Technical Paper

Intel and Quanta Cloud Technology (QCT) collaborate to build an Open Hybrid Platform for AI Workloads

# Converging Paradigms: Architecting a Hybrid and Open Platform for Unified HPC and AI Workloads

**Intel and QCT jointly conducted a Proof of Concept (PoC) evaluation of Intel® Gaudi® 3 PCI Express* (PCIe*) AI accelerators on the QCT* DevCloud platform, built upon the QCT AI POD* reference architecture to demonstrate how a converged, hybrid system can effectively support unified HPC and AI workloads within a single, open infrastructure**


## Authors

### Intel Corporation

Carrie Chen, Andy Shih

Maciej Cichocki, Michal Prostko

Anna Alberska, Bo Yuan Yao

Zhiqi Tao, Dunni Aribuki,

Lakshman Chari

### QCT

Stephen Chang

Ching-Tao Lee, Reid Lin

Min-Hua Chan

### National Taiwan University

Jyh-Pin Chou


## Table of Contents



## Executive Summary


High-Performance Computing (HPC) and Artificial Intelligence (AI) have historically evolved as distinct computing paradigms. Traditional HPC systems are optimized for tightly coupled, latency-sensitive simulations running on bare metal, while modern AI platforms emphasize massively parallel training, elastic resource allocation, cloud-native architectures, and microservice-based inference. Today, this separation is increasingly misaligned with real-world requirements.

Scientific research, industrial engineering, and enterprise digital transformation workflows now routinely combine simulation, data analytics, machine learning, generative AI, and agent-driven reasoning within a single end-to-end pipeline. While some environments maintain domain isolation for operational or security reasons, converged platforms are increasingly adopted where workflow integration and resource pooling outweigh segmentation requirements.

This technical paper presents the QCT AI Platform On Demand (QCT AI POD) Reference Architecture, a unified, open, and hybrid platform designed to converge HPC, AI, and emerging Generative and Agentic AI workloads within a single, operationally cohesive system. The architecture integrates Slurm-managed bare-metal HPC workloads and Kubernetes-based AI services on shared infrastructure, supported by unified authentication, multi-tier storage, and heterogeneous compute resources spanning CPUs, GPUs, and AI accelerators.

To address the rapid adoption of generative AI, the platform incorporates Intel® AI for Enterprise RAG, a modular, cloud-native framework that delivers composable microservices, higher-level megaservices, and validated reference blueprints for Retrieval-Augmented Generation (RAG), multimodal intelligence, and agentic automation. This integration enables organizations to move from AI experimentation to production without sacrificing performance, openness, or portability.

The architecture is realized and validated through QCT DevCloud, a real-world deployment that spans multiple generations of QCT servers with Intel Xeon processors and heterogeneous accelerators including NVIDIA* GPUs and Intel Gaudi 3 AI accelerators. QCT DevCloud demonstrates how converged HPC and AI workloads can execute concurrently within a shared platform while maintaining performance isolation, governance, and operational simplicity.

To illustrate the practical implementation of complete Generative and Agentic AI workflows, QCT developed QChat, a compact yet capable multimodal chatbot built on Intel AI for Enterprise RAG. QChat integrates large language model inference, RAG-based document retrieval, Whisper-based speech processing, image generation, and MCP-driven agentic tool invocation. Running natively on QCT DevCloud across hybrid compute backends, QChat demonstrates the platform's readiness to support modern AI applications that require coordinated orchestration of multiple intelligent components.

Together, the QCT AI POD architecture, its DevCloud deployment, and the QChat demonstration establish a comprehensive and future-proof blueprint for converged HPC–AI infrastructure—enabling organizations to unify simulation, training, inference, and agentic reasoning within a single, scalable, and operationally consistent platform.

# Introduction

## The Evolving Computational Landscape

High-performance computing (HPC) and artificial intelligence (AI) have historically followed distinct architectural trajectories. Traditional HPC systems are engineered for tightly coupled MPI-based simulations and deterministic, bare-metal scheduling, whereas AI platforms prioritize massively parallel deep learning, cloud-native elasticity, and microservice-oriented inference pipelines. Today, these boundaries are rapidly dissolving. Leading scientific and industrial workloads increasingly integrate simulation, statistical learning, multimodal perception, and agentic decision-making within unified computational workflows.

At the same time, the underlying hardware landscape is becoming inherently heterogeneous. Modern data centers routinely deploy multiple generations of CPUs, GPUs, and purpose-built AI accelerators—often sourced from multiple vendors—to keep pace with fast-evolving AI models and diverse workload requirements. This shift is evident in broader industry trends, including emerging generative AI reference architectures, Ethernet-based cloud-native AI fabrics, and open hardware ecosystems that emphasize flexibility, interoperability, and workload portability across silicon platforms.

Within this context, traditional siloed infrastructure models—where separate clusters are dedicated exclusively to HPC or AI—are increasingly misaligned with operational realitiesFigure 1. Such designs exacerbate resource fragmentation, limit cross-domain workload integration, increase operational complexity, and ultimately drive higher Total Cost of Ownership (TCO). Addressing these challenges requires a unified, open, and hybrid platform capable of supporting next-generation workloads across the full lifecycle, from simulation and training to inference and intelligent decision-making.

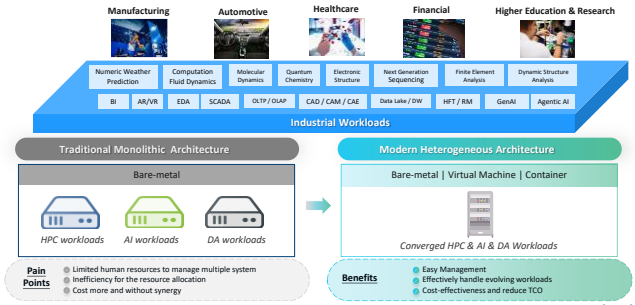



Figure 1. Converging industrial workloads drive the shift from siloed HPC/AI/DA clusters to a unified heterogeneous platform.

## Modern Workload Requirements: Scalability, Composability, Flexibility, and Openness

High-Performance Computing (HPC) and Artificial Intelligence (AI) have historically followed distinct architectural trajectories. Traditional HPC systems are engineered for tightly coupled MPI-based simulations and deterministic, bare-metal scheduling, whereas AI platforms prioritize massively parallel deep learning, cloud-native elasticity, and microservice-oriented inference pipelines. Today, these boundaries are rapidly dissolving. Leading scientific and industrial workloads increasingly integrate simulation, statistical learning, multimodal perception, and agentic decision-making within unified computational workflows.

Modern AI, HPC, and agentic workloads impose a new set of requirements on infrastructure platforms, extending beyond raw performance to include scalability, composability, flexibility, and openness.

### Scalability

The platform must scale horizontally across nodes and vertically across successive generations of CPUs, GPUs, and AI accelerators. Contemporary reference architectures increasingly emphasize scalable CPU–accelerator topologies, multi-tier networking fabrics, and distributed orchestration models to accommodate the exponential growth of AI model sizes, training datasets, and concurrent workloads.

### Composability

Beyond scalability, modern HPC and AI environments require composability—the ability to modularly assemble, combine, and reuse compute, storage, orchestration services, and application frameworks across diverse workloads.

From a software-defined architecture perspective, composability manifests as the platform's ability to provide:

- Composable orchestration layers, in which Kubernetes-based services, Volcano, Slurm, and workflow engines dynamically coordinate resources according to workload intent.
- Composable AI frameworks, such as Intel AI for Enterprise RAG, that expose fine-grained microservices (for example, LLM inference, embedding generation, retrieval, reranking, and vector database integration) alongside higher-level megaservices (for example, ChatQnA, HybridRAG, GraphRAG, and productivity-oriented pipelines) that can be flexibly assembled into end-to-end applications.
- Composable hybrid workflows, enabling simulation, training, inference, and agentic reasoning to be chained into a single operational execution graph.
- Composable resource abstraction, in which underlying hardware heterogeneity (for example, Intel Xeon CPUs, GPUs, Intel Gaudi accelerators) is abstracted at the orchestration and runtime layer, allowing users to focus on application logic rather than device-specific details.
- Composable data services, where shared storage layers can be integrated across multiple pipelines without requiring structural redesign.

This model reflects a broader industry shift toward modular, service-oriented infrastructure, where platforms are constructed from interoperable building blocks rather than delivered as monolithic stacks. Across cloud-native frameworks, HPC workflow systems, and emerging GenAI platforms, there is growing consensus that future computing environments are assembled dynamically rather than prescribed statically.

At the software layer, composability directly improves reuse, accelerates application development, supports heterogeneous hardware integration, and enables rapid adoption of new models, tools, and frameworks—capabilities that are increasingly essential for modern converged HPC and AI environments.

### Flexibility

Flexibility describes the platform's ability to:

- Support multiple accelerator vendors and hardware generations (for example, 4th Gen Intel Xeon processor

Scalable family, Codename Sapphire Rapids

- and Intel Xeon 6 CPUs, NVIDIA GPUs, Intel Gaudi 3 accelerators).
- Allow workloads to execute on the most appropriate compute resources without requiring pipeline rearchitecture.
- Enable mixed HPC, AI, and GenAI workloads to operate concurrently using shared storage, identity, and management services.
- Adapt to evolving software frameworks and programming models without disruptive infrastructure refactoring.

Such flexibility reduces vendor lock-in, improves resource utilization, and enables incremental evolution of hybrid environments as workloads and technologies change.

### Openness

Open-source software and open standards play a foundational role in enabling sustainable, multi-architecture platforms. They provide:

- Hardware neutrality
- Avoidance of proprietary lock-in
- Interoperability across multi-vendor accelerator ecosystems
- Long-term maintainability and operational transparency
- Alignment with global initiatives emphasizing digital sovereignty and open innovation

Across both HPC and AI domains, open frameworks have repeatedly proven to be powerful enablers of innovation—particularly in environments that must integrate diverse hardware architectures and rapidly evolving software ecosystems.

## Overview of the QCT AI POD Architecture

The following figure *Figure 2* illustrates the QCT AI Platform On Demand (QCT AI POD), a pre-validated, on-premises system architecture designed to support converged HPC and AI workloads through best-practice hardware and software integration. The platform is purpose-built to address the evolving requirements of modern HPC, AI, and Generative AI infrastructure. Rather than optimizing for a single workload category, QCT AI POD is positioned as a unified platform capable of supporting heterogeneous and converged workloads within a shared, operationally consistent environment.

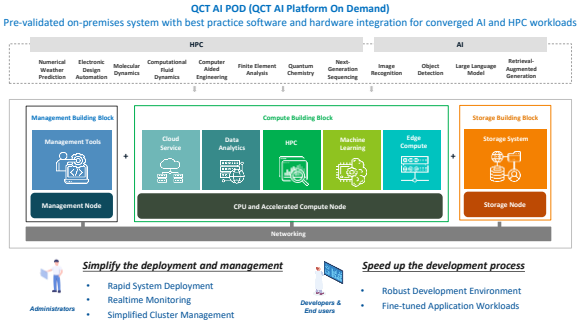



*Figure 2. The illustration of QCT AI POD*

The platform is organized into three modular building blocks—Management, Compute, and Storage—interconnected by a multi-rails networking layer. Centralized management services provide system provisioning, monitoring, and management. The compute layer integrates CPU-based and accelerated nodes to support cloud services, analytics, HPC, machine learning, and edge-extendable workloads, enabling each workload to execute on the most appropriate hardware resources. The storage layer delivers scalable, shared data services to support data-intensive and multi-stage workflows across the platform.

By adopting a modular, building-block design and leveraging best-practice hardware and software integration, QCT AI POD enables scalable deployment, flexible workload composition, and consistent operations across heterogeneous infrastructure, simplifying both system management and application development for converged HPC and AI environments.

