management.** With all components integrated, management and maintenance become more straightforward, because there is a single point of control for the entire system.

Deployment Bill of Materials

Table 2 shows a representative bill of materials (BOM) for both the modular and unified architectures.

Table 2. Bill of Materials for the Modular and Unified Deployment Architectures

Functional Area	Architecture	Quantity	Purpose	Part Number
Compute Block				
Intel® Gaudi® 3 AI Accelerator Node	Modular, Unified	32	Compute node	Dell PowerEdge XE9680 server with 8 Intel Gaudi 3 accelerators (Open Compute Project Accelerator Module [OAM]) 128 GB, 900 W
Accelerator Fabric		32		Dell PowerSwitch Z9864F-ON, Sonic OS (PSU intake to I/O port exhaust air flow)
		64	Copper cable	Dell Networking Cable DAC-800G-O112-2M, CK
		640	Transceiver	Dell Networking Transceiver 800G-OSFP112-VR8
		128	Optical cable	EDGE 16F MTP/MTP JUMP OM4, 3-5 M
		192	Optical cable	EDGE 16F MTP/MTP JUMP OM4, 5-10 M
		320	AOC cable	Standard supported AOCs
Management Fabric		9	Management fabric switch	Arista 7010 48x 1G ports, 4x 10G SFP+ uplinks, layer 2,3 capabilities
		14	Cable	10 GbE AOC
		118	Cable	Cat 6 RJ45 straight BMC, PDU
Control Plane				
Management Racks	Unified	4 minimum*	Infrastructure server	2x 5th Gen Intel® Xeon® Scalable processors
Control Plane Fabric		7	Control fabric switch	Arista 7050SX3-48YC8-F 48x 25 GbE SFP28, 8x 100 GbE QSFP28 ports uplinks
		72	25 GbE AOC	
		14	100 GbE AOC	
Storage Block				
Storage Server	Unified	16 minimum*		2x Intel Xeon Platinum 8480+ processor
Storage Server Disks		128		202 Micron 1.9 TB NVMe disks for WEKA storage
Storage Fabric		7	Storage fabric switch	Arista 7260CX3-64 64x 100 GbE QSFP28 ports, layers 2,3 capabilities
		148	100 GbE AOC	

*Scale as needed.

4. Design Considerations

Network Considerations

In AI clusters, the network interconnection of GPU accelerators is vital for efficient connectivity, cost-effectiveness, and performance optimization. This configuration is recognized as a full Clos, non-blocking, three-ply network due to its adherence to Clos architecture principles. It features a three-stage structure comprising ingress, middle, and egress stages, ensuring efficient connectivity between servers and switches. Figure 5 shows the layout for three-ply scale-out networking.

