UnifiedBus™ (UB)

SuperPoD Reference Architecture White Paper

Contents

1 Critical Challenges in the AI Era

The advent of the AI era is fundamentally reshaping the global computing landscape. The demand for computing power, driven by the explosive growth of large language models (LLMs), is rising exponentially—doubling approximately every six months. This pace far surpasses the hardware advancement rate once guided by Moore's Law. Today, network scales have grown from hundreds of accelerators to tens of thousands, and will likely reach millions in the near future. This evolution places unprecedented requirements on system efficiency and reliability. Meanwhile, general-purpose computing continues to pursue higher resource utilization, while facing the emerging need for tight integration with AI workloads. This calls for breakthroughs in multi-chip pooling and interconnect communication. The accelerating evolution of computing brings forth a new set of challenges across various compute-intensive scenarios:

- **LLM pre-training**

  The Scaling Law continues to push the limits of model size and complexity. The total communication volume for sequence parallelism (SP), tensor parallelism (TP), and expert parallelism (EP) has grown nearly a hundredfold compared to smaller models, with a single batch now reaching hundreds of gigabytes. However, inter-node communication bandwidth is growing at only one-fifth the rate of compute power, making it a critical bottleneck for training scalability and performance.

- **LLM inference**

  Token processing output time (TPOT) is the key performance metric for LLM inference. With the rise of multimodal systems and AI agents, TPOT must be reduced to 5 ms, or even 1 ms. Meanwhile, sequence lengths are extending to millions of tokens, and the key-value (KV) cache size now far exceeds the capacity of high bandwidth memory (HBM). Different stages of inference also exhibit vastly varying demands for compute and memory, further complicating optimization.

- **Agentic AI**

  AI applications are evolving toward Agentic intelligence, where multiple components coordinate to accomplish complex tasks such as database operations, data preprocessing, and inference.

- **Virtualization**

  Today's general-purpose computing services are primarily deployed on virtual machines (VMs) or containers. However, traditionally 30%–50% of VM memory remains unused, and cross-node device pooling is often unavailable, resulting in low overall resource utilization. Additionally, when

VM live migration takes 100 ms–200 ms, users perceive service interruption. From a business continuity standpoint, achieving zero recovery point objective (RPO) and a recovery time objective (RTO) under 50 ms is essential for seamless service during live migration.

- **Database**

  To improve database linearity and peak performance, a layered, decoupled resource pooling architecture is required. However, this architecture demands large-scale, low-latency memory sharing. Achieving inter-node memory access latency in the hundreds of nanoseconds—less than one-tenth of current remote direct memory access (RDMA) latency—is key to aligning with database architectural evolution and unlocking higher peak performance.

To address these growing demands and support AI, general-purpose computing, and converged application scenarios, we propose the SuperPoD reference architecture for data centers in the AI era.

2 Introducing the SuperPoD Reference Architecture

The SuperPoD reference architecture is a next-generation computing system architecture purpose-built for AI-era data centers. It is built upon the UB interconnect protocol, which enables resource pooling and peer-to-peer coordination across heterogeneous components—including CPUs, NPUs, GPUs, memory (MEM), DPUs, scalable storage units (SSUs), and switches—to form a logically unified computing system.

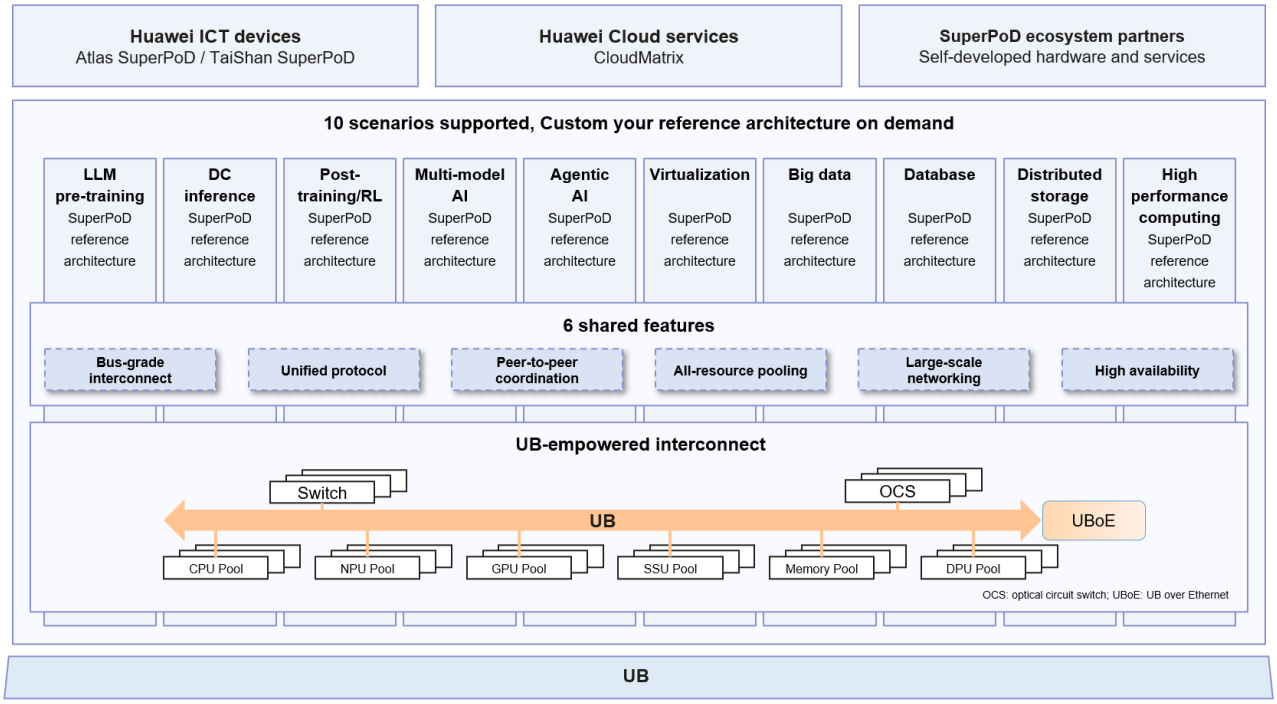


Figure 2-1 SuperPoD reference architecture based on UB

Key features of the SuperPoD reference architecture:

1. **Bus-grade interconnect**

   Built on UB, the system delivers bus-grade interconnect performance with sub-microsecond (hundreds of nanoseconds) synchronous memory-semantic access latency and 2–5 microseconds (μs) asynchronous latency. This meets the ultra-high concurrency demands of modern compute workloads. The architecture provides terabytes per second (TB/s) of component-to-component bandwidth—over 10 times higher than that of traditional data center networks.

2. **Unified protocol**

   The UB protocol standardizes interconnection and communication across diverse components—such as CPUs, NPUs, GPUs, MEM, DPUs, SSUs, and switches—without incurring protocol conversion overhead. It also provides a unified programming model, simplifying software development and system integration within the SuperPoD.

3. **Peer-to-peer coordination**

   The UB-based peer-to-peer coordination mechanism enables decentralized mutual access, invocation, and coordination among all components within the SuperPoD. This significantly improves inter-component memory access and communication efficiency.

4. **All-resource pooling**

   The UB extension operates seamlessly with both UB and Linux systems, handling key functions such as device management, memory allocation, communication control, and virtualization of SuperPoDs. It enhances resource pooling and access efficiency, improving overall system scalability.

5. **Large-scale networking**

   The SuperPoD architecture scales from a single node to 8,192 cards with over 90% linear scalability, and with plans to expand further to 15,488 cards and beyond. By leveraging UB over Ethernet (UBoE), it can interconnect millions of accelerators across clusters while remaining fully compatible with existing Ethernet infrastructure.

6. **High availability**

   The UB-based reliability framework provides application-agnostic fault detection and tolerance at the microsecond level. In an 8,192-card SuperPoD, the optical interconnect achieves over 6,000 hours mean time between failures (MTBF).

The UB interconnect protocol serves as the foundation of the SuperPoD reference architecture. It unifies I/O, memory access, and inter-processor communication within a single technology stack, enabling high-performance data transfer, efficient compute coordination, unified resource management, and flexible system composition.

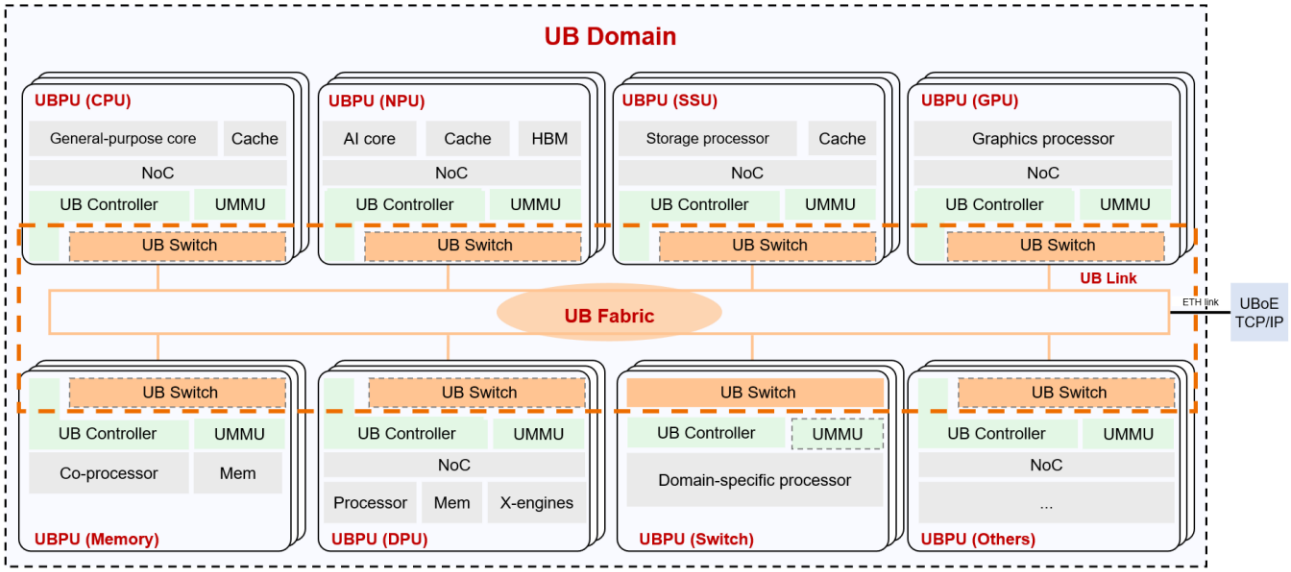


Figure 2-2 UB deployment

UB comprises the following components:

- A UB processing unit (UBPU) is a processing unit that supports the UB protocol stack and performs specific functions.
- A UB Controller is a component within a UBPU that implements the UB protocol stack and provides both software and hardware interfaces.
- A UB memory management unit (UMMU) is a component within a UBPU responsible for memory address translation and access permission control.
- A UB Switch is a mandatory component in the Switch and an optional component in other UBPUs. It forwards packets between UB ports.
- A UB link is a point-to-point connection between UBPUs.
- A UB domain is a collection of UBPUs interconnected via UB links.
- A UB Fabric encompasses all the UB Switches and UB links within a UB domain.
- UB over Ethernet (UBoE) uses Ethernet/IP networks to carry UB transactions, enabling smooth communication across UB domains.

By leveraging the UB protocol, the SuperPoD reference architecture empowers data centers to overcome diverse challenges of the intelligent computing era.