The QCT AI POD architecture is designed around a set of core principles that guide its ability to support converged HPC and AI workloads:

- **Scalability**: Enables horizontal expansion across compute nodes and accelerators, as well as vertical scaling to accommodate increasing workload complexity and data volume.
- **Composability**: Allows compute, storage, orchestration, and AI services to be modularly assembled into flexible, end-to-end workflows.
- **Flexibility**: Supports heterogeneous hardware platforms and mixed HPC, AI, and Generative AI workloads within a single operational environment.
- **Openness**: Leverages open standards and open-source software to ensure portability, interoperability, and long-term sustainability.
- **Multi-User Collaboration**: Provides a shared, secure platform that enables teams to collaborate efficiently across common compute and data resources.

The following sections detail how the architecture is realized in practice and validated through the QCT DevCloud deployment, Intel Gaudi 3 performance benchmarking, and the QChat as an end-to-end multimodal Generative and Agentic AI use case.

## Architectural Vision and Design Principles of the QCT AI POD

Building on the evolving computational landscape and workload requirements discussed in Section 1, this section details the architectural vision and core design principles that guide the QCT AI POD. The architecture is explicitly workload-driven, designed to optimize system composition based on the data flows, execution models, and performance characteristics of modern HPC, AI, and Generative AI applications.

As *Figure 3* illustrated in the QCT AI POD architecture overview, the platform is structured across three tightly integrated layers—Workloads, System Platform, and Infrastructure—enabling a unified yet modular foundation for heterogeneous computing.

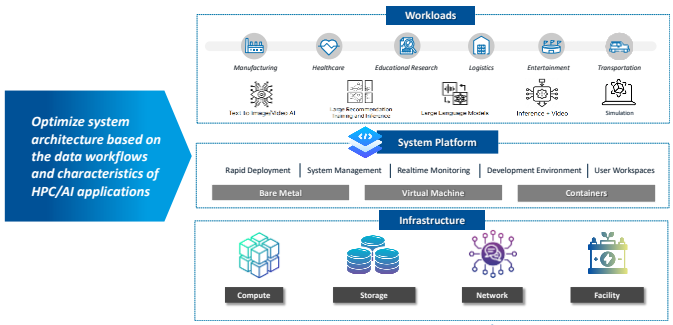



*Figure 3. Workload-Driven Infrastructure for HPC/AI workloads*

## Scalability: Supporting Growth Across Data, Models, and Execution

Modern HPC and AI workloads scale simultaneously across multiple dimensions, including dataset size, model complexity, compute parallelism, and inference demand. The QCT AI POD is architected to scale proportionally across compute, storage, and network domains to sustain this growth without architectural redesign.

**Compute scalability** is achieved through:

- Native support for heterogeneous compute resources, including multi-core CPUs, GPUs, and Intel Gaudi 3 AI accelerators
- Horizontal scale-out across nodes and racks, managed through Slurm for HPC workloads and Kubernetes/Volcano for AI and service-oriented workloads
- High utilization across mixed workloads through coordinated scheduling and resource isolation

**Storage scalability** follows a tiered design aligned with workload I/O behavior:

- High-performance NVMe* tiers for scratch space, fine-tuning, and inference serving
- Distributed parallel file systems for checkpointing and large-scale training datasets
- Scalable object storage for dataset lakes, embeddings, and generative outputs. This design supports petabyte-scale growth while enabling efficient data lifecycle management for training, RAG pipelines, and model iteration.

**Network scalability** underpins distributed execution:

- Ultra-high-bandwidth 200/400 Gb/s InfiniBand*-class or RoCE*-based fabrics for AI training and HPC communication
- Scalable topologies that support both tightly coupled and loosely coupled workloads
- Logical traffic segmentation to isolate compute, storage, and platform services

Together, these capabilities ensure that the platform scales predictably as workloads increase in size, concurrency, and complexity.

## Composability: A Modular, Workflow-Driven Architecture

Rather than prescribing a fixed execution model, the QCT AI POD adopts a composable architecture, allowing systems and workflows to be assembled dynamically from modular building blocks.

Composability is realized across several dimensions:

- **Software services**: Intel AI for Enterprise RAG microservices and megaservices provide modular components for LLM inference, embeddings, retrieval, reranking, multimodal processing, and agentic tool use.
- **Execution environments**: Workloads can run natively on bare metal (HPC simulations), or within containers (AI services and application frameworks), depending on performance and isolation requirements.
- **Orchestration layers**: Kubernetes with Volcano, and Slurm coexist to coordinate hybrid workloads under a unified platform.
- **Data services**: Workflows can select appropriate storage tiers transparently, without changes to application logic.

This composable approach enables advanced pipelines such as multimodal RAG, simulation-to-AI feedback loops, and hybrid HPC–LLM workflows to be constructed as cohesive, end-to-end execution graphs rather than isolated jobs.

## Flexibility: Optimizing Execution Across Diverse Workloads

HPC, AI training, inference, and agentic workloads exhibit fundamentally different execution characteristics. The QCT AI POD is designed to flexibly map each workload to the most appropriate compute, network, and storage resources.

- HPC simulations benefit from low-latency MPI communication and tightly coupled execution.
- AI training workloads require high-bandwidth fabrics and large-scale distributed tensor operations.
- Inference and agentic workloads prioritize throughput, responsiveness, and predictable latency.
- Edge-oriented inference demands localized execution and efficient model caching.

To support this diversity, the platform provides:

- Hardware abstraction through containerization and virtualization
- Flexible network fabrics optimized for both latency-sensitive and bandwidth-intensive workloads
- Tiered data placement aligned with access patterns and persistence requirements

This flexibility enables efficient workload placement across cloud-scale training, datacenter fine-tuning, and edge deployment scenarios without architectural refactoring.

## Openness: Ensuring Interoperability and Long-Term Viability

Openness is a foundational principle of the QCT AI POD, ensuring interoperability across vendors, frameworks, and deployment environments.

The architecture is built entirely on open technologies, including:

- Open-source orchestration and scheduling (Kubernetes, Slurm, Volcano)
- Open identity and access management (Keycloak*, LDAP*)
- Open storage platforms (Ceph*, Lustre*, BeeGFS*)
- Open AI frameworks and ecosystems (Intel AI for Enterprise RAG, Hugging Face*, vLLM*, LangChain*, Triton*)

This open design avoids proprietary lock-in, preserves architectural optionality, and enables continuous innovation as AI models, accelerators, and software frameworks evolve.

### Multi-User Collaboration: Shared Infrastructure, Secure Access

The QCT AI POD is designed as a shared, multi-users environment supporting collaborative research and development.

Key capabilities include:

- Centralized authentication and role-based access control
- Web-based user environments such as JupyterHub and Open OnDemand
- Secure, role-aware access to HPC jobs, containers, and virtualized workloads

This model enables multiple teams to innovate concurrently on shared infrastructure while maintaining governance, isolation, and operational simplicity.

### Infrastructure Foundations: Compute, Storage, Network, and Facility

The architectural principles above are anchored in a robust physical foundation:

- Compute: Scalable nodes integrating CPUs, GPUs, and Gaudi-class accelerators with advanced scheduling for high utilization
- Storage: Multi-tier architectures spanning NVMe, parallel file systems, and object storage
- Networking: RDMA-enabled fabrics with ultra-high bandwidth and low latency
- Facility: High-density rack designs supporting advanced cooling, power delivery, and future accelerator generations

This foundation ensures the platform remains performant, reliable, and future-ready.

### End-to-End Alignment with the AI Lifecycle: Cloud, Datacenter, and Edge

Finally, the QCT AI POD aligns infrastructure design with the full AI lifecycle:

- Cloud environments support large-scale pre-training and global data processing.
- Datacenters enable fine-tuning, RAG pipelines, and enterprise AI deployment.
- Edge environments deliver real-time inference and streaming analytics close to data sources.

This end-to-end alignment allows models to be trained centrally, adapted efficiently, and deployed ubiquitously with consistent performance and operational control.

## System Architecture and Software Stack of the QCT AI POD

While earlier sections establish why a unified, open, and composable HPC–AI platform is required and what principles guide its design, this section describes how those principles are concretely realized in the QCT AI POD through a large-scale reference system architecture and a comprehensive software stack.

The QCT AI POD architecture is intentionally layered, separating concerns across physical infrastructure, system platforms, and application services, while ensuring tight integration across compute, storage, networking, and orchestration domains.

### Large-Scale Cluster Reference Architecture

The *Figure 4* illustrates a scale-out HPC and AI cluster designed to support tightly coupled HPC simulations, distributed AI training, high-throughput inference, and shared multi-user services within a single fabric.

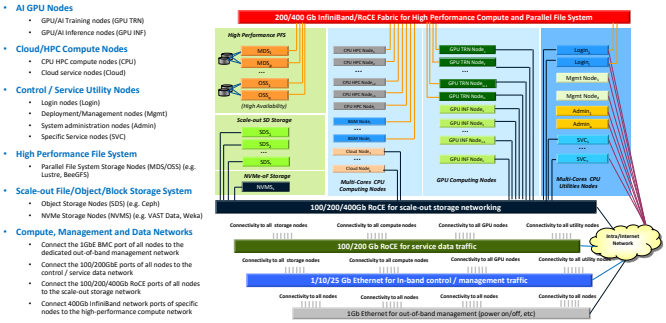



*Figure 4. Reference Architecture for the Large-Scale HPC/AI Cluster System that integrates heterogeneous computing resources via a multi-tier network fabric*

#### Compute Domain: Heterogeneous and Role-Oriented Nodes

The compute layer is composed of multiple node classes, each optimized for a specific execution role:

- **GPU / AI Accelerator Nodes**

  Dedicated nodes equipped with high-performance accelerators (for example, GPUs or Intel Gaudi AI accelerators) support distributed AI training, large-scale inference, and tensor-intensive HPC workloads.

- **CPU-Based HPC Cloud Compute and Big Memory (BGM) Nodes**

  Multi-core CPU nodes support traditional MPI*-based simulations, data analytics, preprocessing pipelines, and cloud-native services.

- **Control and Utility Nodes**

  Login, management, administration, and service nodes host cluster control planes, orchestration services, authentication, monitoring, and user access gateways.

This role-based separation enables performance isolation, predictable scaling, and operational clarity while maintaining a unified scheduling and management framework.

#### Storage Architecture: Tiered and Software-Defined

Storage is architected as a multi-tier, software-defined subsystem aligned with workload I/O characteristics:

- High-Performance Parallel File Systems (PFS)

  Metadata servers (MDS) and object storage servers (OSS) provide high-throughput, low-latency access for HPC simulations, AI training checkpoints, and large-scale data processing.

- Scale-Out Software-Defined Storage (SDS)

  SDS layers deliver elastic capacity for datasets, model artifacts, embeddings, and generative outputs.

- NVMe-oF and Local NVMe Tiers

  NVMe-based tiers accelerate scratch space, fine-tuning, and inference workloads where latency and IOPS are critical.

This tiered design enables efficient data placement, lifecycle management, and concurrent access by heterogeneous workloads.

### Networking Architecture: Purpose-Built, Multi-Fabric Design

Networking is a critical enabler of convergence. The QCT AI POD adopts a multi-fabric networking strategy that aligns traffic patterns with performance requirements:

- *200/400 Gb InfiniBand or RoCE Fabric*

  Dedicated to high-performance compute and parallel file system traffic, enabling RDMA-based communication for AI training and HPC workloads.

- *100/200/400 Gb RoCE Fabric*

  Used for scale-out storage networking, ensuring sustained throughput between compute and storage tiers.