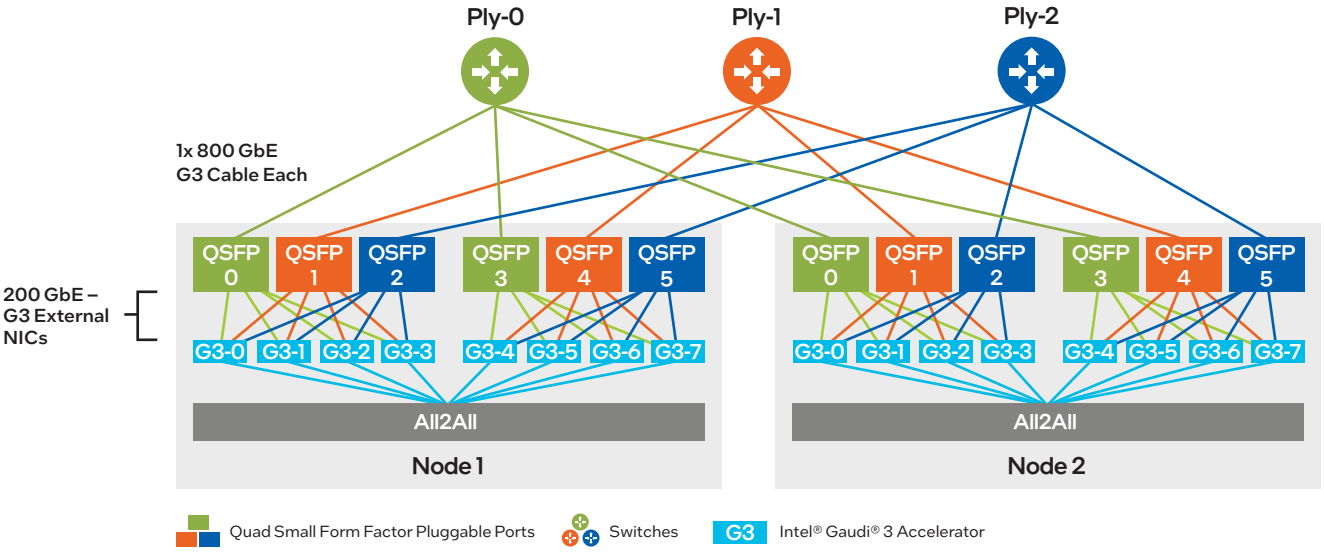

Figure 5. Three-ply scale-out networking layout.

To scale out cluster sizes beyond the capacity of a single-level switch, a non-blocking or Clos fabric is necessary. A basic Clos fabric consists of two switch levels: leafs and spines. Leaf switch ports are divided equally, with 50% connected to Intel Gaudi AI accelerators and 50% to spine switches. Spines exclusively connect to leaf ports, resulting in twice as many leafs as spines in a complete Clos setup when using switches of the same size. Figure 6 shows the network leaf and spine switches.

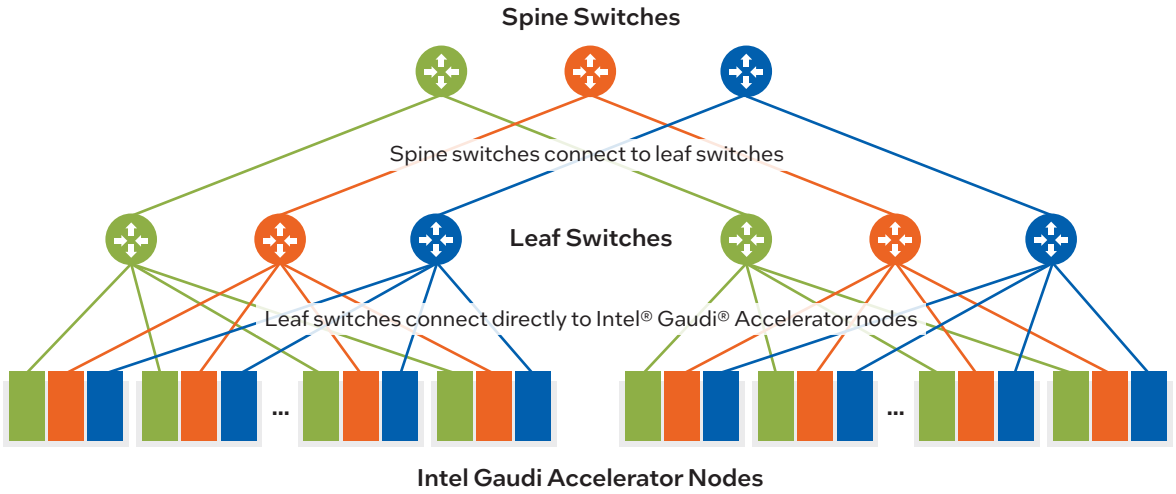

Figure 6. Network leaf and spine switches.

The accelerator fabric mirrors the control plane and storage fabrics, forming a full Clos fabric with 800 Gbps connections configured as 4x 200 Gbps connections. Each compute node has six 800 Gbps connections, with each pair containing a single 200 Gbps connection from each of the eight Intel Gaudi 3 accelerators in the system. The Intel Gaudi architecture supports three independent plys for cluster scale-out, making this topology highly efficient. Three leaf switches are surrounded by eight compute nodes.

The accelerator network covered in this document focuses on L3 network configuration. However, for smaller setups, an L2 network may be more suitable, as it offers advantages in cost, performance, and simplified network configuration. L3 and L2 network fabric connection details are covered in Appendix A and Appendix B, respectively.