3 Scenario-Specific Applications of the SuperPoD Reference Architecture

Built upon UB, the SuperPoD reference architecture provides leading system capabilities for both AI and general-purpose computing. It delivers significant performance and resource utilization improvements across ten core business scenarios, including large language model (LLM) pre-training, data center (DC) inference, post-training and reinforcement learning (RL), multimodal content understanding and generation, Agentic AI, virtualization, big data, databases, distributed storage, and high-performance computing (HPC).

3.1 LLM Pre-training

To enhance LLM performance, the scale of parameters and datasets in mainstream foundational models has grown exponentially—from billions of parameters in early generations to the trillion level today. For example, DeepSeek V3 (released in December 2024) contains 671B parameters, while Kimi K2 (released in July 2025) exceeds 1T parameters. The workload of LLM pre-training features extreme parallelism, often spanning all nodes within a computing center. As compute requirements grow, infrastructure faces increasing challenges in achieving efficient parallelism:

- **Rising communication overhead due to large-scale parallel deployment**

  Taking the 1.8T-parameter GPT-4 model as an example, model deployment requires more than 10 TB of GPU memory, far beyond the capacity of a single GPU or even a single server. To overcome this limitation, the industry typically adopts tensor parallelism (TP), pipeline parallelism (PP), expert parallelism (EP), sequence parallelism (SP), and data parallelism (DP) for distributed deployment. However, this introduces massive communication overhead. With a hyperparameter configuration of TP8/PP16/EP8/SP4/DP64@seq8k/bs8k, the communication volume per card reaches 336 GB for TP, 268 GB for EP, and 86 GB for DP per global batch iteration. Such intensive communication demands ultra-low latency and high-bandwidth interconnects, particularly for intra-parallel-domain exchanges.

- **Bandwidth mismatch in traditional training clusters**

  Over the past eight years, single-node compute capability has increased by approximately 40 times, while inter-node communication bandwidth has only improved by 4–8 times. This imbalance between communication and computation has become a critical bottleneck for training efficiency. A

high proportion of communication overhead cannot be effectively overlapped with computation, resulting in low model FLOPS utilization (MFU) and underutilization of costly compute resources.

The SuperPoD reference architecture for LLM pre-training achieves high-bandwidth, low-latency NPU-to-NPU interconnection, significantly reducing communication overhead caused by highly parallelized deployment and improving training performance. In complex scenarios such as Mixture-of-Experts (MoE) models and long-sequence inputs, it delivers remarkable gains. For instance, in a PP8/DP64/EP64 configuration of DeepSeek V3, the SuperPoD reduces the unmasked communication ratio by 80% compared to traditional server clusters. With doubled increase in raw compute per NPU, overall training performance improves by up to 3-fold, demonstrating the crucial value of advanced architecture in boosting LLM training efficiency.

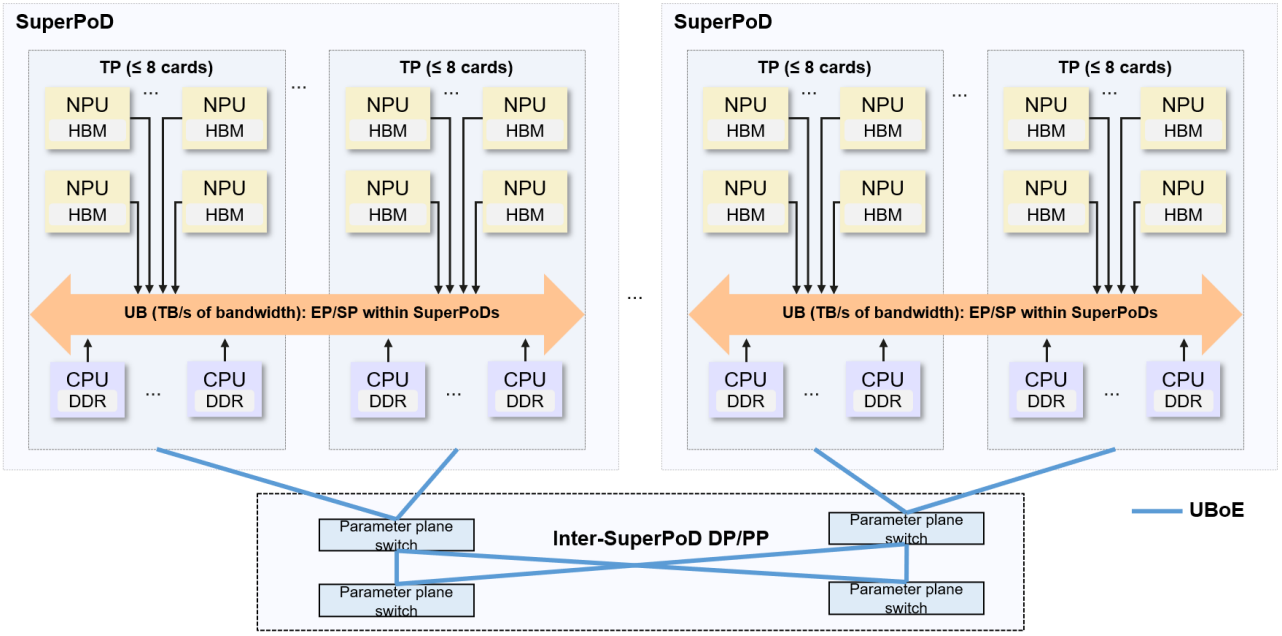


Figure 3-1 LLM pre-training SuperPoD reference architecture

3.2 Data Center Inference

Mainstream LLMs are evolving toward trillion-parameter sparsity, with growing total parameters and expert counts—for example, DeepSeek V3 (671B / 256 experts) and Kimi K2 (1.04T / 384 experts). Inference modes are shifting from single-card/single-node to multi-node expert parallelism. At the same time, sequence lengths for inference continue to expand. The Doubao 1.6 model became China's first commercial model to support 256K tokens, and future multimodal understanding applications involving multiple images or video streams are expected to drive this toward the million-token level. Under these trends, compute infrastructure for DC inference faces two major challenges:

- **Low-latency communication in multi-expert inference**

  Multi-server MoE inference involves frequent small-packet communication. In DeepSeek V3, for each token inference, its 58-layer MoE performs 58 rounds of dispatch and combine operations across nodes, with a minimum granularity of 7 KB. The communication volume increases linearly with batch size, reaching up to hundreds of MB. To balance single-user latency and multi-user concurrency, inference systems must achieve TPOT under 10 ms while maximizing batch size—meaning the communication ratio must stay below 20%. Consequently, each dispatch and combine operation must complete in under 10 μs, which is extremely challenging for RoCE-based networks.

- **KV cache management for long-sequence inference**

  Sparse attention is essential for efficient long-sequence processing. The inference process must store the entire KV cache for each user and offload it to node memory or NAND flash memory. This significantly improves batch size and throughput. For example, measurements on the DeepSeek V3 model with 64K–128K tokens show that KV cache offloading can increase batch size by 4–6 times and throughput by 30–40%.

The SuperPoD reference architecture for DC inference leverages high-bandwidth, low-latency NPU-to-NPU interconnects and unified semantics to enable efficient dispatch and combine operations, dramatically reducing multi-expert communication latency. Additionally, through a global resource pooling architecture, it supports multi-tier KV cache offloading and loading, enabling vertical expansion across multiple media layers and horizontal direct-through NPU-to-NPU data acceleration. This approach replaces compute with cache lookup, significantly improving overall inference throughput.

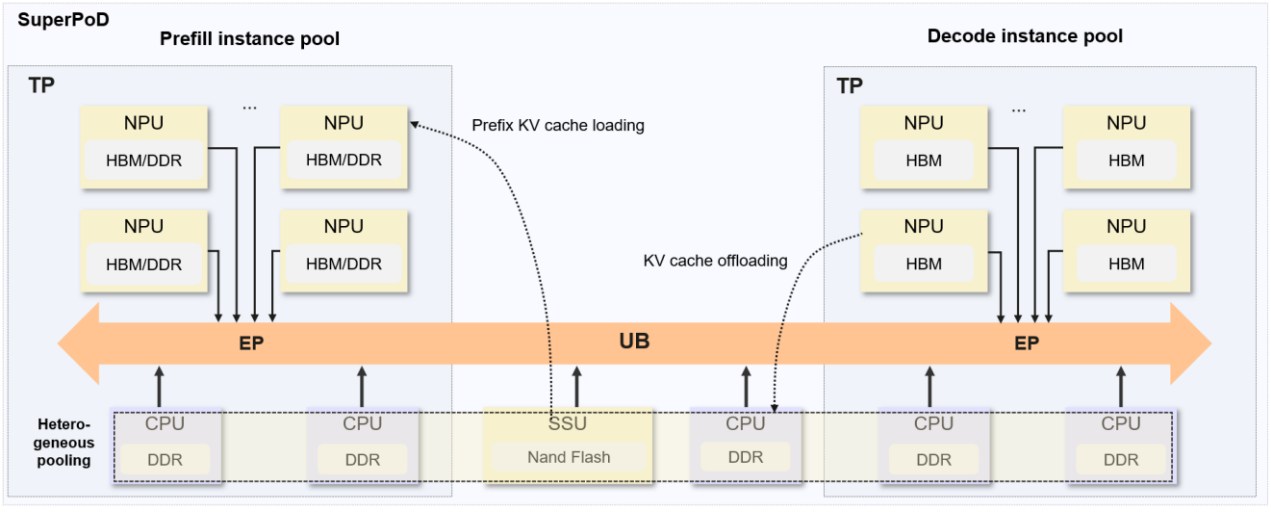


Figure 3-2 DC inference SuperPoD reference architecture

3.3 Post-Training and Reinforcement Learning (RL)

As LLMs grow larger and reinforcement learning from human feedback (RLHF) becomes increasingly critical, both computational and communication demands are becoming more complex and severe. The RLHF pipeline typically consists of three stages: supervised fine-tuning (SFT), reward modeling (RM), and reinforcement learning optimization (PPO). Each stage involves large-scale models with tens or even hundreds of billions of parameters. Advanced dialogue systems such as ChatGPT and Claude require multiple concurrent neural networks—including policy, value, reward, and reference models—whose total parameter count can reach the trillion level. This places tremendous pressure on both compute and network infrastructure:

- **Communication bottlenecks in parameter and gradient synchronization**

  RLHF training differs from pre-training due to its multimodal collaboration and frequent synchronization characteristics. During the PPO stage, complex data flows occur between policy inference, reward calculation, advantage estimation, and policy updates. In a distributed RLHF setup, Actor nodes generate responses, Critic nodes compute value functions, Reward nodes evaluate response quality, and Learner nodes perform policy optimization. These components require massive parameter exchanges and gradient synchronization.

  Traditional server-stacked clusters face bottlenecks when handling the multimodal communication pattern of RLHF. With frequent exchange of massive parameter and gradient information, network bandwidth quickly becomes a limiting factor. The complex dependencies between models cause communication latency to be amplified multiple times. Taking GPT3's training configuration as an example, when using a 175B parameter Policy model and a 6.7B parameter Reward model, the inter-model communication volume for a single PPO iteration can exceed 500 GB. The network bandwidth bottleneck of traditional server-stacked clusters often leads to a 40% to 60% reduction in overall training efficiency.

- **Performance challenges in large-batch sample generation and parallel inference**

  Large-scale RLHF requires generating numerous candidate responses using the current Policy model, then scoring and ranking them via the Reward model. Each prompt may produce dozens of responses, amounting to tens of thousands of samples. The system must efficiently handle large-scale sequence generation and parallel inference while providing timely feedback of reward scores to the optimization process—any delay directly slows convergence and influences stability.