- *Service and Management Networks*

  Separate Ethernet fabrics support service traffic, control-plane operations, and management workflows.

- *Out-of-Band Management Network*

  A dedicated 1 Gb Ethernet network provides power control, firmware access, and hardware monitoring.

This strict separation prevents cross-domain interference, improves fault isolation, and simplifies operational management at scale.

## Modern AI/HPC Software Platform

The modern AI/HPC software platform must support extreme heterogeneity across workloads, hardware architectures, development methodologies, and execution environments. Figure*Figure 5* 5 illustrates a layered software stack that unifies HPC, AI/ML, data analytics, and cloud-native development into a single cohesive platform. This modular design enables organizations to execute diverse workloads—MPI jobs, LLM training, RAG pipelines, data preprocessing, agentic workflows, and online inference—on the same shared infrastructure.

The QCT AI POD platform provides four key layers: Cloud Infrastructure, Development Platform, Application Services, and System Management. Together, these layers enable efficient resource utilization, streamlined development, reproducibility, and high operational reliability.

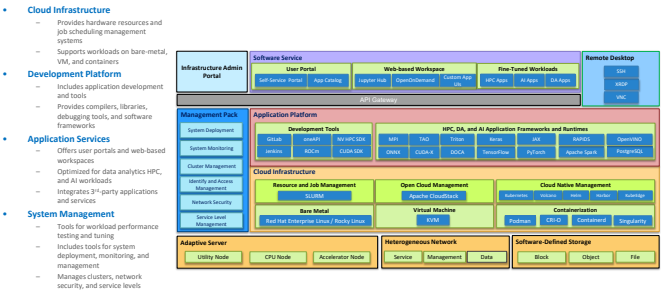



*Figure 5. Modern HPC/AI Application Service Platform for a unified platform that runs HPC/AI applications across bare metal, VMs, and containers with centralized management.*

### *Cloud Infrastructure Layer*

The cloud infrastructure layer provides the foundational compute, network, and storage environment upon which all HPC, AI, and cloud-native workloads run. It combines traditional HPC schedulers, modern container orchestration, and cloud virtualization technologies to support diverse execution models.

#### Operating System Foundation

- Red Hat Enterprise Linux* (RHEL*) and Rocky Linux* serve as the enterprise-grade Linux distributions for both compute and management nodes.
- These OSes provide kernel optimizations for NUMA balancing, high-performance networking (RDMA*, OFED*), cgroup resource isolation, and tuned CPU/GPU scheduling.

#### Bare-Metal Resource Management

- Slurm is used for job scheduling, resource allocation, job accounting, and QoS management across HPC and bare-metal AI workloads.
- Slurm ensures uniform control over CPU/GPU resources regardless of workload type (training, simulation, batch inference, and so on.).

#### Virtualization & Cloud Platform Integration

- Kernel-based Virtual Machine* (KVM*) offers lightweight virtualization for running legacy workflows, isolated services, or custom OS environments.
- Apache CloudStack* provides an IaaS layer enabling dynamic provisioning of virtual machines.
- This enables VM environments for specialized applications, lab environments, or secure service isolation.

#### Containerization & Cloud-Native Runtime

Podman*, Singularity/Apptainer*, and Containerd*/CRI-O* support container execution across HPC and Kubernetes environments.

- Singularity/Apptainer is optimized for HPC workflows with MPI support and secure unprivileged container execution.
- Podman with Containerd provides daemonless, secure OCI*-compatible container runtime for developers and CI pipelines.

#### Orchestration & Cloud-Native Services

- Kubernetes provides large-scale orchestration for AI/ML microservices, model serving, inference endpoints, and cloud-native applications.

- Volcano extends Kubernetes with GPU-aware and gang scheduling capabilities optimized for distributed AI training.
- Helm manages packaged deployments of AI frameworks, model servers, RAG systems, and pipelines.
- Harbor serves as a cloud-native container registry with vulnerability scanning and access controls.
- KubeEdge* extends Kubernetes to support edge inference and device management.

#### Multi-Environment Execution

This hybrid infrastructure natively supports three coexistence models:

- Bare Metal Execution for latency-sensitive HPC workloads
- Virtual Machines for application isolation and legacy environments
- Containers for cloud-native AI services and microservice-based pipelines

This flexibility allows workloads to run in the most appropriate environment without architectural compromise.

### *Development Platform*

This layer provides the complete software ecosystem developers, data scientists, and researchers need to build, optimize, and deploy HPC and AI applications at scale.

#### Source Control and CI/CD

- GitLab* and Jenkins* drive version control, automated builds, model packaging, and continuous deployment pipelines.
- CI/CD integration ensures reproducibility for both HPC simulation code and AI model training workflows.

#### Compiler and Optimization Toolchains

To support hybrid CPU/GPU/accelerator architectures, the platform includes:

*For Intel-based architectures*

- oneAPI: Unified programming model for CPUs, GPUs, XPUs. Includes optimized compilers, libraries and analysis tools to simplify cross-architecture development.

*For AMD architectures*

- ROCm*: GPU compute stack with HIP for CUDA interoperability

*For NVIDIA architectures*

- CUDA Toolkit: Core GPU programming model
- NV HPC SDK*: Optimized compilers for OpenACC*, OpenMP*, and HPC applications

#### Runtime Frameworks for AI and HPC

The platform supports a broad ecosystem of compute and data frameworks:

*HPC Computing*

- MPI (OpenMPI, MPICH, Intel® MPI) for distributed simulations
- TAO*, DOCA* for accelerator-optimized pipelines

*AI/ML Frameworks*

- PyTorch*, TensorFlow*, JAX* for training and fine-tuning
- ONNX* Runtime for portable inference optimization
- Triton Inference Server* to serve LLMs, RAG engines, and multimodal models
- OpenVINO™ to optimize inference on CPUs and XPUs

*Data Analytics Pipelines*

- RAPIDS* for GPU-accelerated analytics
- Apache Spark* for large-scale distributed data processing
- PostgreSQL* for metadata and structured data operations

*RAG / Generative AI / Agentic AI*

- Emerging support for vector databases, embedding frameworks, and agentic toolchains through Intel AI for Enterprise RAG integrations.

### *Application Services*

This layer provides:

- User portals
- Workspace environments (JupyterHub*, Open OnDemand*, custom UI)
- Fine-tuned workload templates for HPC, AI, DA pipelines
- Integration with third-party tools or enterprise ecosystems

Application services enable seamless interaction with the cluster for developers and end users.

### *System Management*

Cluster operators require:

- Deployment automation
- Monitoring and telemetry (Prometheus*, Grafana*, Loki*)
- Network and security policy control
- Service-level configuration management

This ensures system reliability, consistent performance, and operational visibility across the entire infrastructure.

## Hybrid Platform Architecture for AI/HPC Services

The QCT AI POD introduces a hybrid architecture that supports two historically distinct computing ecosystems—Kubernetes-based AI services and Slurm-managed bare-metal HPC workloads—within a single, operationally coherent platform. This design enables the system to support the full spectrum of modern computational demands, ranging from cloud-native AI services and Generative AI pipelines to tightly coupled, performance-critical HPC simulations.

By integrating these execution models on shared infrastructure, shared data services, and unified identity and access management, the QCT AI POD eliminates traditional silos between AI and HPC environments while preserving the performance and operational characteristics required by each workload class.

The hybrid architecture is realized through three tightly integrated service domains.

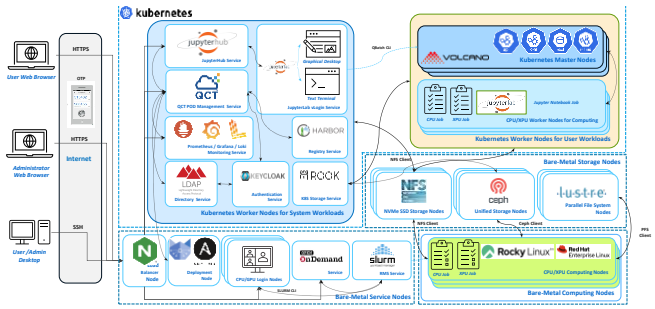



*Figure 6. The hybrid architecture of QCT AI POD*

### *Kubernetes-Based AI Service Plane*

The Kubernetes-based AI service plane provides a cloud-native environment optimized for microservice-oriented, interactive, and elastic AI workloads. It serves as the foundation for modern AI application development, deployment, and operation.

Key capabilities of this service plane include:

- **Container-Oriented Execution:**

  AI services such as LLM inference, embedding generation, retrieval, reranking, multimodal processing, and agentic workflows are deployed as containerized services, enabling rapid scaling and lifecycle management.

- **Advanced Scheduling with Volcano:**

  Volcano extends Kubernetes scheduling capabilities to support batch-oriented and GPU-accelerated AI workloads, enabling fair sharing, gang scheduling, and resource-aware placement for training and inference jobs.

- **User-Facing Workspaces and Portals:**

  Interactive environments such as JupyterHub, graphical desktops, and web-based terminals provide developers and data scientists with intuitive access to compute resources without requiring direct cluster-level interaction.

- **Integrated Platform Services:**

  Core services—including container registries (Harbor*), monitoring and observability (Prometheus, Grafana, Loki), authentication (Keycloak, LDAP), and deployment automation—are natively integrated into the Kubernetes control plane.

This service plane enables rapid experimentation, continuous deployment, and scalable operation of AI and Generative AI services, while maintaining enterprise-grade security and governance.

### *Bare-Metal HPC and AI Service Plane*

Complementing the cloud-native service plane, the bare-metal HPC and AI service plane is optimized for performance-critical, tightly coupled workloads that demand deterministic execution, low latency, and maximum hardware efficiency.

This domain is characterized by:

- **Slurm-Based Resource Management:**

  Slurm orchestrates HPC simulations, large-scale AI training, and batch-oriented workloads running directly on bare metal, ensuring predictable scheduling, efficient resource utilization, and strong workload isolation.

- **Native Access to Hardware Capabilities:**

  Bare-metal execution enables direct access to CPUs, GPUs, and AI accelerators, minimizing software overhead and maximizing performance for MPI-based simulations, distributed deep learning, and data-intensive workloads.

- **Support for Hybrid AI Training Pipelines:**

  Large-scale model training, fine-tuning, and data preprocessing tasks can execute in this plane while seamlessly sharing datasets, checkpoints, and artifacts with Kubernetes-based inference and service workflows.

- **Enterprise Linux Foundations:**

  The service plane is built on production-grade operating systems such as Rocky Linux and Red Hat Enterprise Linux, ensuring stability, long-term support, and compatibility with established HPC ecosystems.

This plane preserves the performance characteristics required by traditional HPC and high-end AI training workloads while remaining fully integrated into the broader platform.

### *Unified Storage and Data Plane*

Bridging both execution domains is a unified, scalable storage and data plane, designed to provide consistent, high-performance data access across all workload types.

Key characteristics include:

- **Multi-Tier Storage Architecture:**

  The platform integrates high-performance NVMe-based storage, distributed parallel file systems (for example, Lustre, BeeGFS), and scalable software-defined storage systems (for example, Ceph, VAST*, WEKA*) to match diverse I/O patterns.

- **Shared Data Accessibility:**

  Datasets, model checkpoints, embeddings, and generative outputs are accessible from both Kubernetes-based services and bare-metal HPC jobs without data duplication or manual synchronization.

- **Performance-Aware Data Placement:**

  Workloads can transparently leverage the most appropriate storage tier—NVMe for low-latency inference and scratch, parallel file systems for training and checkpointing, and object storage for long-term data retention.

- **Consistent Security and Governance:**

  Unified authentication and authorization ensure that data access policies are enforced consistently across execution environments.