The management fabric, also known as the out-of-band fabric, offers isolated access to core data center infrastructure that focuses on three main areas: BMCs, network switches, and PDUs. These devices connect to a 1 Gbps management switch associated with each group of eight systems. While there are various ways to configure this network, three VLANs are implemented across four management switches to separate BMC, PDU, and switch access.

The control plane fabric is constructed as a full Clos fabric leaf/spine/switch network that connects to each compute node linked via a 25 Gbps connection.

Similarly, the storage fabric is a full Clos fabric, featuring two 100 Gbps connections to the leaf from the compute nodes. The leaf switch is an Arista.

Each of the cabling methods used in this Reference Design offers clear advantages:

- **AOCs:** AOCs are favored in the modular architecture for their ability to deliver high-speed data transmission over long distances without signal degradation. Integrated transceivers eliminate the need for separate purchases, reducing costs depending on compatible vendor-validated cables. AOCs are ideal for connecting leaf switches to Intel Gaudi AI accelerator nodes, supporting high bandwidth requirements and minimizing latency. They allow for flexible rack arrangements, spanning up to 100 meters, accommodating various data center layouts. Known for reliability and ease of installation, AOCs maintain consistent performance across compute and network racks.
- **DACs:** DACs are used for short-distance connections within the network rack, such as between leaf and spine switches. They are cost-effective and provide a simple plug-and-play solution for high-speed data transfer over short ranges. DACs provide highly secure and efficient connections between critical network components, supporting the architecture's non-blocking fabric design. They are advantageous in space-limited scenarios, require minimal setup, and are easy to integrate into existing infrastructure. DACs reduce cabling complexity and improve signal integrity, contributing to overall network stability and performance.
- **Copper Cables:** Copper cables, while limited in range, offer a practical solution for short-distance connections within the network rack. Typically used for connections that do not require long-range transmission, such as linking management switches to devices within the same rack, copper cables are durable and cost-effective. In modular architecture, they provide straightforward and reliable connectivity, helping to ensure seamless integration of management and monitoring functions into the overall system. Their use minimizes costs while maintaining essential network operations, supporting efficient resource utilization.

In summary, the combination of AOCs, DACs, and copper cables offers a comprehensive cabling solution that balances cost, performance, and flexibility. Each cabling method is strategically employed to optimize connectivity between compute, network, and service components, ensuring reliable and efficient data center operations.

Note that integrating the leaf switches directly into the compute racks within the accelerator network generally offers an advantage by significantly shortening the cable lengths required to connect the accelerator systems to these switches. Additionally, by minimizing cable clutter, this approach can lead to a more organized and manageable infrastructure, simplifying maintenance and scalability for future network expansions.

Storage Considerations

Storage is the most customer-dependent aspect of the cluster. For the 32-node cluster Reference Design, the storage block is based on 32 2U servers with a minimum of 8x NVMe SSDs and 2x 100 Gbps network connections per server. This configuration can deliver about 20 GB/s large-block random read performance per server (approximately 0.5 TB/s random read per cluster). Storage capacity is very user- and workload-dependent and can be configured both by the size and the number of SSDs used per storage server.

Each storage server has 2x 100 Gbps links to a leaf switch at the top of each storage block rack. Like all the installed systems, the storage servers are connected to redundant power. The BMCs are linked to the Arista 7010 management switches; the 100 Gbps connections to the storage leafs are also linked to the Arista 7010 management switches.

Control Plane Considerations

The control plane consists of several functions. Control plane servers run the control stack, management fabric backbone, control plane fabric spine, storage spine, service leaf, and console fabric. They also provide highly reliable storage for the control plane.

The control plane configuration can be flexible depending on customer requirements, such as the level of fault tolerance required. For smaller clusters, as few as two control plane servers are required. For larger configurations, such as the 32-node cluster in this Reference Design, up to six control plane servers may be needed.