The SuperPoD reference architecture for RL training addresses these bottlenecks through high-bandwidth NPU-to-NPU interconnect, cutting inter-model synchronization latency from seconds or tens of seconds down to milliseconds or seconds. This enables more frequent PPO and GRPO updates, improving learning efficiency and convergence speed. Moreover, the SuperPoD supports large-scale parallel inference and generation across hundreds of NPUs, allowing larger batch sizes within the same hardware footprint and increasing overall sample throughput. In large-scale dialogue model RLHF training, the Critic–Reference–Reward–Actor system deployed on the SuperPoD achieves multi-fold

improvements in convergence speed and significantly higher inference throughput compared to traditional server-stacked clusters. For example, in training a 70B-parameter model, traditional clusters typically exhibit tens of seconds of inference latency for a 32K batch size, whereas the SuperPoD reduces this latency to just a few seconds, substantially boosting sample generation efficiency.

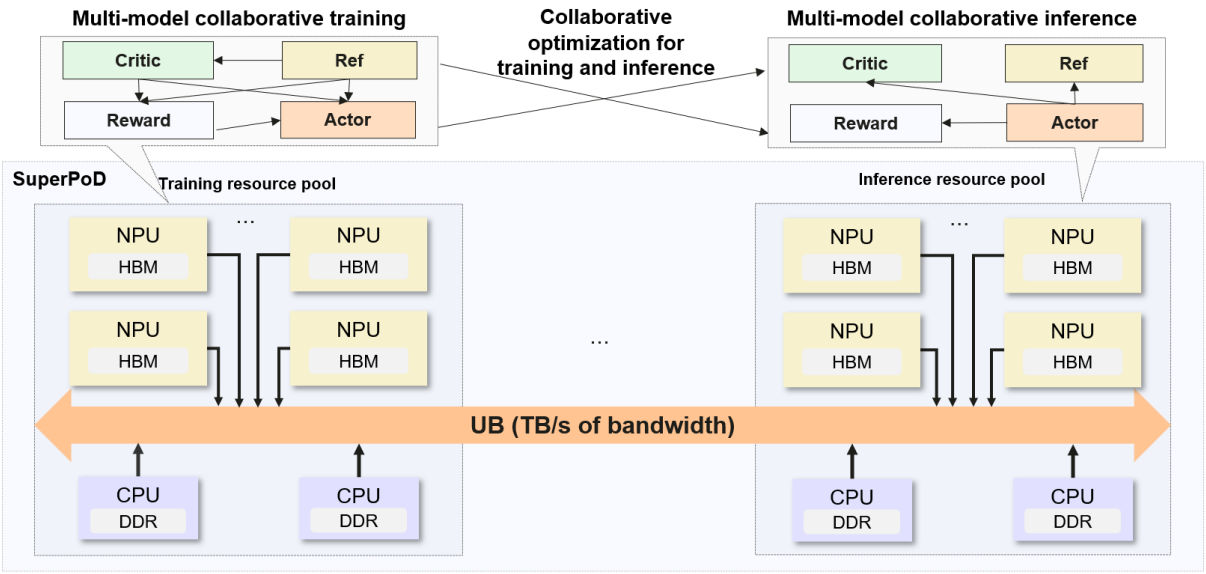


Figure 3-3 RL SuperPoD reference architecture

3.4 Multimodal Content Understanding and Generation

Multimodal content understanding and generation have emerged as a pivotal application frontier in AI, with three transformative trends reshaping the field:

1. Multimodal models are transitioning to sparse architectures scaling to hundreds of billions of parameters while achieving real-time content generation capabilities (e.g., producing a 5-second 720p video in 5 seconds). This advancement is driving a fundamental shift from single-card/server deployments to distributed, multi-node parallel computing architectures.
2. Multimodal workloads are evolving from single-model/task scenarios to multimodal/task scenarios. Different tasks, such as generation, editing, and understanding, have significantly varying computing and memory requirements, driving deployments toward heterogeneous setups that combine high-performance and high-bandwidth NPUs.
3. In multimodal simulation and generation applications like autonomous driving and embodied intelligence, 3D Gaussian Splatting (3DGS) has emerged as a mainstream technique, with key scenarios involving reconstruction of tens of millions of Gaussians within minutes.

The aforementioned trends pose critical challenges to the computational foundation.

- **Multi-node parallel multimodal inference**

  Techniques such as sequence parallelism and expert parallelism require multi-node parallel bandwidth of hundreds of GB/s, which exceeds current networking capabilities.

- **Concurrent multi-task processing in heterogeneous NPU deployments**

  Token-level modality grouping and reordering for dynamic computing resource allocation generates numerous small packets, imposing strict latency requirements on communications.

- **High-performance distributed 3DGS**

  Reconstructing tens of millions of Gaussians requires compressing single iterations to tens of milliseconds, necessitating synchronization of millions of ellipsoid gradients (including position, covariance, and opacity) with inter-GPU bandwidth reaching hundreds of GB/s.

- **Gradient transmission of massive dispersed Gaussians**

  Gradient data may be scattered across discrete memory spaces, necessitating a large volume of fine-grained communications (with packet sizes as small as tens of bytes). This requires support for low-latency small-packet transmission.

The multimodal content understanding and generation SuperPoD reference architecture uses the UB technology to address these challenges through core innovations. First, UB aggregates multiple discrete GPUs into a logically unified resource pool, significantly improving the generative efficiency in distributed 3DGS reconstruction. Second, its heterogeneous NPU orchestration facilitates seamless networking across different types of NPUs, which enhances both dynamic resource scheduling capabilities (such as dynamic ratio configuration and dynamic parallel policy optimization) and overall flexibility. Third, memory-semantic communication achieves ultra-low latency for small-packet transmissions. This supports token-level grouping and reordering in multi-task inference, boosting inference efficiency, while also enabling rapid synchronization of gradients from sparsely distributed Gaussians in 3DGS, significantly accelerating the generation of 3D rendered views.

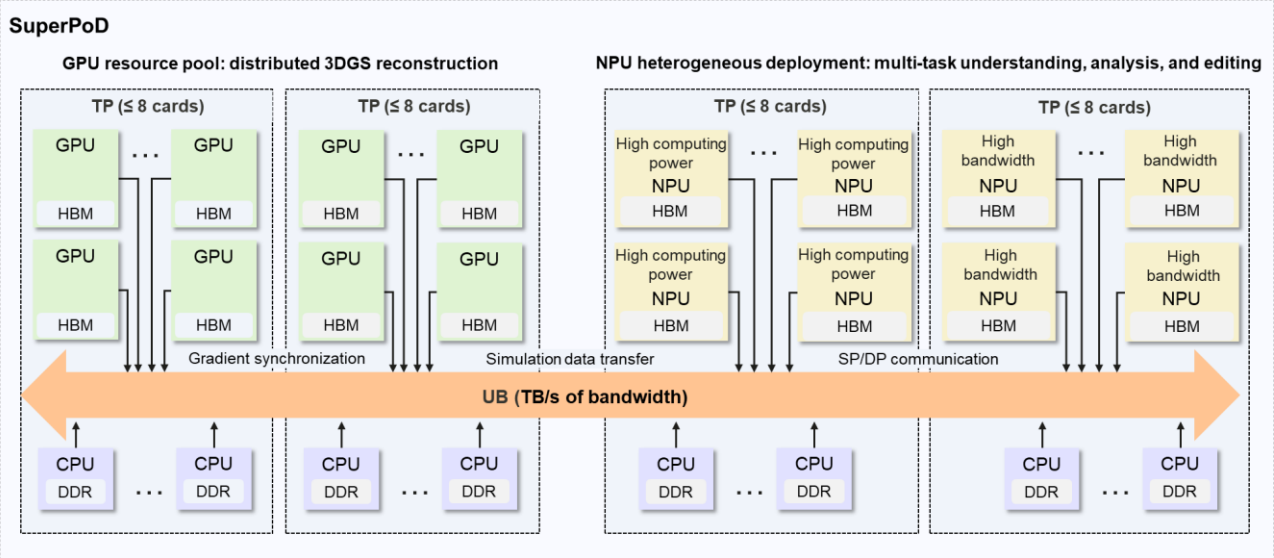


Figure 3-4 Multimodal understanding and generation SuperPoD reference architecture

3.5 Agentic AI

Gartner has ranked Agentic AI as the top strategic technology trend for 2025, further predicting that by 2028, 15% of daily decisions globally will be handled by AI agents with autonomous decision-making capabilities. This marks the evolution of AI into intelligent agents capable of autonomous perception, reasoning, decision-making, and task execution, ushering in the era of Agentic AI. For example, by further integrating multimodal capabilities (such as text, image, and speech) with physical world interaction (like robotic control), AI is being propelled to a shift from passive response to proactive action. By merging with the Internet, Agentic AI creates a distributed agent network designed for collaboration on a large scale.

- **New service workload challenges from Agentic AI**

  In Agentic AI, the collaboration of multiple LLMs significantly increases system complexity, which involves four major aspects: training, inference, general-purpose computing, and database management. Beyond the existing challenges in training, inference, reinforcement learning, and multimodal processing, Agentic AI brings new hurdles. These include the storage and memory access demands arising from long-chain reasoning and long-term memory, resource utilization inefficiencies caused by long-tail tasks and heterogeneous hardware bottlenecks, as well as rapid context switching overhead and frequent image retrieval overhead associated with multi-task and multi-replica agent operations.

- **New computing system challenges from Agentic AI**

  The evolution of Agentic AI's workload scaling patterns presents significant challenges to system design, including a surge in memory requirements, increased complexity in resource scheduling, heightened communication randomness, and issues with storage persistence management. As

diverse environments are changing dynamically, systems must adapt to continuously increasing workloads, ensuring efficient memory utilization, precise resource scheduling, and persistent data storage to meet long-term operational demands.

The Agentic AI SuperPoD reference architecture leverages collaborative and equal-priority computing across diverse computing resources to achieve high-throughput optimizations, including multimodal training data preprocessing, Retrieval-Augmented Generation (RAG), and slow-thinking reasoning. By utilizing the UB global resource pool for lossless live migration, it effectively addresses the exponential growth in heterogeneous data retrieval, such as KV cache and RAG, driven by long-term memory in agents. Furthermore, the architecture enables millisecond-level loading of sandboxed images through UB, ensuring strong workload affinity for Agentic AI workloads.