This unified data plane enables end-to-end AI and HPC workflows—spanning data ingestion, training, inference, and agentic reasoning—to operate as cohesive pipelines rather than fragmented stages.

### Summary – Why Hybrid Matters

Together, the Kubernetes-based AI service plane, the bare-metal HPC/AI service plane, and the unified storage and data plane form a cohesive hybrid execution platform. This architecture enables the QCT AI POD to simultaneously support:

- Cloud-native AI services and microservice-based applications
- Large-scale AI training and fine-tuning
- Traditional HPC simulations
- Hybrid and agentic workflows that combine multiple execution models

By unifying these domains under a single architectural framework, the QCT AI POD delivers a future-ready foundation for converged HPC, AI, and Generative AI infrastructures.

## Intel AI for Enterprise RAG - a Gen AI Turnkey Solution

Intel AI for Enterprise RAG is an open, modular framework designed to speed up the creation and deployment of enterprise grade Generative AI (GenAI) solutions. It connects innovation across the AI ecosystem with the performance, security, and scalability needs of modern enterprises by offering unified architecture and ready-to-use components.

The framework transforms enterprise data into actionable intelligence while optimizing Total Cost of Ownership (TCO). By leveraging Intel Xeon processors, it streamlines computing and operational efficiency.

Intel AI for Enterprise RAG reduces the complexity of moving from AI experimentation to full production, enabling organizations to integrate reliable, high-performance GenAI workflows directly into their business systems.

At its foundation, Intel AI for Enterprise RAG provides a complete RAG pipeline, delivering intelligent, context-aware responses by combining retrieval, processing, and generation capabilities. Beyond RAG, the platform also supports additional AI pipelines, including text translation and document summarization.

When integrated with the QCT AI POD and validated through the QCT DevCloud environment, Intel AI for Enterprise RAG moves beyond reference implementations to become a production-ready GenAI platform, capable of supporting hybrid workloads across cloud, datacenter, and edge deployments.

### Key Components of Intel AI for Enterprise RAG

#### *Infrastructure*

Enterprise RAG delivers a robust, production ready foundation designed to meet the operational, security, and governance standards of modern IT organizations. Deployable on Kubernetes, the industry's leading enterprise orchestration platform, it enables fully automated solution provisioning and cloud native deployment through integrated ansible playbooks and Helm charts.

This approach ensures consistent rollouts across environments, whether organizations deploy onto an existing cluster or provision a new one with complete infrastructure setup.

These infrastructure choices collectively enable seamless deployment of diverse, production ready AI pipelines - such as Chat Q&A, document summarization, and translation.

#### *Microservices*

Enterprise RAG is built on a modular, cloud native architecture with key elements derived from the Intel AI for Enterprise RAG - an open-source, backend by the Linux Foundation*, initiative designed to standardize and accelerate the development of enterprise grade Generative AI solutions.

Each microservice performs a discrete function in the pipeline and is containerized with well-defined APIs, enabling flexible composition into larger application patterns. This allows for microservice reusability in different parts of a deployment or different pipelines altogether. Typical microservices include:

- Embedding generation
- Retrieval and search services
- Vector Search and data extraction
- Reranking, and prompt templating
- Guardrails for input and output protection
- LLM and Model Server microservices

This design promotes modularity and ease of extension. Supporting such architecture introduced support for Horizontal Pod Autoscaler, enabling automatic scaling of microservice capacity based on demand.

#### *Performance*

Enterprise RAG is purpose built to deliver fast, consistent, and efficient performance. At its core, the platform takes advantage of Intel® Advanced Matrix Extensions (AMX), built into modern Intel Xeon processors. AMX speeds up the matrix multiplication operations that dominate LLM inference, particularly BF16 and INT8 workloads, enabling significantly faster generation without the need for dedicated accelerators.

To ensure predictable performance under real world load, Enterprise RAG uses the Kubernetes Node Resource Interface (NRI) and its balloons policy to isolate CPU resources. By placing workloads into dedicated CPU pools that expand or contract based on demand, the system maintains NUMA affinity and reduces noisy neighbor interference.

Scalability is handled automatically through Horizontal Pod Autoscaling (HPA), which monitors resource utilization and adjusts microservice replicas dynamically

Supporting all of this is a built-in observability stack, combining Prometheus metrics, Grafana dashboards, Tempo tracing, and Loki log aggregation. These tools provide full visibility across the pipeline, including ingestion, retrieval, and generation flows.

#### *Value*

As a result, Intel AI for Enterprise RAG is purpose-built to run optimally on Intel Xeon processors, leveraging their built-in accelerators. It is designed as a fully integrated, end-to-end solution, offering a complete set of capabilities out of the box - essentially a comprehensive, "batteries included" platform that minimizes integration effort and accelerates time to value.

## Error! Reference source not found.Strategic Benefits of Intel AI for Enterprise RAG on QCT DevCloud

Integrating Intel AI for Enterprise RAG with QCT DevCloud yields strategic advantages grounded in both technical capability and operational maturity:

### *Hardware-Accelerated AI Workloads*

- Seamless support for heterogeneous compute architectures, including Intel Gaudi 3, Intel Xeon CPUs, and NVIDA GPUs
- Ability to benchmark and validate AI service performance across multiple hardware vectors
- Data-driven insights into optimal compute placement based on throughput, cost, and efficiency

### *End-to-End Pipeline Enablement*

- Support for comprehensive AI workflows encompassing data ingestion, training, inference, evaluation, and deployment
- Validation of distributed GenAI pipelines in realistic environments
- Rich operational telemetry for performance analysis, bottleneck identification, and continuous improvement

### *Cross-Platform Portability and Openness*

- Native compatibility with standard Kubernetes orchestration and hybrid deployment models
- Architecture that is agnostic to underlying infrastructure choices
- Avoidance of vendor lock-in through open standards and open-source components, facilitating flexibility in multi-cloud and hybrid scenarios

### *Developer Productivity and Adoption*

- Reference examples and reusable components significantly reduce onboarding time
- Modular design encourages consistent, maintainable system construction
- Rapid prototyping and iterative validation accelerate time-to-value for enterprise AI initiatives

Intel AI for Enterprise RAG is more than a microservice repository; it is a comprehensive composable enterprise AI platform. When paired with QCT DevCloud, it becomes a validation and acceleration engine that enables:

- Rapid prototyping of AI workflows
- Benchmark-driven architectural and hardware decisions
- Comparative evaluation for performance and Total Cost of ownership (TCO)
- Real-world evaluation of pipeline behavior under load

Intel AI for Enterprise RAG and QCT DevCloud provide a robust foundation for enterprise adoption of Generative and Agentic AI—delivering a solution that is open, scalable, secure, and ready for production workloads across datacenter and cloud environments.

## QCT DevCloud: Real-World Deployment

QCT DevCloud is a platform tailored for high-performance computing, data analysis, and machine learning workloads and built on the QCT POD reference architecture and integrates QCT's next-generation computing platform, pre-compiled HPC/AI applications, and a comprehensive software development tool environment. This open platform enables users to access it remotely for application system development, integration testing, and optimization.

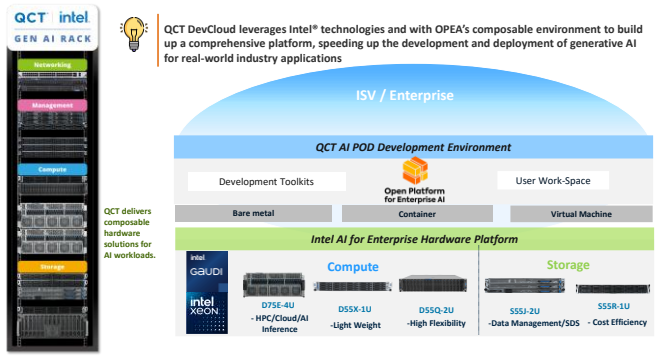



*Figure 7. QCT – Intel GenAI rack reference stack.*

Architecturally, QCT DevCloud combines containerized services, centralized authentication, and hybrid storage backends within a Kubernetes-based compute cluster. Users access the platform via a secure login gateway with OTP, then interact through JupyterHub, Open OnDemand, or SSH-based terminals. Notebook jobs, CPU/XPU workloads, and metadata-aware storage tasks are dispatched to Kubernetes worker nodes or bare-metal nodes running Rocky Linux*Figure 8*. Behind the scenes, the system integrates distributed file systems (for example, Ceph, Lustre), job orchestration frameworks (for example, Volcano, Slurm), and data management services (for example, iRODS*, ROOK*) for compute and storage scalability. Each component is modular and containerized to support flexible, reproducible AI workflows across multiple user roles. By reducing the barrier to entry for edge AI application and integrating with hierarchical data governance systems, QCT DevCloud plays a pivotal role in accelerating the proliferation of AI across all levels of the computing stack.

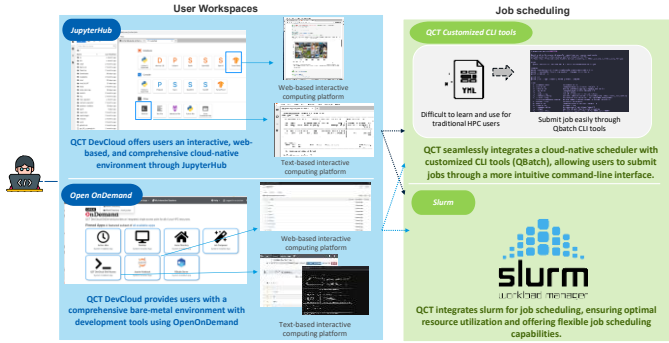



*Figure 8. QCT DevCloud integrates JupyterHub/Open OnDemand with Slurm scheduling via customized CLI tools.*

## Hybrid Hardware Composition

QCT DevCloud includes*Figure 9*:

- QCT servers with multiple generations of Intel Xeon CPUs, including Xeon 6
- NVIDIA GPU servers
- Intel Gaudi 3 AI accelerator servers
- Network Ethernet / InfiniBand Fabrics
- Software Defined Storage: Ceph and Lustre storage

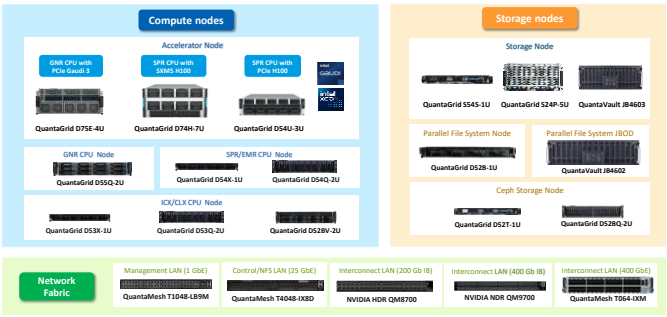



*Figure 9. The hybrid hardware composition of QCT DevCloud.*

## Integrated Software Stack for Converged HPC and AI Workloads

The *Figure 10* illustrates the end-to-end software ecosystem underpinning the QCT DevCloud environment, highlighting how pre-integrated applications, development toolchains, and management frameworks are unified to support heterogeneous HPC, AI, and data analytics workloads.

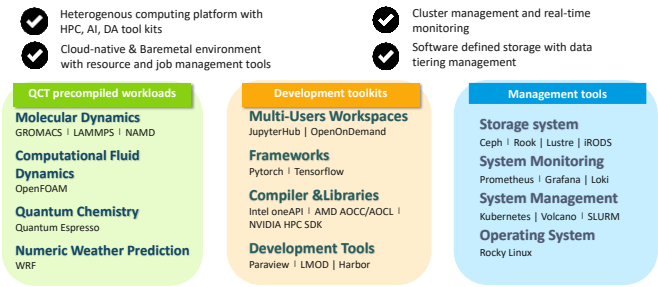



*Figure 10. QCT DevCloud environment overview*

On the application layer, QCT DevCloud provides a set of precompiled and validated workloads targeting key scientific and engineering domains, including Molecular Dynamics (GROMACS*, LAMMPS*, NAMD*), Computational Fluid Dynamics (OpenFOAM*), Quantum Chemistry (Quantum ESPRESSO*), and Numerical Weather Prediction (WRF). These workloads are optimized for modern CPU and accelerator architectures, enabling users to rapidly deploy production-ready simulations without extensive software tuning.