In a configuration including three sets of six control plane servers, a triple-redundant system for the control plane is recommended:

- Each control plane server resides in an independent rack with dedicated leaf switches on each fabric.
- The control plane servers are connected to a redundant power supply.
- The BMC is connected to the Arista 7010 management switches. The 100 Gbps connections to both of the cluster and storage leafs in each rack are also linked to the Arista 7010 switches.

The minimum requirements of the control plane servers are as follows:

- 2x 5th Gen Intel Xeon Scalable processors
- 1 TB DRAM
- 1x 2 TB SSD for storage
- 4x 200 GbE connections per client
- 2x 100 GbE connections per client

Servers should be sized based on your load requirements.

PDU Considerations

The arrangement of PDUs is critical to delivering efficient power management and redundancy. Each rack within the architecture is designed to house four compute nodes, and the power requirements for these nodes necessitate a robust PDU setup. To achieve high availability and redundancy, each rack is equipped with two PDUs. This dual-PDU configuration allows for load balancing, and if one PDU fails, the other can continue to supply power to the nodes, minimizing downtime and maintaining operational continuity.

Figures 3 and 4 show power-dense configurations. Alternative configurations are feasible but beyond the scope of the paper.

The PDUs are strategically placed within the racks to optimize space and facilitate easy access for maintenance. They are connected to the nodes and other critical components, such as switches, through dedicated power cables. This setup not only supports the power needs of the nodes but also accommodates the additional power requirements of the management switches and other ancillary devices within the rack. The PDUs are designed to handle the power density of the rack, keeping the power supply stable and sufficient for all components.

Furthermore, the PDUs are integrated into the overall power management system of the data center, allowing for centralized monitoring and control. This integration enables real-time tracking of power usage and alerts for any anomalies, enabling proactive management of power resources.

The PDUs used in this configuration operate on a three-phase power supply with a voltage of 400/415V and can handle a current of up to 63A, or 48A when derated. The rack devices include server and switch devices. The power requirement for a rack depends on the number of devices in the rack. Table 3 provides power requirements for racks with Intel Gaudi 3 accelerator-based servers, storage and cluster servers, and switches.

Table 3. Power Requirements

Device	Watts
Intel® Gaudi® 3 Accelerator-based Nodes	10,000
Switches	100 to 2,200
Storage and Control Plane Servers	850 to 1,500

5. Software

The software stack for an Intel Gaudi 3 AI accelerator cluster is crafted to simplify the deployment and management of the cluster. This encompasses tasks such as cluster provisioning, configuration, upgrades, monitoring, visualization, network setup, performance and system health monitoring, and user management. Table 4 on the following page outlines the software components for the 32-node cluster Reference Design. It's important to note that Intel does not supply the software stack; however, it can be constructed using open-source tools or OEM solutions like Omnia by Dell AI Factory. For more information, review the detailed deployment instructions for enterprise inferencing solutions.