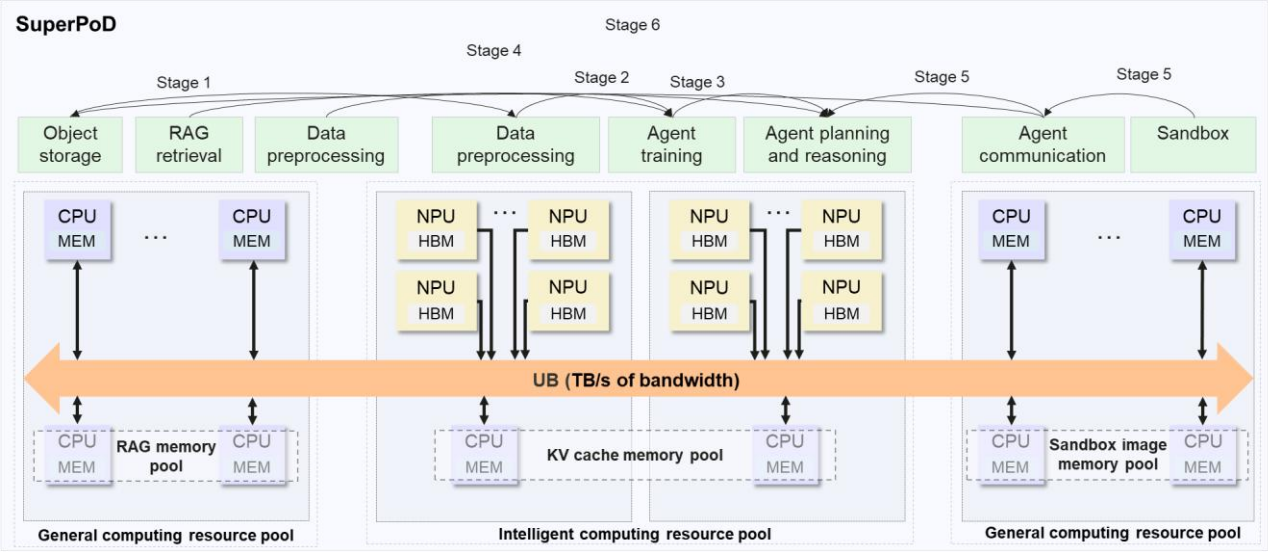


Figure 3-5 Agentic AI SuperPoD reference architecture

3.6 Virtualization

While traditional cloud computing virtualization enhances data center deployment flexibility and resource utilization by abstracting compute, storage, and network resources, its practical implementation still encounters the following key challenges:

- **Resource fragmentation leading to reduced utilization**

  Dynamic creation and termination of VMs lead to fragmentation of CPU, memory, and storage resources across physical servers. In conventional cloud environments, this leads to 20% to 25% of memory being stranded, unavailable for effective allocation or optimization, consequently driving the overall resource utilization rate below expected levels.

- **Bottlenecks in live migration efficiency of VMs**

  During the migration of high-load VMs, an excessively high dirty page rate significantly prolongs the migration process or even causes it to fail. Additionally, the destination host must meet the demand for continuous resource availability, yet resource fragmentation often impedes the successful completion of the migration.

- **Low memory resource utilization**

  Constrained by the incompressibility and real-time access requirements of memory resources, traditional cloud platforms widely adopt a reservation-based allocation mechanism. This results in low actual memory utilization, with nearly half of VMs leaving 30% to 50% of their allocated memory unused, ultimately leading to significant resource waste.

- **Inflexibility due to fixed resource allocation**

  The preset CPU-to-memory ratio of general-purpose servers, such as 1:4, cannot meet the differentiated requirements of memory/compute-intensive applications, causing unbalanced resource configuration.

The virtualization SuperPoD reference architecture supports memory overcommitment ratios exceeding 25%, thereby significantly boosting resource utilization. This breakthrough provides an optimized solution for next-generation cloud and virtualization platforms.

- ✓ **Elastic resource allocation**

  The SuperPoD architecture leverages a UB-based global memory pool to enable dynamic resource orchestration, increasing memory allocation efficiency from 80% to 95% while delivering unprecedented scheduling flexibility.

- ✓ **Efficient lossless live migration**

  By leveraging the high-bandwidth, low-latency communication provided by UB and unified semantics technology, the architecture enables ultra-fast live migration of VMs with the latency as low as 50 ms. This effectively mitigates the risk of memory shortages caused by memory overcommitment.

- ✓ **Transparent cold and hot memory tiering**

  The UB's memory page access analytics mechanism enables efficient identification and swapping of hot and cold memory, with performance overhead constrained within 5%, thereby enhancing memory resource utilization without application awareness.

- ✓ **Support for ultra-large memory instances**

  Leveraging UB, the architecture enables the creation of cloud instances with ultra-large memory capacities (exceeding 3 TB), addressing the diverse resource requirements of memory-intensive workloads.

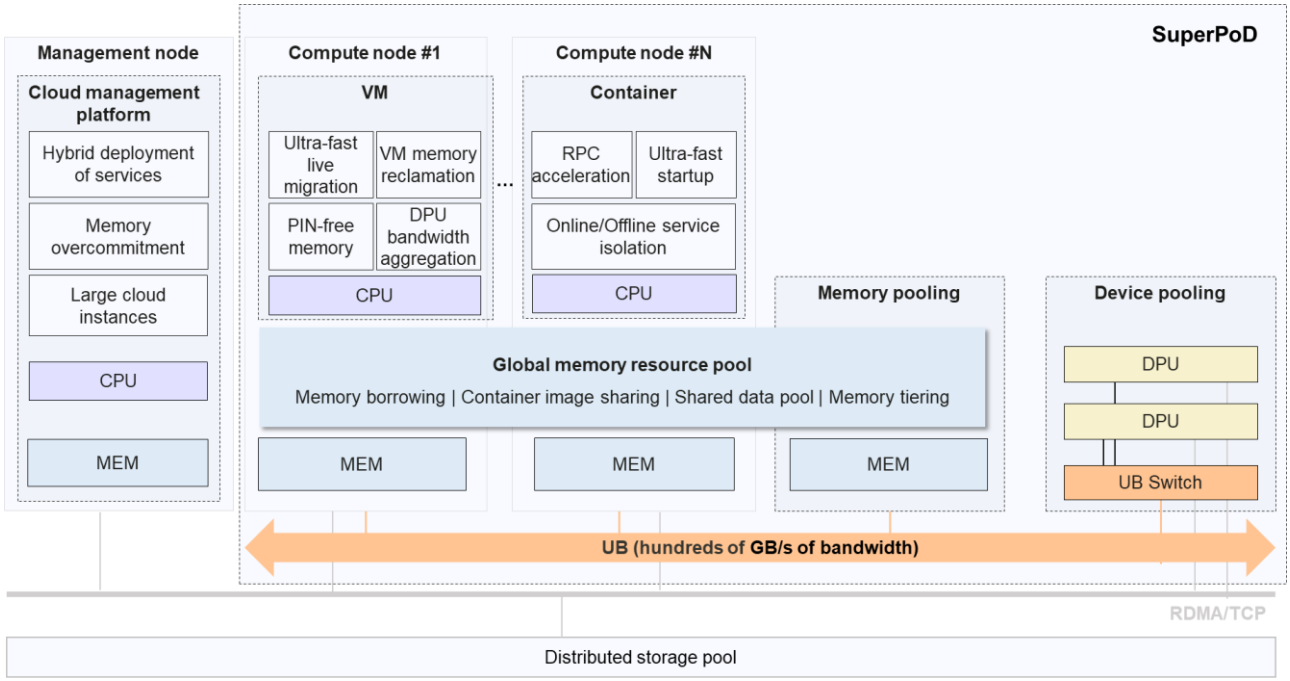


Figure 3-6 Virtualization SuperPoD reference architecture

3.7 Big Data

Conventional big data platforms commonly face two critical performance bottlenecks that constrain both data processing efficiency and resource utilization.

- **Resource provisioning imbalance and waste**

  Static resource allocation mechanisms lead to a mismatch between provisioned resources and actual workload demands, resulting in uneven resource utilization. In extreme cases, only 30% to 40% of memory resources are effectively used. In addition, to prevent task failures, O&M personnel tend to over-provision resources, which in turn leads to resource idleness.

- **Data exchange efficiency constraining overall performance**

  Data read/write and exchange efficiency during cross-node data shuffle operations becomes a major system performance bottleneck. The pressure on network and disk I/Os drives the average CPU utilization down to just 50% of its peak value.

The big data SuperPoD reference architecture delivers an optimized solution by leveraging high-bandwidth, low-latency communication and global resource pooling.

✓ **Resource pooling and intelligent overcommitment**

By leveraging UB to achieve flexible memory pooling, borrowing, and sharing within the SuperPoD domain and combining a resource overcommitment strategy and optimized pooling operators tailored for big data workloads, the execution efficiency of Spark tasks is improved by 30%.

✓ **In-depth optimization of shuffle performance**

Deep optimization of the Spark shuffle mechanism, powered by UB communication, boosts Spark processing performance by 10%, significantly reducing the completion time of batch jobs.

✓ **Efficient RPC communication**

Reconstructing the Remote Procedure Call (RPC) mechanism using UB enables shared memory and ultra-low-latency communication with high bandwidth. This minimizes serialization/deserialization overhead and eliminates redundant memory copying, ensuring ultra-fast and highly efficient communication for real-time computing and similar workloads.

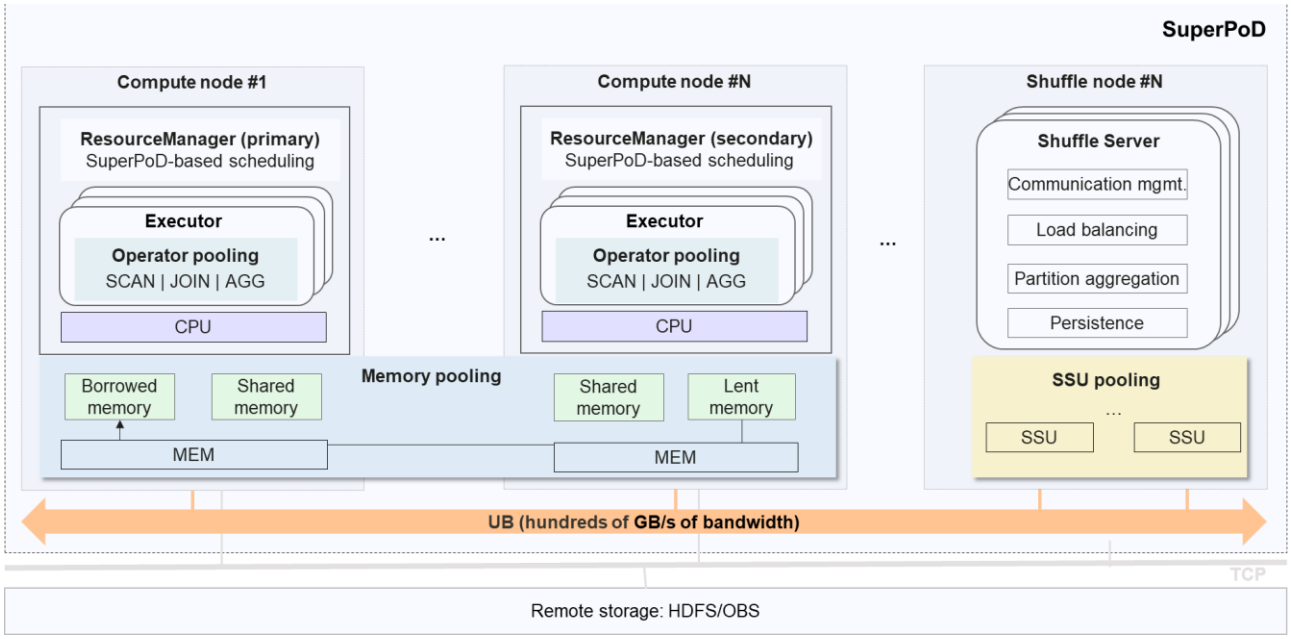


Figure 3-7 Big data SuperPoD reference architecture

3.8 Databases

As computing power demands escalate, database applications encounter critical challenges under traditional architectures:

- **Architecture limitations and expansion bottlenecks**

  In traditional primary/secondary architectures, write operations are concentrated on the primary node, turning it into both a performance bottleneck and a single point of failure (SPOF). With scalability dropping below 70%, this severely restricts both write scalability and high availability of the system.

- **Network and I/O performance constraints**

  Cross-node data synchronization, distributed transaction coordination, and log replication impose heavy network overhead and bandwidth pressure, making the system highly sensitive to latency. The network protocol stack overhead may exceed 33%, leading to substantial waste of CPU resources. Log synchronization further exacerbates I/O pressure, degrading performance in high-concurrency and high-throughput scenarios. I/O bottlenecks and cache misses lead to long-tail SQL queries, with response times exceeding 10 ms.