The development toolkit layer delivers a comprehensive computational environment for both interactive and batch-oriented workflows. Web-based platforms such as JupyterHub and Open OnDemand simplify access to shared HPC/AI resources, while mainstream AI frameworks (PyTorch, TensorFlow) and high-performance compiler stacks (Intel® oneAPI, AMD* AOCC/AOCL, NVIDIA HPC SDK*) ensure portability and performance across heterogeneous hardware. Supporting tools such as ParaView* and LMOD* further enhance visualization, environment management, and user productivity.

The management and operations layer integrates system-level services required for scalable and reliable cluster operation. This includes software-defined storage systems (Ceph, Lustre, iRODS), real-time monitoring and observability (Prometheus, Grafana, and unified resource orchestration across Kubernetes, SLURM, and Volcano. A stable enterprise Linux* foundation (Rocky Linux) provides consistency, security, and long-term maintainability.

Together, these layers form a cohesive, cloud-native yet bare-metal-capable software stack, enabling DevCloud to deliver a flexible, scalable, and production-ready platform for converged HPC and AI workloads—from traditional scientific simulations to modern generative and agentic AI applications.

## Intel Gaudi 3 AI Accelerator Benchmarking on QCT DevCloud

QCT DevCloud provides an engineering validation environment for evaluating AI infrastructure technologies under realistic deployment conditions. As part of the enablement of Intel Gaudi 3 PCIe accelerators (HL-338) within the QCT AI POD solution framework, an extensive benchmarking campaign was conducted to characterize both inference performance and distributed communication behavior on a scalable, Ethernet-based architecture.

The evaluation focuses on two complementary dimensions:

- Single-accelerator inference performance, measuring the efficiency of Intel Gaudi 3 PCIe accelerators on representative large language model (LLM) inference workloads.
- Multi-accelerator scalability, assessing collective communication performance across intra-node (two-quad) and inter-node (two-node) configurations using standard PCIe 5.0 and RoCE v2 Ethernet interconnects.

This section presents the benchmarking methodology, system architecture, performance results, and technical interpretation required to understand the performance characteristics, scalability limits, and deployment suitability of Intel Gaudi 3 PCIe–based AI systems in enterprise and cloud environments.

### Benchmarking Environment Overview

All benchmarks were executed in the QCT DevCloud environment, using two identical QuantaGrid D75E-4U Table 1 servers interconnected via a QCT QuantaMesh TA064-IXM 400/800 GbE RoCE v2 Ethernet switch. This configuration enables comprehensive evaluation of both intra-node communication and inter-node scale-out communication over Ethernet.

The environment was deliberately designed to reflect real-world PCIe-based AI deployments, stressing key system components including PCIe 5.0 bandwidth, RDMA networking, and collective communication efficiency. As a result, the measured performance accurately represents the behavior and scalability limits of Intel Gaudi 3 PCIe accelerators when deployed in AI POD architectures.

| Hardware | |
|---|---|
| Server Model | QuantaGrid D75E-4U  |
| CPU | Dual Intel Xeon 6767P (2.4GHz / 64 Cores) |
| Memory | 1TB (64 GB DDR5 6400 DIMM x 16) |
| OS Disk | Samsung PM9A3 1.92TB E1.S x 1 |
| Data Disk | Samsung PM9A3 1.92TB E1.S x 6 (Software RAID 0) |
| PCIe Fabric | Dual PCIe Gen5 switches |
| AI Accelerators | 4× Intel Gaudi 3 PCIe (HL-338) accelerators / quad (two quads per node) |
| N/S Network adaptors | 1× NVIDIA dual-port 25Gb Ethernet NIC (CX5, 25GbE dual port) |
| E/W Network adaptors | 4× NVIDIA dual-port 200Gb Ethernet HCA (CX7, 200GbE dual port) |
| **Software** | |
| OS | Rocky Linux Release 9.5 (Blue Onyx*) |
| Kernel | 5.14.0-503.14.1.el9_5.x86_64 |
| Gaudi Driver | 1.22.1-97ec1a4 |
| Resource Manager | SLURM 25.05.5 |

Table 1. Hardware configuration of the Intel Gaudi 3 compute node (QuantaGrid D75E-4U) used in this work.

## Single Intel Gaudi 3 PCIe AI Accelerator Inference Performance

To evaluate real-world inference performance for modern generative AI workloads, QCT DevCloud conducted controlled benchmarks using Llama* 3.x Instruct models Table 2on a single Intel Gaudi 3 PCIe accelerator, with results compared against a dual-socket Intel Xeon 6 CPU platform.

The results*Figure 11* demonstrate that Intel Gaudi 3 PCIe accelerators deliver substantial inference throughput advantages for transformer-based models. By efficiently accelerating the compute-intensive components of LLM inference—such as attention mechanisms and feed-forward layers—Intel Gaudi 3 achieves significantly higher token generation rates than CPU-only execution.

Because inference workloads typically keep model weights resident on the accelerator and require minimal host–device data movement, PCIe connectivity does not present a measurable performance bottleneck under the tested conditions. These characteristics make Intel Gaudi 3 PCIe accelerators well-suited for high-throughput, cost-efficient inference deployments, including enterprise use cases such as Retrieval-Augmented Generation (RAG), agent-based AI workflows, and multimodal inference pipelines.

| AI Model Settings | |
|---|---|
| Docker Version | Docker version 28.4.0, build d8eb465 |
| Docker Image | vault.habana.ai/gaudi-docker/1.22.0/ubuntu22.04/habanalabs/pytorch-installer-2.7.1 |
| vLLM for CPU | https://github.com/vllm-project/vllm/tree/v0.9.0.1 |
| vLLM for GPU | https://github.com/HabanaAI/vllm-fork/tree/v0.9.0.1%2BGaudi-1.22.0 |
| AI Inference Model | Llama 3.1 8B Instruct \| Llama 3.2 3B Instruct |
| AI Data Type | BF16 |
| Input Length | 1024 |
| Output Length | 1024 |
| Maximum Concurrency | 24 (64 CPU Cores) \| 24 (1 Gaudi 3 Accelerator) |

Table 2. AI inference software stack and vLLM configuration for Llama 3.x models

## *Performance Results*

| Model | CPU Tokens/s | Gaudi 3 Tokens/s | Speedup |
|---|---|---|---|
| Llama 3.2 3B | 238.69 | 4,856.8 | 20.3× |
| Llama 3.1 8B | 175.66 | 4,471.95 | 25.4× |

*Table 3. The vLLM inference throughput (tokens/s) and Gaudi 3 AI accelerator speedup over a 64-core Xeon 6 CPU for Llama 3.2 3B and Llama 3.1 8B.*

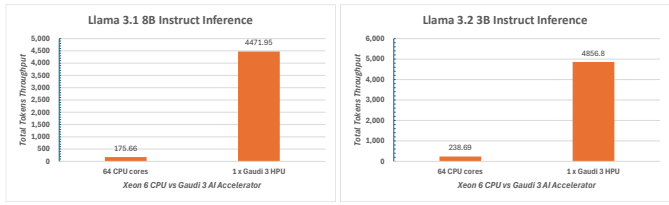



*Figure 11. vLLM inference throughput comparison (tokens/s) between a 64-core Xeon 6 CPUs and a single Gaudi 3 AI Accelerator for Llama 3.1 8B and Llama 3.2 3B.*

Intel Gaudi 3 utilizes specialized Transformer Engines with FP8/BF16 GEMM acceleration, dramatically reducing compute latency for:

- QKV attention projections
- Multi-head attention
- Feed-forward layers
- LayerNorm and rotary embeddings

CPU vector units (AVX-512/AMX) cannot match the parallelism and memory bandwidth of Intel Gaudi's matrix-multiplication engines.

Since LLM inference with small parameters can:

- Keeps weights fully resident on the device
- Performs minimal host-device communication
- Is dominated by GEMM-heavy compute steps

PCIe connectivity introduces virtually no runtime overhead. This confirms that Intel Gaudi 3 PCIe remains an optimal choice for high-throughput inference loads.

The results validate several architectural design decisions:

- Single HPU = Replaces dozens of CPU cores for high-throughput inference
- Scales seamlessly with multiple Intel Gaudi 3 devices for enterprise RAG, agentic AI, and multimodal pipelines
- Highly cost-efficient, enabling lower TCO per token compared to CPU inference

Intel Gaudi 3 consistently delivers >20× acceleration over CPU-based inference, reaffirming its role as a cost-effective inference accelerator for modern GenAI workloads.

## Multi-Accelerator Performance

Each QCT D74E-4U server node includes two PCIe switch domains, each hosting 4× Intel Gaudi 3 accelerators. Performance was evaluated by evaluating the collective communication performance of Intel Gaudi 3 PCIe accelerators using HCCL in a distributed configuration. Benchmarks were executed on two identical QuantaGrid D75E-4U servers, each equipped with dual PCIe Gen5 switches and up to eight Gaudi 3 accelerators, interconnected via a QCT QuantaMesh TA064-IXM switch with 400/800 GbE

RoCE v2 Ethernet fabric.

The objective is to characterize:

- Effective application-level bandwidth
- Underlying fabric (bus/network) load
- Scaling behavior across 1 quad (4 devices), 2 quads (8 devices), and 2 nodes (16 devices)

The system architecture for the test environment is shown below *Figure 12*.

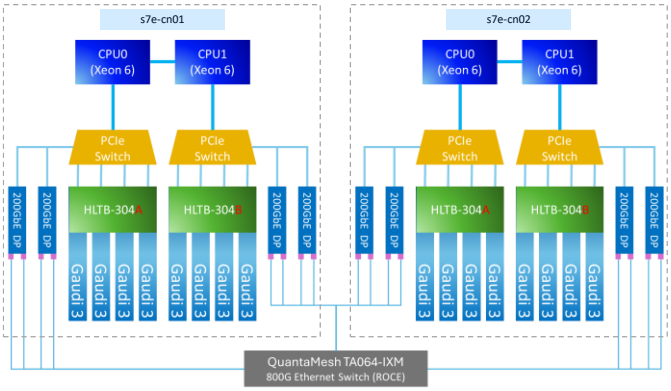



*Figure 12. Two-node Intel Gaudi 3 cluster topology showing dual Intel Xeon CPUs and PCIe switches connecting eight Intel Gaudi 3 accelerators per node, interconnected via 200 GbE RoCE links through an 800G Ethernet switch.*

## *Benchmark Methodology*

The Habana Collective Communications Library (HCCL) benchmark evaluates the efficiency and scalability of collective communication primitives that are fundamental to distributed AI training. These primitives—including **all-reduce**, **reduce-scatter**, **all-gather**, and **all-to-all**—are used extensively for gradient aggregation, parameter synchronization, activation exchange, and Mixture-of-Experts (MoE) traffic in modern deep learning workloads.

Running HCCL benchmarks on multiple Intel Gaudi 3 PCIe AI accelerators provides direct insight into how effectively the system supports data-parallel and hybrid-parallel training at scale. As training workloads extend beyond a single accelerator, overall performance becomes increasingly sensitive to communication efficiency rather than raw compute throughput. HCCL benchmarking therefore isolates and characterizes the behavior of the communication stack under realistic distributed execution patterns.