Table 4. Software Components

Category	Functions and Tools
Virtualization, Containerization, and Orchestration	▪ Container foundation: **Kubernetes (K8s)** provides a robust platform for automating deployment, scaling, and operations of application containers across clusters of hosts. It offers container-centric infrastructure that allows developers to build and manage applications with agility and efficiency. ▪ K8s cluster setup: **Kubespray** is a tool that simplifies the deployment of Kubernetes clusters by providing a set of Ansible playbooks. It supports a wide range of configurations and environments, making it easy to set up and manage Kubernetes clusters in production. ▪ K8s device plugin: The **Gaudi K8s** plugin enables Kubernetes to manage and schedule workloads that require specific hardware accelerators, such as Intel® Gaudi® accelerators. This plugin enables applications to access specialized hardware for improved performance and efficiency. ▪ K8s virtualization: **KubeVirt** extends Kubernetes by adding virtualization capabilities, allowing users to run virtual machines alongside containers. This integration provides a unified platform for managing both containerized and virtualized workloads, enhancing flexibility and resource utilization.
Authentication and Key Management	▪ Key storage and management: **HashiCorp Vault** is a tool for securely storing and accessing secrets, such as API keys, passwords, and certificates. It pro-vides a centralized solution for managing sensitive information, helping to protect data and control access. ▪ AAA, LDAP, DNS, and key management: **Freeipa** is an integrated security information management solution that provides authentication, authorization, and account management (AAA) services. It also includes LDAP, DNS, and key management capabilities, offering a comprehensive approach to identity and access management.
Network	▪ K8s networking and security: **Calico** is a networking and security solution for Kubernetes that provides high-performance, scalable networking with built-in security policies. It enables highly secure communication between containers and offers features like network isolation and policy enforcement. ▪ Load balancing: **MetalLB** is a load balancer implementation for Kubernetes that provides network load balancing for services running within a cluster. It helps to ensure that incoming traffic is distributed evenly across multiple instances, improving application availability and performance. ▪ Network source of truth/IP address management (IPAM): **NetBox** is a tool for managing and documenting networks, including IPAM. It serves as a source of truth for network configurations, helping teams to plan, track, and manage network resources effectively.
Storage	▪ Storage: **WEKA** is a high-performance storage solution designed for modern data-intensive applications. It provides scalable and efficient storage capabilities, enabling organizations to handle large volumes of data with ease and reliability.
Provisioning and Management	▪ Provisioning and configuration management: **Ansible** is an open-source automation tool that simplifies provisioning, configuration management, and application deployment. It uses simple, human-readable YAML files to define tasks, making it easy to automate complex processes. ▪ Network Time Protocol (NTP): **Chrony** is a versatile NTP implementation that synchronizes the system clock with external time sources. It is designed to work well in various network environments and provides accurate timekeeping for applications and systems. ▪ Kubeflow: **Kubeflow** is an open-source platform for machine-learning operations (MLOps) on Kubernetes. It provides tools and frameworks for building, deploying, and managing ML workflows, enabling teams to scale their ML efforts efficiently.
Monitoring	▪ Real-time metrics: **Prometheus** is a monitoring and alerting toolkit that collects and stores real-time metrics from applications and infrastructure. It provides powerful querying capabilities and integrates with Grafana for visualization, helping teams to monitor system performance effectively. ▪ Visualization: **Grafana** is an open-source platform for visualizing metrics and logs from various data sources. It offers customizable dashboards and alerts, enabling teams to quickly gain insights into system performance and troubleshoot issues. ▪ Centralized log management: **Loki** is a log aggregation system that collects and stores logs from applications and infrastructure. It integrates with Grafana for visualization, providing a centralized solution for managing and analyzing logs. ▪ Log aggregation: **Promtail** is an agent that collects logs from various sources and forwards them to Loki for aggregation. It supports multiple log formats and provides filtering capabilities, so that relevant log data is captured and stored.
Operating System	▪ Libraries, compiler, runtime: **Intel® Gaudi® software** provides libraries, compilers, and runtime environments. It enables developers to build and run applications that leverage the unique capabilities of Intel Gaudi accelerators. ▪ Node OS: **Ubuntu Linux** is a popular operating system for running Kubernetes nodes. It offers a stable and highly secure environment for containerized applications and provides a wide range of tools and packages, making it a versatile choice for deploying and managing Kubernetes clusters. Please review the list of supported operating systems for Intel Gaudi 3 AI accelerator-based clusters.

6. Summary

As GenAI becomes increasingly integrated across industries, Intel Gaudi 3 AI accelerators emerge as a pivotal solution, offering the high performance required for rapid training and inference at scale. These accelerators leverage open-source software and industry-standard Ethernet connections to help reduce the cost of AI solutions, making them accessible to companies of all sizes. By providing efficient and scalable AI capabilities, Intel Gaudi 3 accelerators support the growing demand for advanced AI applications, ensuring that businesses can harness the power of AI without incurring prohibitive costs.

In the realm of AI data center adoption, modular and unified architectures serve as key references for rapid implementation. Modular architecture offers a balanced approach to compute, network, and service integration, providing advantages in space, cost, and performance, particularly for established data centers with existing infrastructure. It allows for flexibility and scalability, facilitating the seamless integration of new systems into current operations. Conversely, unified architecture is tailored for new data centers that require a complete, integrated solution that can simplify deployment, optimize space, and reduce costs. The Reference Design further streamlines cluster deployment by offering prescriptive guidance for selecting and configuring system components, enabling enterprises to swiftly innovate and advance their AI journey.