The database SuperPoD reference architecture delivers the following capabilities:

✓ **Multi-primary-node architecture**

Relying on UB's global resource pooling and high-bandwidth, low-latency communication, the architecture enables shared memory services to support multi-primary-node deployments. This design effectively eliminates write performance bottlenecks with a linear scalability greater than 0.75. By removing single points of failure on write nodes and supporting dynamic online expansion of database instances, it significantly improves system reliability and flexibility.

✓ **OLTP performance acceleration**

By leveraging UB's global resource pooling and unified semantic access, the architecture extends the database buffer pool into a global buffer pool, establishing efficient multi-level page caching. Combined with transparent hot-cold data migration between local and remote memory, it increases the hit ratio for accessing hot data in local memory, thereby boosting TPC-C performance by 20%.

✓ **OLAP performance optimization**

Based on UB's global resource pooling and unified semantic access, the shared memory service accelerates execution of OLAP SQL operators, enables efficient data sharing between row- and column-hybrid storage engines, and enhances OLAP query performance, thereby achieving a 65% improvement in TPC-H performance.

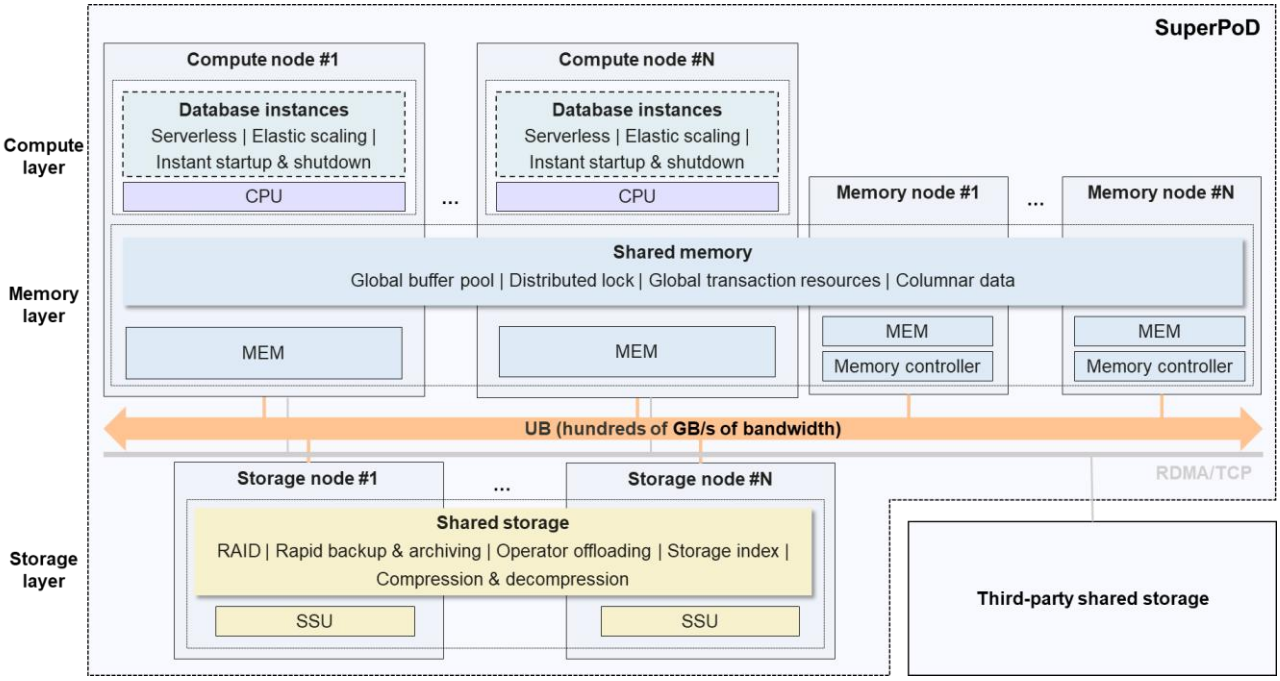


Figure 3-8 Database SuperPoD reference architecture

3.9 Distributed Storage

Conventional distributed storage systems face the following challenges:

- **Discrete hardware resources**

  In conventional storage systems, SSD capacity is statically allocated per server, causing resource utilization to be constrained by individual physical servers. This results in the SSD bandwidth utilization of only about 25%, leading to inefficient infrastructure and complex operations and maintenance (O&M).

- **Weak software-hardware synergy**

  The CPU-centric architecture suffers from fundamental flaws: critical tasks such as EC encoding/decoding, CRC validation, and cross-node data migration contention for host CPU resources, turning it into a system performance bottleneck and thus constraining the scalability of distributed storage. Taking the garbage collection task in distributed storage systems as an example, the CPU reads data from the source disk and rewrites it to the destination disk. This process generates network and I/O stack overheads for multiple times, which competes with business-side processing resources and results in a drop in effective IOPS by 25% to 35%.

Built on the principle of equal-priority collaboration across diverse computing resources, the distributed storage SuperPoD reference architecture delivers the following technological innovations:

✓ **Storage pooling**

By utilizing UB's global resource pooling and unified semantic access capabilities, the architecture constructs a cross-node storage resource pool. This enables logical abstraction and unified scheduling of physical media and supports on-demand dynamic allocation of multi-node storage capacity and performance resources. By eliminating resource silos, the architecture improves SSD bandwidth utilization by up to 60%.

✓ **Heterogeneous computing power offload engine**

Relying on the equal-priority collaboration across diverse computing resources, the architecture enables equal interconnection and access channels across devices like CPUs, GPUs, and SSUs. Through a hardware-level service offload engine, storage-side tasks such as garbage collection are offloaded from the host CPU to the SSU, delivering a measured IOPS performance improvement of 25% to 35% and ensuring performance for high-density storage scenarios. Furthermore, by offloading storage-side Erasure Coding (EC) reconstruction tasks from the host CPU to the SSU and leveraging UB's ultra-high bandwidth, reconstruction speed is accelerated by 6 times, thus ensuring reliability for high-density storage scenarios.

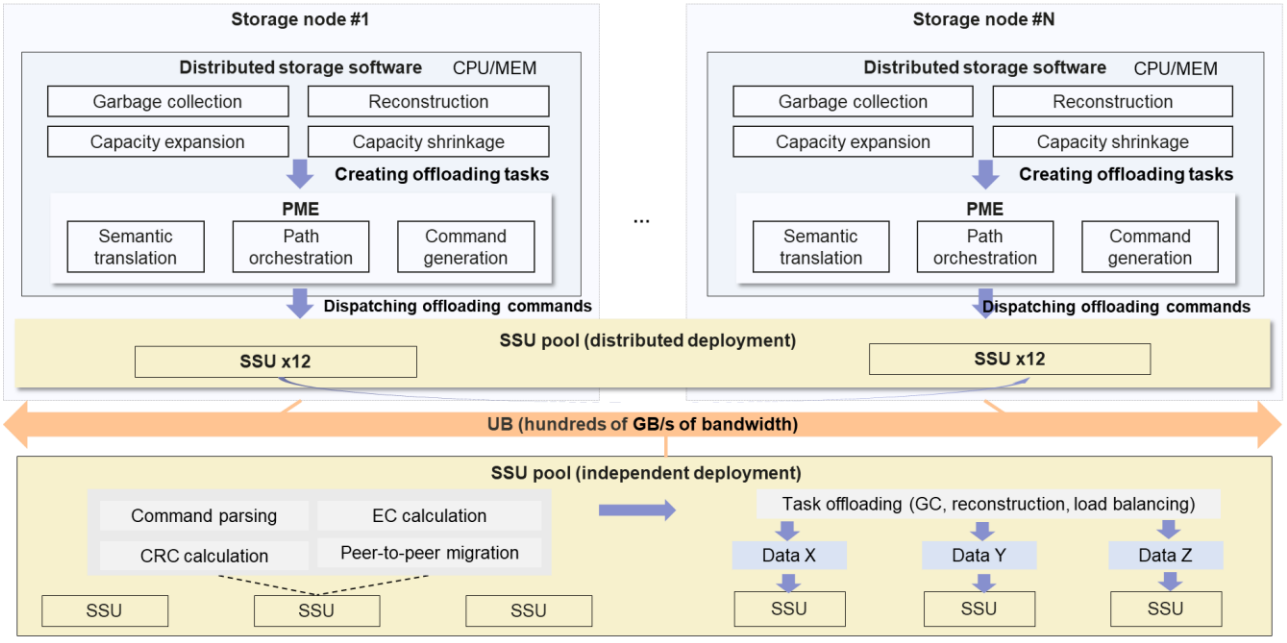


Figure 3-9 Distributed storage SuperPoD reference architecture

3.10 High-Performance Computing

As computing demands in scientific research continue to rise, high-performance computing (HPC) clusters are expanding in scale. At the same time, the past decade's AI breakthroughs have catalyzed the AI for Science (AI4S) paradigm shift. The convergence of HPC and intelligent computing is creating new architectural and performance challenges for HPC systems.

- **Communication performance challenges**

  As computing capabilities of processors continue to advance, the proportion of HPC applications in communication will further increase, potentially reaching up to 30% in some cases. To reduce the communication overhead while enhancing application performance and scalability, it is essential to boost inter-node communication bandwidth for large-packet scenarios and reduce transmission latency for small-packet scenarios. In AI4S applications, training and inference involve a large amount of distributed matrix computing tasks, which generates dependent compute and communication operators. Fine-grained pipelining of these operations is commonly used to hide overhead. However, this approach generates KB-level small to medium packet communications, which adversely impacts communication efficiency. Thus, reducing communication latency becomes critical.

- **Network congestion and reliability challenges in ultra-large clusters**

  The computing power of industry-leading HPC clusters has already reached the exascale level and is now evolving toward 10-exascale level, requiring support for ultra-large-scale networks connecting tens of thousands to a hundred thousand of nodes. At such extreme scale, these clusters face significant network congestion issues, necessitating optimized routing and congestion control algorithms to enhance performance. Furthermore, traditional fat-tree network topologies encounter challenges related to high volume of optical modules and network reliability. To address this issue, the network topology needs to be optimized to enhance networking reliability and reduce the number of optical modules.

- **Performance deterioration from multiple memory copies in storage I/Os**

  Conventional high-performance storage systems use a proprietary client-based architecture that involves multiple memory copies during disk writes. Data is first copied from the application memory on the compute node to the client memory, then transferred over the network to the storage node memory, and finally written to the disks. These multiple memory copies increase I/O latency, highlighting the need to reduce the number of memory copies along the I/O path to lower the I/O latency.

The HPC SuperPoD reference architecture drives technological innovations targeting next-generation ultra-large-scale networking and AI4S applications:

- ✓ **Optimization in collaborative software-hardware communication performance**

  By delivering interconnection bandwidth of hundreds of GB/s to TB/s through UB, and combining topology-aware communication optimization and job scheduling, the architecture accelerates large-packet communication for communication-intensive applications such as molecular dynamics, meteorology, and fluid dynamics. This results in an end-to-end performance improvement of up to 10% for typical applications. In addition, memory-semantic-based communication optimization within the SuperPoD reduces small-packet latency from microseconds (us) to hundreds of nanoseconds (ns), boosting end-to-end performance by 8% for typical applications. For distributed matrix computing in AI4S applications, memory-semantic-based operators are used to enable fine-grained pipelining of computing and communication. This allows direct access to HBM data on the peer NPU, reduces memory access overhead, lowers small-packet communication latency, and improves both training and inference performance.