Specifically, multi-accelerator HCCL benchmarks are used to:

- Quantify application-level effective bandwidth and fabric-level bandwidth consumption
- Validate scaling behavior as the number of accelerators increases
- Identify the impact of different collective communication patterns on performance
- Assess the readiness of the platform for large-scale AI training and multi-node deployments

By analyzing HCCL performance across multiple Gaudi 3 accelerators, the benchmarks provide a critical measure of whether the converged infrastructure can sustain efficient distributed training and serve as a reliable foundation for scalable AI workloads within the QCT AI POD architecture.

## *Summary of configuration for HCCL benchmarks*

| HCCL Benchmark Configuration | | | |
|---|---|---|---|
| Collective Operations | Type | Description | Insights |
| | All Reduce | Aggregates values from all ranks using a reduction operation (for example, sum) and distributes the identical reduced result back to all ranks | Core collective for data-parallel training; scales predictably and efficiently on Gaudi 3 over RoCE |
| | Reduce-Scatter | Performs reduction across ranks and scatters distinct reduced partitions to each rank | Most communication-efficient primitive; ideal for gradient sharding and large-scale training |
| | All-Gather | Gathers distinct data blocks from all ranks and distributes the full aggregated dataset to every rank | Synchronization-heavy and latency-dominated; effective BW decreases sharply at scale |
| | All-to-All | Each rank sends a unique portion of its data to every other rank and receives distinct data from them | Most network-stressful collective; sensitive to congestion, especially on Ethernet fabrics |
| Message size | 64MB | | |
| Iterations | 1000 | | |
| Execution model | One MPI rank per Gaudi device<br>SLURM-managed placement | | |
| Software stack | HCCL via Gaudi driver 1.22.1<br>OpenMPI* 4.1.6<br>Libfabric* 1.20.0 | | |
| Networking | NVIDIA CX7 200 GbE HCAs<br>QuantaMesh TA064-IXM RoCE v2 switch | | |

Table 4. HCCL benchmark setup and collective operation definitions for Intel Gaudi 3 over 200 GbE RoCE.

The benchmark reports two bandwidth metrics for each collective:

### Algo Bandwidth (Algo BW)

- Represents algorithmic or effective bandwidth
- Measures the rate at which useful payload data progresses from the application's perspective
- Directly correlates with training or inference step efficiency

### Network / Bus Bandwidth (NW BW)

- Represents fabric-level bandwidth consumption
- Reflects the actual traffic carried by the interconnect
- Includes algorithm-induced data replication inherent to collective communication

### Definitions

- **$N$: number of participating ranks (1 rank per Gaudi)**
  - 1 Quad = $N = 4$
  - 2 Quads = $N = 8$
  - 2 Nodes = $N = 16$
- **All-Reduce (ring):**

$$\text{NW}_{\text{theory}} = \frac{2(N-1)}{N} \times \text{Algo}$$

- **All-Gather (ring):**

$$\text{NW}_{\text{theory}} = (N-1) \times \text{Algo}$$

- **Reduce-Scatter / All-to-All:**

$$\mathrm{NW}_{\mathrm{theory}} = \frac{(N-1)}{N} \times \mathrm{Algo}$$

The test results are shown in the following *Figure 13* and Table 5.

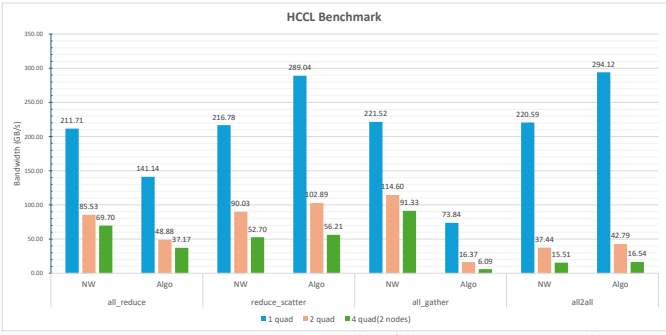



*Figure 13. HCCL benchmark bandwidth (GB/s) for key collectives (all-reduce, reduce-scatter, all-gather, all-to-all), comparing 1-quad, 2-quad, and 4-quad (two-node) configurations.*

| Collective | Scale | N | Algo BW | NW BW (Measured) | NW BW (Theoretical) | Integrated Technical Insight |
|---|---|---|---|---|---|---|
| All-Reduce | 1 Quad | 4 | 141.14 | 211.71 | 211.71 | Ring all-reduce behavior fully matches theory; efficient intra-node reduction |
| | 2 Quads | 8 | 48.88 | 85.53 | 85.54 | Scaling limited by synchronization, not fabric bandwidth |
| | 2 Nodes | 16 | 37.17 | 69.70 | 69.69 | Smooth cross-node scaling confirms healthy RoCE fabric |
| Reduce-Scatter | 1 Quad | 4 | 289.04 | 216.78 | 216.78 | Most bandwidth-efficient collective; ideal for gradient sharding |
| | 2 Quads | 8 | 102.89 | 90.03 | 90.03 | Algo BW remains high; fabric utilization closely tracks payload |
| | 2 Nodes | 16 | 56.21 | 52.70 | 52.70 | Near-unity Algo↔NW ratio indicates minimal communication amplification |
| All-Gather | 1 Quad | 4 | 73.84 | 221.52 | 221.52 | Latency-dominated collective with high traffic amplification |
| | 2 Quads | 8 | 16.37 | 114.60 | 114.59 | Effective bandwidth collapses as rank count increases |
| | 2 Nodes | 16 | 6.09 | 91.33 | 91.35 | Coordination cost dominates despite stable fabric throughput |
| All-to-All | 1 Quad | 4 | 294.12 | 220.59 | 220.59 | High payload efficiency at small scale |
| | 2 Quads | 8 | 42.79 | 37.44 | 37.44 | Rapid efficiency loss due to traffic fan-out |
| | 2 Nodes | 16 | 16.54 | 15.51 | 15.51 | Most network-stressful collective on Ethernet fabrics |

Table 5. HCCL collective performance summary comparing algorithm and network bandwidth across 1-quad, 2-quad, and 2-node scaling.

### *Technical Interpretation*

The multi-accelerator HCCL benchmarks demonstrate that Intel Gaudi 3 PCIe AI accelerators exhibit stable, predictable, and theoretically consistent scaling behavior across intra-quad, inter-quad, and multi-node configurations when supported by sufficient external network bandwidth.

At the single-quad (4-device) level, collective operations—particularly all-reduce and reduce-scatter—closely match theoretical network-to-algorithm bandwidth relationships. This confirms that Intel Gaudi 3's local communication path delivers efficient data movement with minimal overhead, making a single quad an effective scale-up unit for data-parallel training workloads.

As scaling extends to two quads (8 devices) and two nodes (16 devices), performance trends remain well-behaved and consistent with collective communication theory. The measured Network Bandwidth (NW BW) aligns precisely with theoretical expectations for all tested collectives, indicating that performance degradation is driven primarily by algorithmic synchronization and communication amplification, rather than hardware or fabric inefficiencies. This behavior confirms the robustness of Intel Gaudi 3's communication stack under distributed execution.

Among the evaluated collectives:

- **Reduce-scatter** consistently delivers the highest efficiency, with near-unity algorithm-to-network bandwidth ratios even at 16 devices, making it particularly well suited for gradient sharding and AI model training.
- **All-reduce** scales smoothly across quads and nodes, validating Intel Gaudi 3's suitability for conventional data-parallel training patterns.
- **All-gather and all-to-all** expose expected scalability limits due to inherent traffic fan-out and synchronization costs, especially over Ethernet fabrics, but do so without unexpected instability or congestion.

A critical observation from these results is that Intel Gaudi 3 PCIe accelerators do not include native network interfaces. Consequently, external high-speed Ethernet connectivity (200/400 GbE RoCE v2) is mandatory to sustain efficient communication beyond a single quad or node. The benchmark data confirms that when adequately provisioned with external NICs and a high-performance switch fabric, Intel Gaudi 3 scales reliably across servers without introducing abnormal communication penalties.

Overall, the multi-accelerator benchmarking results validate that Intel Gaudi 3 PCIe delivers predictable, communication-efficient scaling characteristics, with performance bounded by well-understood collective algorithms rather than architectural limitations. This makes Intel Gaudi 3 PCIe a viable and scalable accelerator choice for distributed AI training, provided that external network bandwidth is designed to match the communication demands of the target workload

## Conclusion

These benchmarking results obtained on QCT DevCloud clearly demonstrate that the Intel Gaudi 3 PCIe AI accelerator delivers strong, scalable, and production-ready performance for modern AI workloads, spanning both single-accelerator inference and multi-accelerator distributed training scenarios.

### Exceptional Single-Accelerator Efficiency

Single-accelerator inference benchmarks using Llama 3.1 8B and Llama 3.2 3B Instruct models show that a single Intel Gaudi 3 PCIe accelerator achieves more than 4,400–4,800 tokens/s, outperforming a fully utilized 64-core Intel Xeon 6

CPU by over 20× under identical conditions.

This confirms that Intel Gaudi 3 PCIe is not merely competitive but highly compelling as a cost-effective inference accelerator, especially for enterprise-scale GenAI deployments.

### Predictable and Efficient Intra-Quad Scaling

When scaling from one to four Intel Gaudi 3 accelerators (one quad), collective communication benchmarks using HCCL demonstrate:

- Close alignment between measured and theoretical bandwidth
- High algorithmic efficiency for all-reduce and reduce-scatter
- Minimal communication amplification

These results indicate that Intel Gaudi 3's intra-quad communication architecture is well optimized, enabling efficient data-parallel execution without requiring external networking. For many AI workloads, this provides an excellent scale-up path within a single server domain, simplifying deployment while maintaining high utilization.

### Robust Scale-Out Characteristics for Distributed Training

As workloads expand beyond a single quad—across multiple quads or nodes—the benchmarks confirm that Intel Gaudi 3 continues to scale in a stable and predictable manner, provided sufficient external network bandwidth is available.

Importantly, the measured network bandwidth closely matches theoretical models across all tested scales, demonstrating that Intel Gaudi 3's communication stack behaves deterministically and transparently, a critical requirement for large-scale training.

### Software and Ecosystem Maturity

The test results also validate the maturity of the Intel Gaudi software ecosystem:

- HCCL delivers consistent, standards-compliant collective behavior
- Integration with OpenMPI, libfabric, and SLURM enables seamless adoption in HPC and AI clusters
- vLLM and PyTorch support enable modern inference and training workflows without architectural compromise

This positions Intel Gaudi 3 PCIe as a practical accelerator choice, not only for experimentation but for sustained production use.

### Strategic Value of Intel Gaudi 3 PCIe

Taken together, the findings establish Intel Gaudi 3 PCIe AI accelerators as:

- Highly efficient for single-accelerator inference
- Well-balanced for intra-node scale-up
- Predictable scaling within properly provisioned Ethernet fabrics for distributed AI training
- Supported by a mature, open, and enterprise-ready software stack

The absence of integrated scale-out networking on the PCIe variant does not inherently limit Intel Gaudi 3's applicability. Instead, it enables greater flexibility in system design, allowing architects to tailor networking bandwidth, fabric technology, and topology to specific workload requirements. While this approach may introduce additional considerations around interconnect latency and PCIe resource allocation compared to tightly integrated accelerator fabrics, it provides deployment versatility and cost optimization advantages across a wide range of AI and hybrid HPC environments.

The benchmark results conclusively demonstrate that Intel Gaudi 3 PCIe AI accelerators deliver strong performance, excellent scaling characteristics, and operational predictability across a wide range of AI workloads.