Appendix A: L3 Scale-Out Accelerator Fabric Configuration

The scale-out fabric on each Intel Gaudi 3 accelerator-based node consumes 24 IP addresses. For L3 network scale-out to function, these 24 interfaces require configuration on the compute node. This involves generating a `gaudinet.json` file in `/etc/`. This appendix provides suggested IP assignment schemes, steps to manually generate the `.json` file, and an example script to automatically generate the configuration, which can be time-consuming otherwise.

Example Scale-Out Fabric Switch

The following example can help administrators in configuring the Intel Gaudi 3 accelerator-based scale-out infrastructure. Administrators may use different schemes.

Note: Depending on system size and workload traffic patterns, congestion may lead to packet loss and, therefore, network performance degradation. The following reference switch configurations recommend enabling Priority Flow Control (PFC) to alleviate such problems.

Reference IP Address Assignment: A `/16` private address space is assigned to each ply. This document assumes the IP address `10.208.0.0/16` for ply0, which implies the following address space for the three plys:

- `10.208.0.0/16` for Ply 0
- `10.209.0.0/16` for Ply 1
- `10.210.0.0/16` for Ply 2

Switch IP Address Assignment Scheme: IP addresses for leaf switch interfaces connecting to the cluster's NICs can be assigned with the following rule:

`10.(starting_second_octect+ply_id).(leaf_switch_id).(2+port_seq_idx4)/30`

Where:

`starting_second_octet` = administrator's choice; 208 is used in this document

`ply_id` = 0,1,2

`leaf_switch_id` = the sequence number (0,1,...,N) from left to right in the leaf/spine topology in a given ply

`port_seq_id` = the sequence number (0,1,...,N) of 200 Gb/s interfaces with Intel Gaudi 3 accelerator-based servers (ignore interfaces to spines or unused interfaces)

Examples:

1. Assume the first switch interface connected to an Intel Gaudi 3 accelerator-based server in the left-most leaf switch in ply0 is Ethernet2/1:

   `starting_second_octet` = 208

   `ply_id` = 0

   `leaf_switch_id` = 0, because this is the first (left-most) switch

   `port_seq_id` = 0, because this is the first interface

   So, the IP address assigned to Ethernet2/1 is: 10.(208+0).0.(2+0x4)=10.208.0.2/30.

2. Assume the seventh switch interface connected to an Intel Gaudi 3 accelerator-based server in the second leaf switch in ply1 is Ethernet4/5:

   `starting_second_octet` = 208

   `ply_id` = 1

   `leaf_switch_id` = 1 for the second switch

   `port_seq_id` = sixth or seventh interface

   So, the IP address assigned to Ethernet4/5 is: 10.(208+1).1.(2+6x4)=10.209.1.26/30

Note: The ports on an Arista 7060DX4 are in two rows: the top row has odd-numbered ports, and the bottom row has even-numbered ports. For convenient cable management, connect the bottom ports to Intel Gaudi 3 accelerator-based servers and the top ports to spine switches. In this case, the interfaces to servers are in this order: Ethernet2/1, 2/3, 2/5, 2/7, 4/1, 4/3, 4/5, 4/7... .

Scale-Out IP Address Configuration

The Intel Gaudi 3 accelerator-based servers require the file `/etc/gaudinet.json` to configure the Intel Gaudi 3 accelerator network interfaces. This file includes the NIC MAC address, IP address, subnet mask, and gateway MAC address for the 24 1x 200 GbE connections in the following format:

```
{
“NIC_NET_CONFIG”: [
{
“NIC_MAC”: “b0:fd:0b:d9:22:4d”,
“NIC_IP”: “10.208.0.1”,
“SUBNET_MASK”: “255.255.255.252”,
“GATEWAY_MAC”: “e8:b2:65:79:b8:38”
},
{
“NIC_MAC”: “b0:fd:0b:d9:22:5b”,
“NIC_IP”: “10.209.0.1”,
“SUBNET_MASK”: “255.255.255.252”,
“GATEWAY_MAC”: “ec:8a:48:43:c9:81”
},
{
“NIC_MAC”: “b0:fd:0b:d9:22:5c”,
“NIC_IP”: “10.210.0.1”,
“SUBNET_MASK”: “255.255.255.252”,
“GATEWAY_MAC”: “ec:8a:48:44:3b:41”
},
…
]
}
```