- ✓ **Optimization in ultra-large-scale networking topology and network congestion control**

  For HPC workloads with extensive nearest-neighbor communication, the architecture leverages UB within SuperPoD to enhance inter-process communication performance through a full-mesh/Torus + Clos topology. Between SuperPoDs, a network architecture with a higher overcommitment ratio is adopted to support ultra-large-scale networking. By utilizing UB's link-layer virtual channels to prevent deadlocks caused by detours in direct paths, and incorporating spatial-temporal load balancing to fully utilize both minimal and non-minimal paths along with per-packet/per-flow dynamic routing technologies, the solution reduces network congestion, improves bandwidth utilization, and achieves high-performance, large-scale, and highly reliable networking.

- ✓ **UB-based data pass-through mechanism and unified storage architecture**

  UB enables direct data transfer from compute nodes to storage media, completely bypassing the storage node memory. This eliminates the extra overhead caused by memory copies and delivers a 20% improvement in I/O performance. In AI4S applications, SuperPoDs for both AI computing and HPC can access shared high-performance distributed storage through UB, thereby reducing data migration overhead.

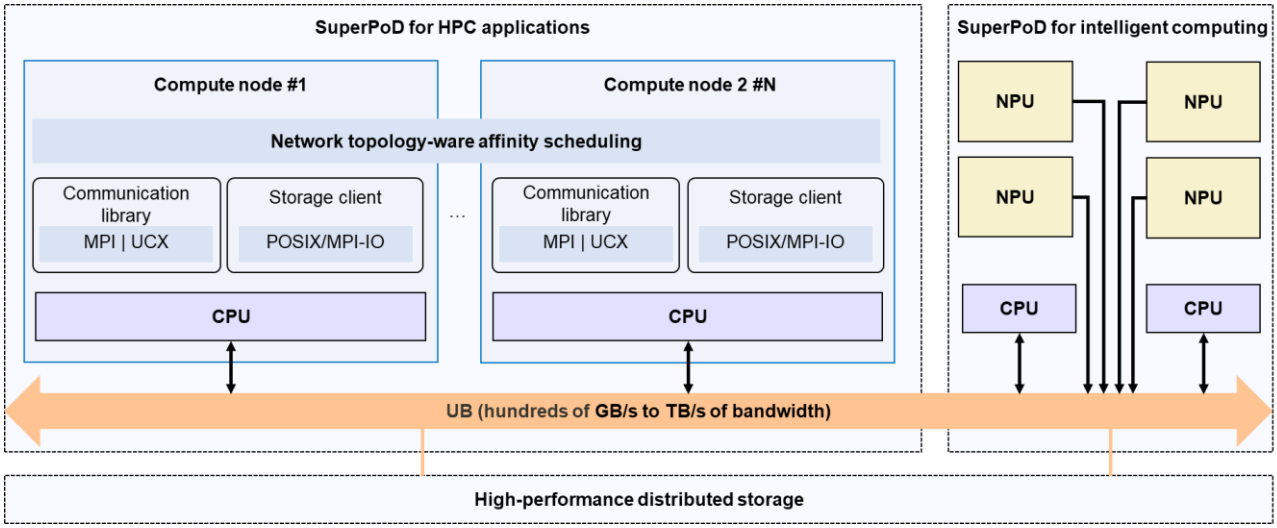


Figure 3-10 HPC SuperPoD reference architecture

4 UB Protocol Stack and Mechanism

4.1 UB Protocol Stack

The UB protocol stack is layered, consisting of the physical layer, data link layer, network layer, transport layer, transaction layer, function layer, UMMU, and UB Fabric Manager (UBFM). The following figure shows the protocol stack, in which an Entity is a basic unit of global communication and URMA is Unified Remote Memory Access.

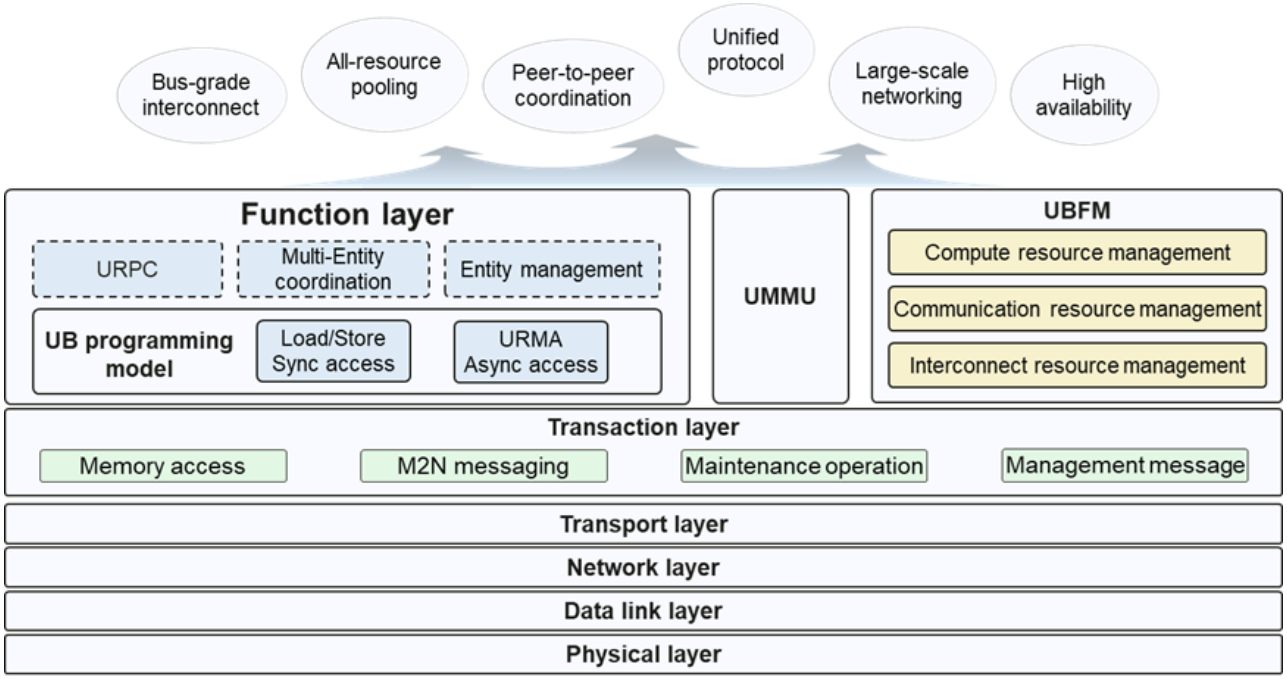


Figure 4-1 UB protocol stack

- **Physical layer:** provides high speed at low latency, fully utilizes physical channel margins, and supports linear direct drive optics technology.
- **Data link layer:** supports point-to-point flow control and retry, with efficient transmission formats and diverse topologies.
- **Network layer:** enables large-scale networking, dynamic routing, and multipath balancing, and provides the optimal end-to-end latency and bandwidth.

- **Transport layer:** supports bandwidth aggregation, connection sharing, and end-to-end reliable transmission, ensuring reliable scaling.
- **Transaction layer:** provides unified transaction operations to support stable programming interfaces.
- **Function layer:** provides simplified programming models (such as load/store and URMA) and efficient collaboration mechanisms.
- **UBFM:** manages computing, communication, and interconnect resources, and efficiently applies resources.
- **UMMU:** performs global address translation and memory access permission check to implement global memory sharing.

For more details on the UB protocol stack, see the *UnifiedBus™ (UB) Base Specification*.

4.2 UB Enabling SuperPoDs

4.2.1 Bus-grade Interconnect

- **100 ns to μs memory semantics latency**

  Based on the unified addressing and access control mechanism, UB allows compute units in UBPUs to directly initiate synchronous and asynchronous memory access instructions. This reduces control command interactions to achieve a low latency from 100 ns to several μs. The UB memory semantics supports a single transaction of 64 bytes to 4 KB or variable lengths, reducing out-of-order processing workload and boosting transmission efficiency. In addition, the Flit transmission mechanism at the data link layer further ensures low-latency transmission and forwarding.

- **TB/s-level bandwidth**

  UB enhances lane speeds and employs a multi-port, multi-path aggregation design to deliver high bandwidth in the AI era. It uses multiple FEC modes at the physical layer and retry techniques at the data link layer to lower the BER requirements. This increases the single-lane speed to 118 Gbit/s, surpassing the IEEE Ethernet standard of the same generation.

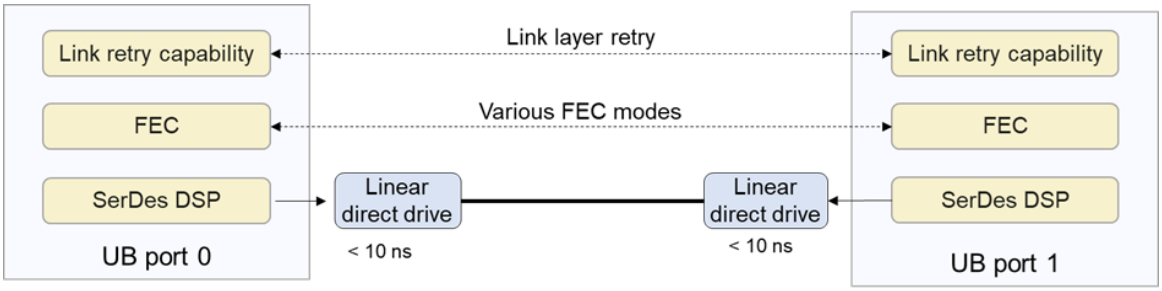


Figure 4-2 High speed of a single port

Multi-port aggregation and high-density optical-electrical interconnect technologies enable TB/s-level bandwidth between UBPUs. UB allows load/store semantics and URMA semantics to share multi-port bandwidth, achieving multi-path transmission across multiple ports.

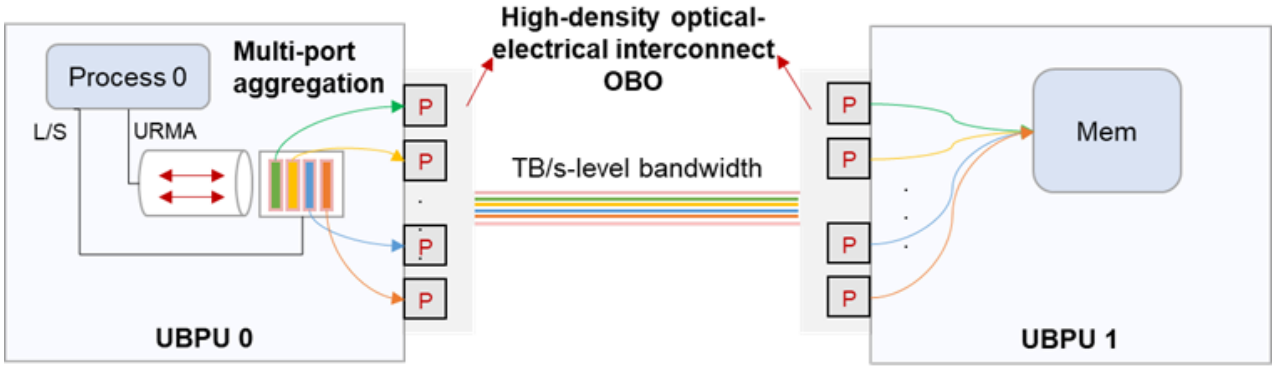


Figure 4-3 Multi-port aggregation and high-density optical-electrical interconnect

4.2.2 Unified Protocol

UB can map resources such as memory and registers directly to global addresses. Compute units use UB to access and synchronize data globally. Compute unit instructions can schedule UB Controllers, which natively aligns with the out-of-order and parallel nature of compute units. By computing hardware, UB classifies all user operations into memory access, message transfer, process calling, and resource management. UB uses a single protocol stack for all these operations, eliminating the need for protocol conversions and allowing software developers to fully leverage hardware capabilities.