Intel Gaudi 3 PCIe stands out as a high-performance, cost-efficient, and scalable alternative for enterprises and research institutions seeking an open, non-proprietary accelerator platform for next-generation Generative AI and large-scale AI training

## HPC Workload Benchmarking on QCT DevCloud

### Overview and Benchmarking Environment

QCT DevCloud provides a production-grade validation environment for evaluating HPC infrastructure technologies under realistic deployment conditions. Following the validation of Intel Gaudi 3 AI accelerators for GenAI workloads, an extensive HPC workload benchmarking campaign was conducted to characterize CPU-based computational performance across multiple generations of Intel Xeon processors within the QCT AI POD solution framework.

This evaluation demonstrates the platform's ability to support traditional HPC simulations alongside AI workloads, validating the converged architecture's capacity to handle heterogeneous computational demands. The benchmarking focuses on two widely adopted foundational HPC application domains:

- Electronic structure calculations and ab initio molecular dynamics using VASP (Vienna Ab initio Simulation Package) [4], one of the most widely adopted codes in computational materials science and quantum chemistry
- Density functional theory calculations using Quantum ESPRESSO, [5] a leading open-source suite for electronic-structure calculations and materials modeling at the nanoscale

These workloads are representative of compute-intensive scientific applications that require sustained floating-point performance, efficient memory bandwidth utilization, and optimized parallel execution across multiple CPU cores—characteristics that are fundamental to modern HPC cluster deployments.

### Hardware Configuration

All HPC benchmarks were executed on QCT QuantaGrid D-series compute nodes deployed within the QCT DevCloud environment. The evaluation compares two Intel Xeon processor generations to assess performance characteristics and validate architectural scalability:

| Component | Specification |
|---|---|
| Server Platform | QCT QuantaGrid D-series compute nodes |
| CPU Configuration | **Intel Xeon 8592+ (Emerald Rapids):** Dual-socket, 64 cores per socket, 128 cores total<br>**Intel Xeon 6767P (Granite Rapids):** Dual-socket, 64 cores |

| Component | Specification |
|---|---|
| | per socket, 128 cores total |
| OS and Software Stack | Operating System: Rocky Linux 9.5<br>Compiler: Intel oneAPI Fortran Compiler<br>Math Libraries: Intel oneMKL (Math Kernel Library)<br>MPI Implementation: Intel MPI<br>Resource Manager: SLURM |

Both processor configurations provide identical core counts (128 cores total), enabling direct performance comparisons that isolate architectural improvements between processor generations while maintaining consistent parallelism.

## VASP Performance Evaluation

### Application Overview

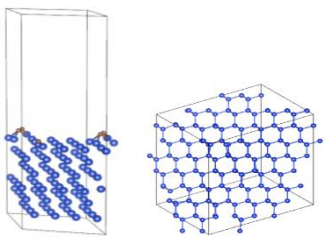
The Vienna Ab initio Simulation Package (VASP) is one of the most widely used codes for electronic-structure calculations and first-principles molecular dynamics in computational materials science, condensed matter physics, and quantum chemistry. VASP implements density functional theory (DFT) using pseudopotentials or the projector-augmented wave method and a plane wave basis set, enabling highly accurate predictions of material properties from first principles.

VASP workloads are characterized by intensive floating-point operations, large-scale matrix diagonalization, Fast Fourier Transforms (FFT), and frequent inter-process communication, making them ideal benchmarks for evaluating HPC system performance, memory bandwidth, and parallel scalability.

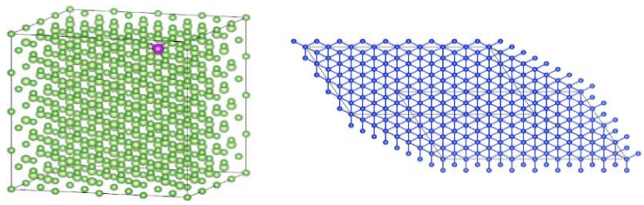
The VASP benchmark test cases were performed in collaboration with **Jyh-Pin Chou, Associate Professor** at the **Graduate School of Advanced Technology, National Taiwan University**. Professor Chou's expertise in computational materials science and density functional theory calculations was instrumental in selecting representative test cases that reflect real-world research workloads and ensuring that the VASP benchmarking methodology adheres to best practices in the computational chemistry community.

### Benchmark Configuration

| Parameter | Configuration |
|---|---|
| VASP Version | 6.5.1 |
| Compilation | Intel oneAPI Fortran, Intel oneMKL, Intel MPI |
| Test Cases | CuC-vdW: Copper-Carbon van der Waals system<br>Si256-VJT-HSE06: 256-atom Silicon HSE06 hybrid functional<br>GaAsBi-512: 512-atom Gallium Arsenide Bismuth alloy<br>Si-Huge: Large-scale Silicon supercell |
| Execution Model | Fully distributed MPI execution across 128 cores |



CuC_VD Si256_VJT_HS



GaAsBi_512 GaAsBi_512

*Figure 14. The structure of VASP test cases*

The selected test cases represent a range of system sizes and computational complexity, from small van der Waals systems to large supercells requiring hybrid functionals—ensuring comprehensive evaluation of CPU performance across diverse DFT workloads.

### Performance Results

| Test Case | Intel 8592+ | Intel 6767P | Speedup |
|---|---|---|---|
| CuC-vdW | 117.13 | 94.62 | **1.24×** |
| Si256-VJT-HSE06 | 877.90 | 751.23 | **1.17×** |
| GaAsBi-512 | 573.29 | 486.07 | **1.18×** |
| Si-Huge | 1272.77 | 1079.05 | **1.18×** |

Table 6. VASP execution time (in seconds, lower is better) and speedup for Intel Xeon 6767P relative to Intel Xeon 8592+

## Quantum ESPRESSO Performance Evaluation

### Application Overview

Quantum ESPRESSO (Open-Source Package for Research in Electronic Structure, Simulation, and Optimization) is an integrated suite of open-source computer codes for electronic-structure calculations and materials modeling at the nanoscale. It is based on density-functional theory, plane waves, and pseudopotentials, and is widely used for research in condensed matter physics, materials science, and nanoscience.

Quantum ESPRESSO workloads involve intensive linear algebra operations, iterative diagonalization of large sparse matrices, and distributed FFT calculations. These characteristics make Quantum ESPRESSO an excellent benchmark for assessing CPU computational throughput, memory subsystem performance, and parallel communication efficiency.

### Benchmark Configuration

| Parameter | Configuration |
|---|---|
| Quantum ESPRESSO Version | 7.0 (Released March 14, 2025) |
| Compilation | Intel oneAPI Fortran, Intel oneMKL, Intel MPI |
| Test Cases | **GRIR443**: 443-atom graphene-related system<br>**GRIR686**: 686-atom graphene-related system |
| Execution Model | Fully distributed MPI execution across 128 cores |

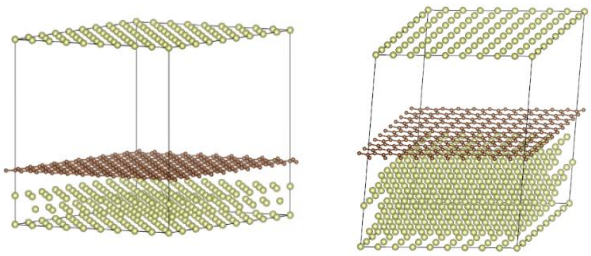



*Figure 15. The structure of test cases GRIR443 and GRIR686*

The GRIR test cases represent large-scale electronic structure calculations typical of modern materials research, particularly in two-dimensional materials such as graphene and related carbon nanostructures.

### Performance Results

Table 7 Quantum ESPRESSO execution time (in seconds, lower is better) and speedup for Intel Xeon 6767P relative to Intel Xeon 8592+.

### Technical Interpretation

The HPC workload benchmarking results demonstrate that the Intel Xeon 6767P (Granite Rapids) processor consistently outperforms the Intel Xeon 8592+ (Emerald Rapids) processor across all tested VASP and Quantum ESPRESSO workloads, despite both configurations utilizing identical core counts (128 cores total across dual sockets).

### Architectural Improvements

The observed performance improvements—ranging from 1.17× to 1.24× across all workloads—can be attributed to several key architectural enhancements in the Granite Rapids microarchitecture:

- Enhanced memory bandwidth and latency: Granite Rapids features improved memory subsystem architecture with support for DDR5 memory at higher frequencies, providing greater sustained bandwidth for memory-intensive DFT calculations.
- Optimized floating-point execution units: Improvements to the floating-point and vector units accelerate the dense linear algebra operations that dominate electronic structure calculations.
- Improved inter-core communication: Enhanced mesh interconnect topology reduces on-package latency for MPI communication patterns typical of distributed DFT codes.
- Advanced prefetching and caching mechanisms: More sophisticated cache hierarchies and prefetchers better anticipate memory access patterns in iterative scientific algorithms.

### Workload Characteristics and Performance Trends

The performance improvements are remarkably consistent across both applications and test cases, indicating that the architectural enhancements benefit a broad range of HPC workload characteristics:

- VASP workloads show speedups ranging from 1.17× to 1.24×, with smaller systems (CuC-vdW) demonstrating slightly higher relative performance gains than larger systems. This suggests that both compute throughput and communication efficiency have improved.
- Quantum ESPRESSO workloads exhibit similar performance trends (1.22× to 1.24× speedup), confirming that the architectural improvements translate effectively across different code bases and algorithmic implementations.
- Large-scale workloads (Si-Huge, GRIR686) maintain strong speedups, demonstrating that the platform scales effectively as problem size increases and validating the memory subsystem enhancements.

### Software Stack Optimization

All benchmarks were compiled using Intel oneAPI toolchains, including the Intel Fortran Compiler, Intel MKL (Math Kernel Library), and Intel MPI. These software components are specifically optimized for Intel Xeon processors and include architecture-specific optimizations that extract maximum performance from the underlying hardware.

| Test Case | Intel 8592+ | Intel 6767P | Speedup |
|---|---|---|---|
| GRIR443 | 632.70 | 511.30 | **1.24×** |
| GRIR686 | 1999.60 | 1644.00 | **1.22×** |

The consistent performance gains across different workloads confirm that both the hardware architecture and the software stack are functioning synergistically, delivering optimized execution paths for compute-intensive scientific applications.

### Validation of Converged HPC-AI Platform

These HPC benchmarking results, when combined with the AI accelerator benchmarking presented earlier, provide comprehensive validation of the QCT AI POD's ability to support converged HPC and AI workloads within a unified infrastructure:

- CPU-based HPC workloads demonstrate strong performance on modern Intel Xeon processors, validating the platform's capability to support traditional scientific computing applications.
- AI accelerator workloads (as demonstrated with Intel Gaudi 3) deliver substantial inference acceleration and predictable scalability.
- Shared infrastructure enables flexible resource allocation and workload coexistence, with both HPC simulations and AI training/inference tasks executing efficiently on the same platform.

The QCT DevCloud deployment successfully demonstrates that heterogeneous computing resources—spanning multiple generations of CPUs and diverse AI accelerators—can be unified within a single operational environment while maintaining the performance characteristics required by both computational domains.

### Conclusion

The HPC workload benchmarking campaign on QCT DevCloud validates the platform's readiness to support production-grade scientific computing applications alongside modern AI workloads. The Intel Xeon 6767P processor delivers consistent and substantial performance improvements across VASP and Quantum ESPRESSO benchmarks, demonstrating that generational architectural enhancements translate directly into measurable application-level speedups.

Combined with the AI accelerator performance validation presented earlier, these results establish that the QCT AI POD Reference Architecture successfully realizes the vision of a unified, scalable, and high-performance platform for converged HPC and AI workloads. Organizations can confidently deploy this architecture to support diverse computational demands—from large-scale quantum chemistry simulations to distributed deep learning training and high-throughput GenAI inference—within a single, operationally coherent infrastructure.