Manual Generation of the gaudinet.json Network Configuration File

An example helper script to generate the `gaudinet.json` is available at the end of this section. However, the file can be generated manually using the following steps:

1. Get the Intel Gaudi accelerator network bus IDs with this command:

   ```
   hl-smi -Q module_id,bus_id -f csv,noheader | sort
   ```

   The output shows the bus, driver, function (BDF) for each accelerator, as shown in the following example:

   ```
   0, 0000:19:00.0
   1, 0000:3b:00.0
   2, 0000:5d:00.0
   3, 0000:4c:00.0
   4, 0000:db:00.0
   5, 0000:cb:00.0
   6, 0000:9b:00.0
   7, 0000:bb:00.0
   ```

2. Get the three MAC addresses (one address for each ply) for each Intel Gaudi accelerator node, starting with module 0, by replacing `{BDF}` in the following command with the BDF found in Step 1:

   ```
   cat /sys/bus/pci/drivers/habanalabs/{BDF}/net/*/address | sort
   ```

   For example:

   ```
   cat /sys/bus/pci/drivers/habanalabs/0000:4d:00.0/net/*/address | sort
   ```

   This generates the following three MAC addresses:

   ```
   b0:fd:0b:d9:22:4d #Ply0 MAC
   b0:fd:0b:d9:22:5b #Ply1 MAC
   b0:fd:0b:d9:22:5c #Ply2 MAC
   ```

   Repeat for nodes one through seven to get all 24 MACs.

3. For the NICs from the list, associate MAC addresses with IP addresses:

   ```
   10.(starting_second_octect+ply_id).(leaf_switch_id).(2+port_seq_idx4)/30
   ```

   Where:

   `starting_second_octet` = user’s choice, 208 in this example

   `ply_id` = 0,1,2

   `leaf_switch_id` = the ID of the connected leaf switch

`port_seq_id` = the sequence number for the 200 Gb/s interfaces across all the servers connected to the same leaf switch. Each server has eight interfaces connected to each of the three leaf switches, so the current server may not be the first server connected to a leaf switch.

Examples:

a) For the first server connected to the first leaf switch, the NIC in node 0 for ply0 is assigned the following IP address:

`10.(208+0).(0).(1+0x4)/30 = 10.208.0.1/30`

b) For the second server connected to the second leaf switch, the NIC in node 2 in ply1 is assigned the following IP address:

`10.(208+1).(1).(1+10x4)/30 = 10.209.1.41/30`

The `port_seq_id` is 10 because this is the second server connected to the switch, whose first eight Intel Gaudi accelerator-facing interfaces are connected to the first server, and this NIC is node 2, so 8+2=10.

4. The subnet mask is 255.255.255.252 for a point-to-point /30 network.
5. The gateway MAC address is the address that belongs to the connected switch. It can be acquired directly from the switch or by using the command `lldpctl show neighbor` on the server. The `lldpd` package may need to be installed to use the command.