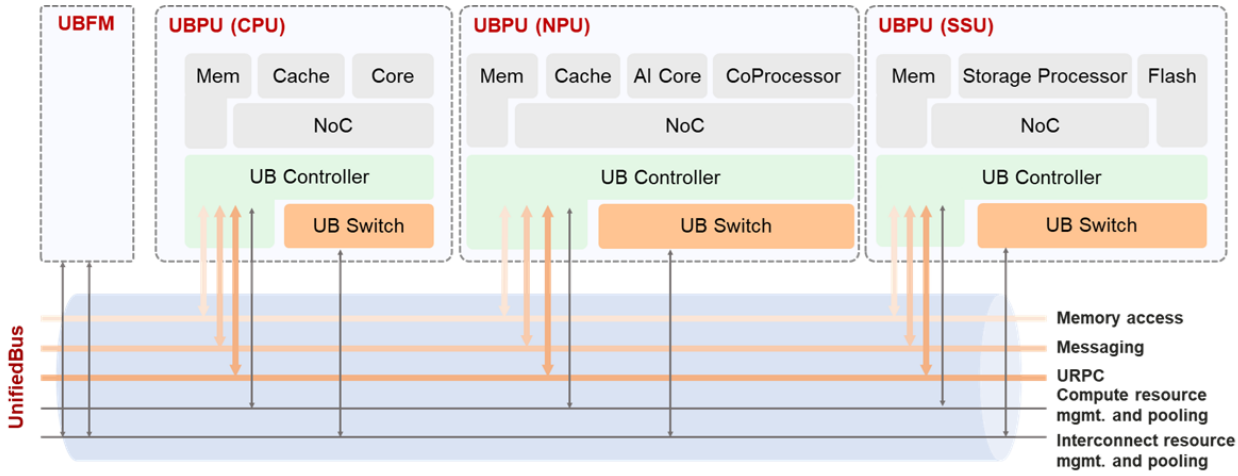


Figure 4-4 Unified programming model

The UB transaction layer sets up the groundwork for the unified programming model by providing four types of transactions: memory, message, maintenance, and management. Memory transactions enable both synchronous and asynchronous memory access between UBPUs. Message transactions provide one-to-one, one-to-many, many-to-one, and many-to-many message exchange across UBPUs. Maintenance transactions update the internal module status of remote UBPUs, such as the access security and cache status. Management transactions configure UBPUs and report operating status, such as address settings and fault notification. In addition, UB allows flexible adjustment of latency, bandwidth, and reliability settings to deliver optimal performance in diverse scenarios and communication ranges. The following figure illustrates the UB protocol configurations.

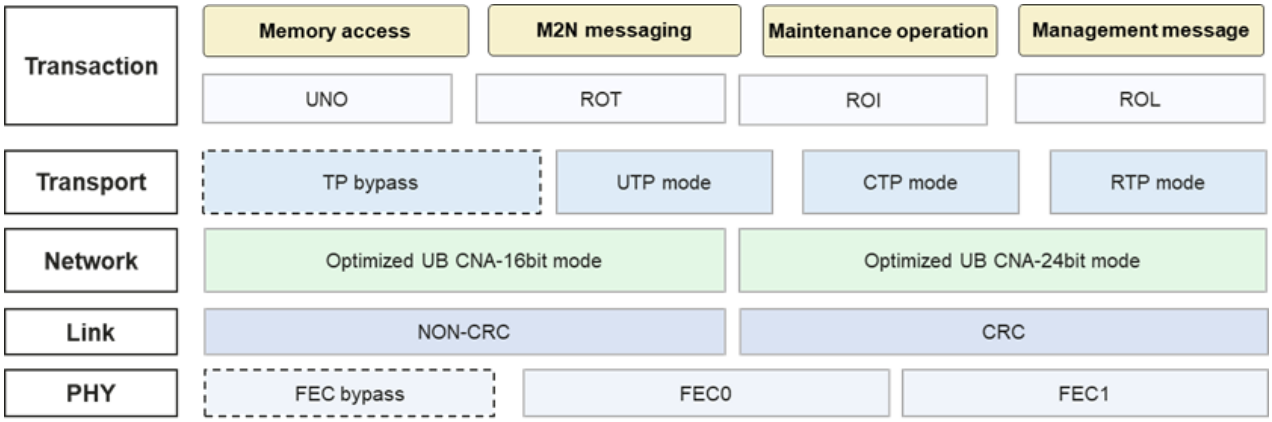
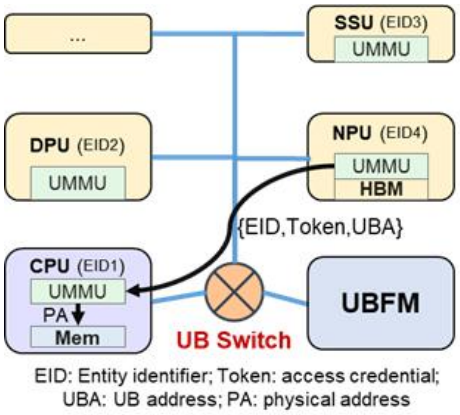


Figure 4-5 UB protocol configurations

4.2.3 Peer-to-Peer Coordination

UB supports peer-to-peer coordination across all components within a SuperPoD. Entities and UMMUs implement unified addressing and access control, enabling peer-to-peer interconnect and access of computing resources. Each UB Entity has a unique Entity identifier (EID) for global communication. UMMUs authenticate memory access and translate addresses to enable CPU-bypassed access across devices. The UBFM manages logical Entities, and oversees the interconnect and resource configurations such as device EIDs, network addresses, and routes within the Fabric.



Communication Range	H2H	H2D	H2N	H2S	D2D	D2N	D2S
Within a node	√	√	√	√	√	√	√
Across nodes in one cabinet	√	√	√	√	√	√	√
Within a SuperPoD	√	√	√	√	√	√	√
Within a cluster	√	√	√	√	√	√	√

H: CPU; D: NPU; N: DPU; S: SSU

Figure 4-6 Peer-to-peer interconnect and access

4.2.4 All-Resource Pooling

UB defines new native system management permissions and messages for in-band resource management of SuperPoDs. It uses UB partitions to support multiple users and reduce virtualization overhead. UBFM assigns multiple UB Entities of a UBPU to different users, and groups UB Entities belonging to the same user into one UB partition. This ensures user isolation based on UB partition IDs (UPIs), as shown in the following figure.

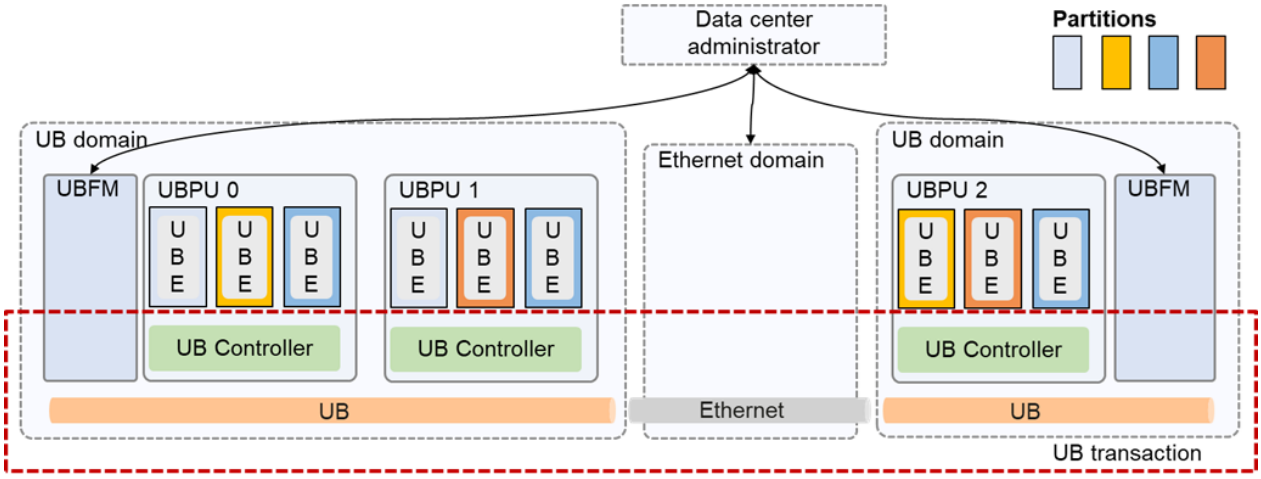


Figure 4-7 Unified resource management

UB supports Entity resource sharing, allowing direct resource access and calling through memory-mapped I/O (MMIO) or messages. For example, once the UBPU authorizes Entity 1 for Entity 0's resources, both the UBPU and Entity 1 can directly call the functions or services declared by Entity 0 through MMIO or messages.

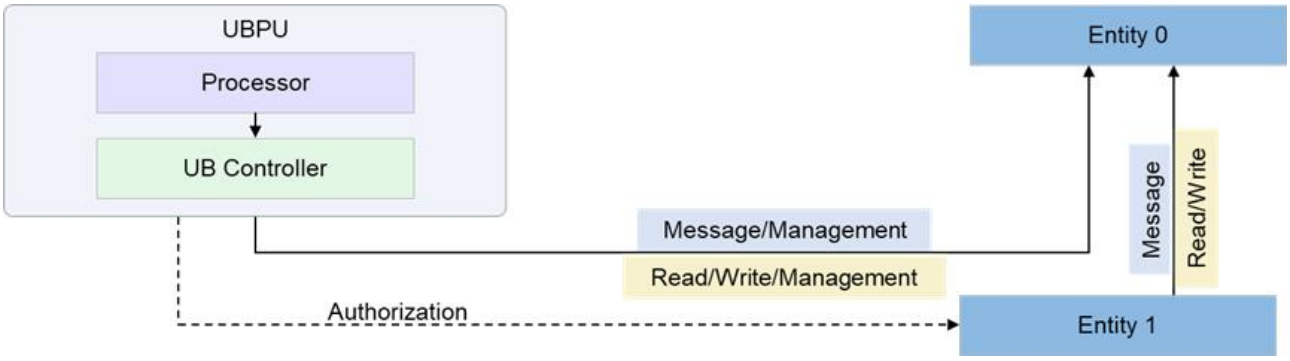


Figure 4-8 Efficient resource calling

In addition to compute and memory resources, UB pools and shares TB/s-level interconnect resources through multiple channels, enabling all Entities to use any reachable path and port.