This comprehensive benchmarking demonstrates that the converged platform is not merely conceptual, but production-ready, performant, and capable of supporting the full spectrum of modern computational research and enterprise AI deployment scenarios.

## QChat: End-to-End GenAI & Agentic AI Use Case

The QChat AI Chatbot is designed to demonstrate how modular AI model microservices - each purpose-built for specialized AI tasks and seamlessly integrated – can be composed to enable generative and agentic AI on the QCT AI POD with Intel AI for Enterprise RAG Platform support.

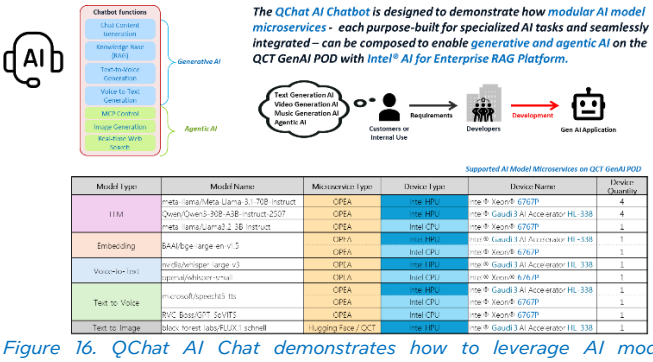



*Figure 16. QChat AI Chat demonstrates how to leverage AI model microservices and target devices (Intel CPU and Intel Gaudi 3) on the QCT AI POD for GenAI and Agentic AI workloads.*

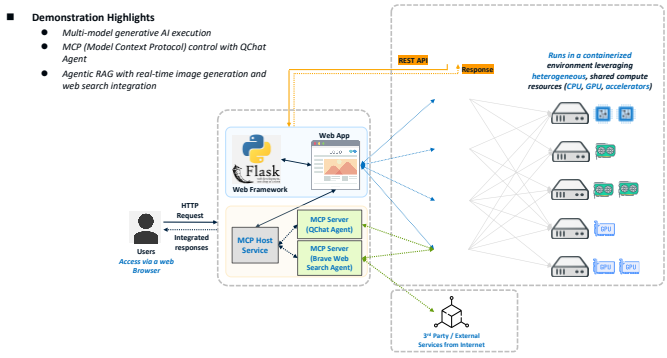



*Figure 17. Agentic chatbot inference workflow and architecture.*

### Scenario 1: CPU versus Intel Gaudi 3 LLM Inference Comparison

#### Overview

This demo demonstrates the performance difference between CPU-based and accelerator-based LLM inference using the same model and the same prompt within QChat.

#### What the Demo Shows

- QChat provides a configuration view where multiple AI model backends can be selected.
- The LLaMA-3.1-8B model is executed on a CPU and then on Intel Gaudi 3, using the same input question.
- The performance difference is clearly visible to the user.

#### Demo Flow

1. *Model Configuration Overview*

   QChat briefly displays the current AI model settings.

2. *Available Model Types*

   Four different AI model execution types are shown as selectable options.

3. *CPU-Based Inference*

   The user asks a question using LLaMA-3.1-8B running on CPU. The response is generated with noticeably higher latency.

4. *Switch to Intel Gaudi 3*

   The backend is switched to LLaMA-3.1-8B running on Intel Gaudi 3.

5. *Performance Comparison*

   The same question is asked again. The response is returned much faster, making the performance advantage immediately apparent.

#### Key Capabilities Highlighted

- Runtime switching between CPU and accelerator backends
- Consistent model behavior across different hardware
- Accelerator-optimized inference on QCT AI POD

#### Demonstration Value

- Shows real, user-perceived performance improvement
- Demonstrates that applications can benefit from accelerators without changing the model or prompt
- Highlights Intel Gaudi 3 as a practical solution for high-throughput GenAI inference

### Scenario 2: Using AI Agents to Change QChat Settings

#### Overview

This demo shows how AI agents can be used to modify system settings via natural language, reducing manual configuration effort.

#### What the Demo Shows

- The user explores the MCP server, which exposes multiple AI agent capabilities.
- After enabling AI agents, system behavior is changed through an LLM-driven interaction.

#### Demo Flow

1. *AI Agent Activation*

   The AI agent feature is enabled.

2. *CPU-Based Text-to-Voice*

   A text-to-voice model running on CPU is used to generate audio output.

3. *Agent-Driven Configuration Change*

   The user asks the LLM (with AI agent enabled) to switch the text-to-voice backend.

4. *Intel Gaudi 3-Based Text-to-Voice*

   The system switches to a text-to-voice model running on Gaudi, demonstrating improved responsiveness.

### Key Capabilities Highlighted

- AI agents with controlled system-level actions
- Natural-language driven configuration changes
- Dynamic backend switching through Intel AI for Enterprise RAG

### Demonstration Value

- Demonstrates agent-assisted operations, not just chat
- Reduces operational complexity
- Shows how AI can automate configuration tasks in real environments

Highlights Intel Gaudi 3 as a practical solution for high-throughput GenAI inference

## Scenario 3: AI Agent–Enabled Image Generation (Intel AI for Enterprise RAG on Intel Gaudi 3)

### Overview

This demo illustrates how AI agents extend LLM capabilities, enabling image generation when the base LLM alone cannot perform the task.

### What the Demo Shows

- The user asks the LLM to draw an image.
- Without AI agents enabled, the request cannot be fulfilled.
- After enabling AI agents, the same request successfully generates an image.

### Demo Flow

1. *Model Configuration Overview*

   QChat briefly displays the current AI model settings.

2. *Image Request Without AI Agent*

   The user asks the LLM to generate an image. The request fails or returns a limited response.

3. *AI Agent Activation*

   The AI agent feature is enabled.

4. *Image Request with AI Agent*

   The same request is submitted again. The AI agent invokes the appropriate image generation service, and the image is successfully generated.

### Key Capabilities Highlighted

- AI agent orchestration across different model types
- Multimodal workflow support (text + image)
- Intel AI for Enterprise RAG's extensible microservice architecture

### Demonstration Value

- Shows how GenAI systems can go beyond text-only interactions
- Demonstrates easy extensibility without redesigning the application
- Highlights flexibility in integrating specialized AI services

## Scenario 4: RAG and AI Agent–Assisted Web Search

### Overview

This demo shows how QChat progressively expands its knowledge and reasoning ability by combining RAG and AI agents with web search.

### What the Demo Shows

- The LLM initially cannot answer certain questions.
- RAG enables answers based on curated knowledge.
- AI agents further extend capability by enabling web search.

### Demo Flow

1. *Initial Question (No RAG)*

   The user asks a question that the LLM cannot answer.

2. *Knowledge Base Setup*

   The user opens the Knowledge Base menu, prepares RAG data, and selects the knowledge source.

3. *RAG-Based Query*

   The same question is asked again. The system retrieves relevant content and provides a correct answer.

4. *Unknown Question (LLM + RAG)*

   A new question is asked that neither the LLM nor the Knowledge Base can answer.

5. *AI Agent with Web Search*

   The AI agent feature is enabled, allowing the LLM to perform web search.

6. *Final Answer*

   The system answers the question using external, real-time information.

### Key Capabilities Highlighted

- Retrieval-Augmented Generation (RAG)
- Dynamic knowledge base selection
- Tool-augmented AI via external service such as web search

### Demonstration Value

- Solves the LLM knowledge-cutoff problem
- Combines private knowledge with public information
- Improves answer accuracy and trustworthiness

## Summary

These QChat demonstrations show how Intel AI for Enterprise RAG enables a flexible, extensible, and production-capable GenAI platform on QCT AI POD—supporting heterogeneous hardware, AI agents, retrieval-augmented reasoning, and multimodal AI within a single, unified system.

# Conclusion

The convergence of High-Performance Computing, Artificial Intelligence, and emerging Generative and Agentic AI represents a fundamental shift in how modern computational

platforms must be designed and operated. Workloads are no longer isolated to simulation, training, or inference alone; instead, they increasingly span integrated pipelines that combine modeling, learning, reasoning, multimodal perception, and autonomous decision-making. Supporting these workflows demands infrastructure that is not only powerful, but also heterogeneous, composable, and open by design.

This technical paper has presented the QCT AI POD Reference Architecture as a practical realization of this converged paradigm. By unifying Slurm-managed HPC workloads and Kubernetes-based AI services within a single hybrid platform, and by embracing open-source ecosystems, it enables organizations to break down traditional silos between HPC and AI while preserving the performance characteristics required by each domain. The integration of Intel AI for Enterprise RAG further extends this foundation by providing a composable, cloud-native framework for enterprise-grade Generative and Agentic AI workflows.

The architecture is validated through QCT DevCloud, a real-world deployment spanning heterogeneous compute resources—including multiple generations of CPUs, GPUs, and Intel Gaudi 3 PCIe AI accelerators—while supporting secure, multi-user collaboration across shared infrastructure. Benchmarking results demonstrate that Intel Xeon 6 processors and Intel Gaudi 3 PCIe accelerators deliver efficient, scalable computing performance for both AI and HPC workloads. The QChat use case further illustrates how composable AI microservices, retrieval-augmented reasoning pipelines, and agent-driven automation can be orchestrated seamlessly on this unified platform, showcasing its readiness for enterprise Generative and Agentic AI deployments.

These results confirm that converged HPC–AI platforms are not merely a conceptual aspiration, but an achievable and necessary foundation for next-generation computing. The QCT AI POD provides a future-ready blueprint for organizations seeking to unify simulation, training, inference, and agentic intelligence within a single, scalable, open, and operationally consistent platform—positioning them to innovate with confidence as computational paradigms continue to evolve.

Among the many forward-looking directions enabled by this architecture, inference serving disaggregation represents a promising approach for improving hardware resource allocation, increasing utilization efficiency, and reducing Total Cost of Ownership (TCO) while maintaining service-level agreement (SLA) requirements for Large Language Model (LLM) deployments. Rather than over-provisioning accelerator capacity to meet peak demand—particularly for latency-sensitive workloads—disaggregated inference architectures enable resources to be dynamically composed and right-sized to workload characteristics.

As part of this exploration, we are evaluating heterogeneous inference configurations in which multiple accelerator classes—for example, Intel Gaudi 3 alongside other GPU-based accelerators—are deployed within a unified POD environment. In such architectures, distinct phases of LLM inference, including prefill and decode, may be mapped to the accelerators best aligned with their computational profiles. While this model introduces additional orchestration and scheduling considerations, it presents a viable pathway toward improving infrastructure efficiency, performance-per-dollar optimization, and scalable enterprise AI service delivery.

## References

[1] QCT GPU Server, “QuantaGrid D75E-4U”, Product Overview

Available: https://www.QCT.io/product/index/Server/rackmount-server/GPGPU-Xeon-Phi/QuantaGrid-D75E-4U

[2] Intel AI for Enterprise RAG Project, " Enterprise RAG GitHub Repository." GitHub.

Available: https://github.com/opea-project/Enterprise-RAG.

[3] Intel AI for Enterprise RAG Project, "User Guide".

Available: https://github.com/opea-project/Enterprise-RAG/blob/main/docs/Intel_AI_for_Enterprise_RAG_User_Guide_2.0.0.pdf

[4] G. Kresse and J. Furthmüller, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Physical Review B*, vol. 54, pp. 11169–11186, 1996.

[5] P. Giannozzi *et al.*, "QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials," *J. Phys.: Condens. Matter*, vol. 21, no. 39, p. 395502, Sep. 2009, doi: 10.1088/0953-8984/21/39/395502.