Example gaudinet.json Generator Script

This script can be saved in a `.sh` file and used to generate the `gaudinet.json` configuration file for each node. The script requires a few positional inputs and must run under `sudo` or `root`, and must be executed on the node for which the configuration file is intended. To keep the script short and simple, user input validation and path validation are not implemented. Instructions are embedded in the script at the end.

```
#!/bin/bash
octet2="$1"
leafID="$2"
portSeq="$3"
gwMACS=( $4 $5 $6 )
readonly _scriptName="${0##*/}"
readonly _subnetMask="255.255.255.252"
function generateConfig()
{
local readonly _path="/etc/gaudinet.json"
local readonly _BDFList=`hl-smi -Q module_id,bus_id -f csv,noheader | sort | cut -d ',' -f2`
local macs=" "
local plyID="0"
printf "{\n" > $_path
printf "\t\"NIC_NET_CONFIG\": [\n" >> $_path
for bdf in $_BDFList;
do
    plyID="0"
    let "portSeq++"
    macs=`cat /sys/bus/pci/drivers/habanalabs/${bdf}/net/*/address | sort`
    for mac in $macs;
    do
                NICMAC="$mac"
                address="10.$((octet2 + plyID)).${leafID}.$((2 + portSeq * 4))"
                printf "\t\t{\n" >> $_path
 printf "\t\t\t\"NIC_MAC\": \"${NICMAC}\",\n" >> $_path
 printf "\t\t\t\"NIC_IP\": \"${address}\",\n" >> $_path
 printf "\t\t\t\"SUBNET_MASK\": \"${_subnetMask}\",\n" >> $_path
 printf "\t\t\t\"GATEWAY_MAC\": \"${gwMACS[${plyID}]}\"\n" >>
$_path
 printf "\t\t},\n" >> $_path
 let "plyID++"
 done
 done
 printf "\t]\n" >> $_path
 printf "}\n" >> $_path
 printf "\n$_path config file generated.\n\n"
}
[[ "$EUID" -ne 0 ]] && printf "Error: Script must run as sudo or root!\n"
&& exit 1
if [[ -z $octet2 ]] || [[ -z $leafID ]] || [[ -z $portSeq ]] || [[ -z
${gwMACS[2]} ]]; then
 printf "\nError: Four positional arguments required:\n
 Ex: $_scriptName <Octet 2> <Leaf ID> <Port Sequence> <Gateway
Macs>\n
 Octet 2: Starting value for Second Gaudi IP octet\n
 Leaf ID: Leaf switch number in the sequence.\n
 Port Sequence: Switch port number to start at.\n
 Gateway Macs: The switch MACs space separated for ply 0 1 and 2\n"
exit 1
fi
generateConfig
```

Appendix B: L2 Scale-Out Accelerator Fabric Configuration

The AI cluster network is built on a flat L2 fabric composed of three standalone switches (three-ply design), forming a non-blocking peer topology without routing (for a visual, refer to Figure 5, earlier in this document). All switches operate within VLAN 100, providing a single broadcast domain for high-bandwidth east-west traffic. The fabric is designed for minimal latency and maximum throughput across all compute nodes.

Each AI node is connected to each switch (ply) using two high-speed 800 GbE interfaces, derived from 800 GbE breakout ports (8x 200 GbE per switch). This helps to ensure uniform load distribution and redundancy across the fabric. All ports are configured with MTU 9216 to support Jumbo frames, and Ethernet flow control is enabled bidirectionally to prevent congestion-related drops. The network is L2-only, with no dynamic routing protocols required.

Reference IP Address Assignment

A `/16` private address space is assigned to each ply. This document assumes `10.208.0.0/16` for ply1, so the address spaces for the three plys are as follows:

`10.208.0.0/16` for ply 1

`10.209.0.0/16` for ply 2

`10.210.0.0/16` for ply 3

[1] Intel® Gaudi® 3 accelerator training vs. NVIDIA H100; average performance measured across models: Llama 2 70B & Llama 3 8B. Gaudi 3 inference vs. NVIDIA H100; average performance projected across multiple models, multiple configurations: Llama2 7B & 70B, Falcon 180B, https://github.com/NVIDIA/TensorRT-LLM/blob/main/docs/source/performance/perf-overview.md. Intel results obtained in September 2024. Results may vary. NVIDIA H100 GPU (900 GB/s closed NVLink connectivity) vs. Intel Gaudi 3 accelerator (1200 GB/s open standard RoCE). Source: https://www.intel.com/content/www/us/en/products/details/processors/ai-accelerators/gaudi3.html

[2] The recommended system components in this Reference Design have been tested and validated with Intel® Gaudi® 3 AI accelerators.

[3] See endnote 1.

[4] Source: https://www.arista.com/assets/data/pdf/Datasheets/7060X5-Datasheet.pdf