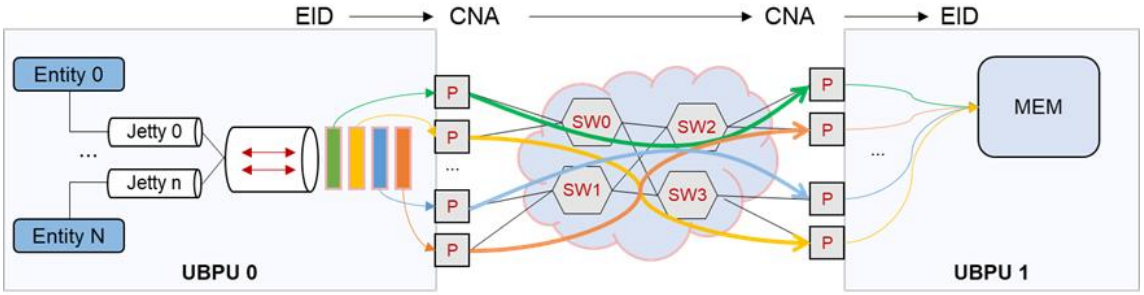


Figure 4-9 Interconnect resource pooling

4.2.5 Large-Scale Networking

Each UBPU has built-in UB Switches, which forward UB packets to directly connected UBPUs without software participation. In addition, UB leverages designs such as virtual lanes at the data link layer, per-packet/per-flow multipath routing at the network layer, and shared channels at the transport layer to enable UB packet detour. This solves problems like deadlock and reduced bandwidth utilization on direct paths. UB offers UB-Mesh (a hierarchically localized nD full-mesh datacenter network architecture, details at https://arxiv.org/abs/2503.20377) and optical switching networking technology for the large-scale, cost-effective deployment of SuperPoDs.

The nD full-mesh topology of UB-Mesh capitalizes on local organization of service data by prioritizing short, direct paths, as illustrated below. This minimizes data movement distances and use of switches, ensuring both high performance and cost efficiency.

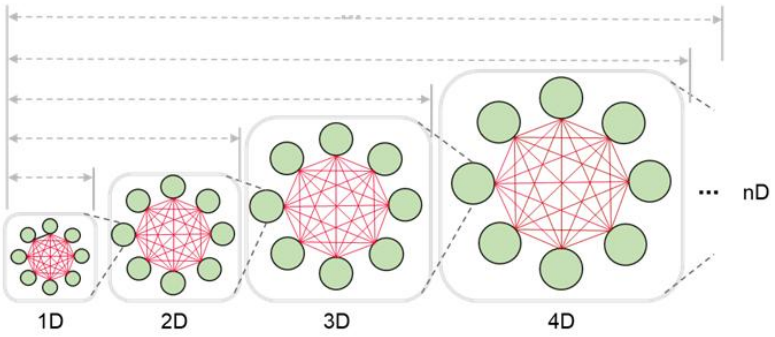


Figure 4-10 nD full-mesh topology

2D full-mesh is an implementation of nD full-mesh. It minimizes switch-related latency overhead, making it suitable for situations that require ultra-low latency, such as memory sharing and borrowing.

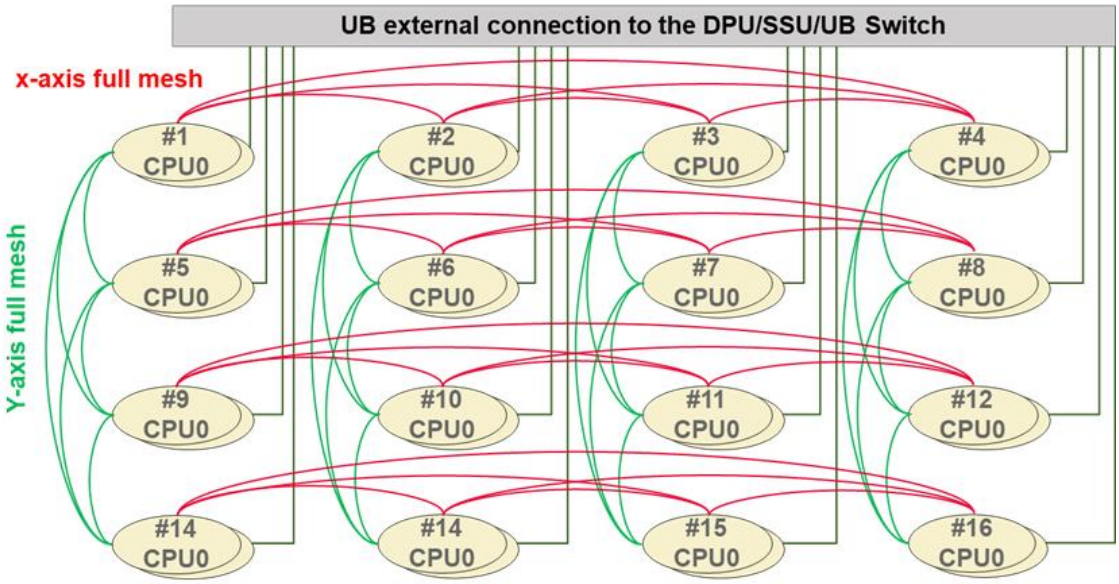


Figure 4-11 2D full-mesh direct connection topology

UB-Mesh also supports hybrid topologies. For example, it uses the 1D/2D full-mesh topology within a rack to deliver high local bandwidth through full-cable interconnect. Between racks, it employs the Clos topology with a switching layer to reduce or prevent bandwidth convergence. This topology is applicable to training and inference scenarios. As shown in the following figure, the network uses the 2D full-mesh topology within each rack and connects different racks using a layer of UB Switches. It supports linear scaling from 64 NPUs to 8,192 NPUs.

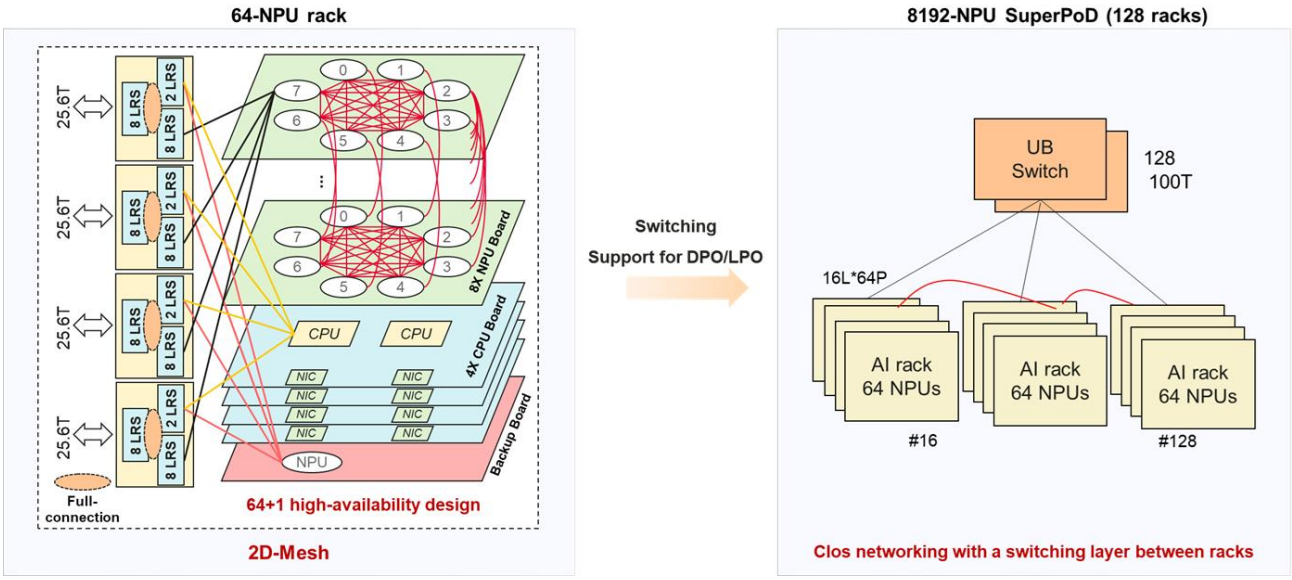


Figure 4-12 2D full-mesh + Clos hybrid topology

UB not only employs UB Switches at various levels, but also interconnects with Ethernet switches through UBoE to further expand the network. Moreover, it supports variable topologies based on OCS networking to adapt to dynamic service traffic.

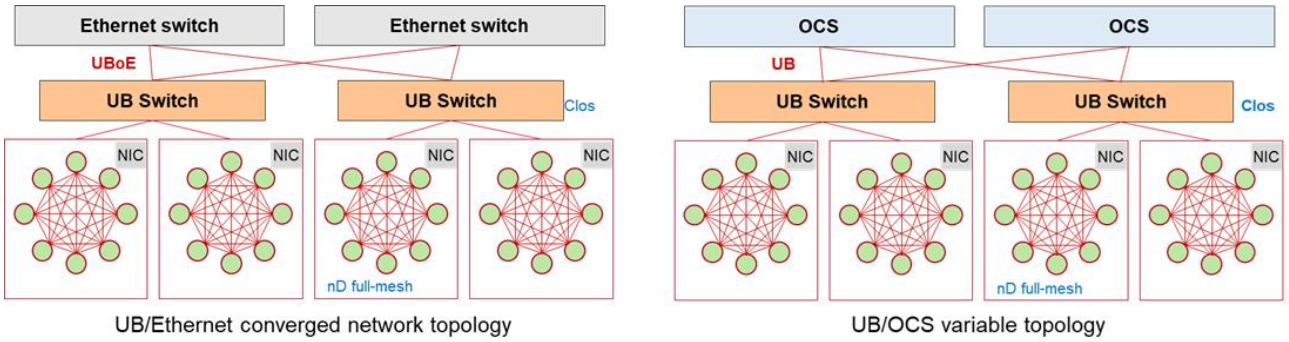


Figure 4-13 Converged networking and optical switching

4.2.6 High Availability

SuperPoDs use several times more optical modules to offer ultra-high bandwidth interconnect, which introduces significant reliability challenges. UB leverages hierarchical reliability coordination to detect and handle faults within microseconds without affecting applications. The data link layer provides the link layer retry (LLR) technology to handle bit errors or intermittent disconnections on optical links. For single-lane faults, the lane decrease technology at the physical layer is used with LLR to ensure zero packet loss during point-to-point transfers. In the event of a single optical module fault, the 2+2 optical module backup implements service-unaware fault recovery. All these technologies guarantee reliable transmission of memory semantics. UB also supports network multipath switching, reliable transport layer, and end-to-end retry.

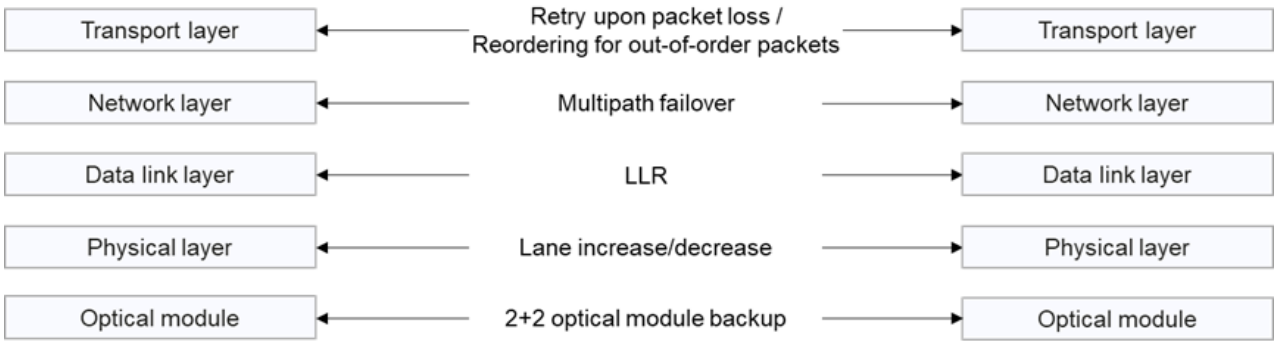


Figure 4-14 Hierarchical reliability coordination

- **LLR**

  The UB data link layer provides a retry buffer. If the remote data link layer detects packet loss, it sends a retry request to the source end. This technology addresses packet loss caused by intermittent link disconnections and bit errors.

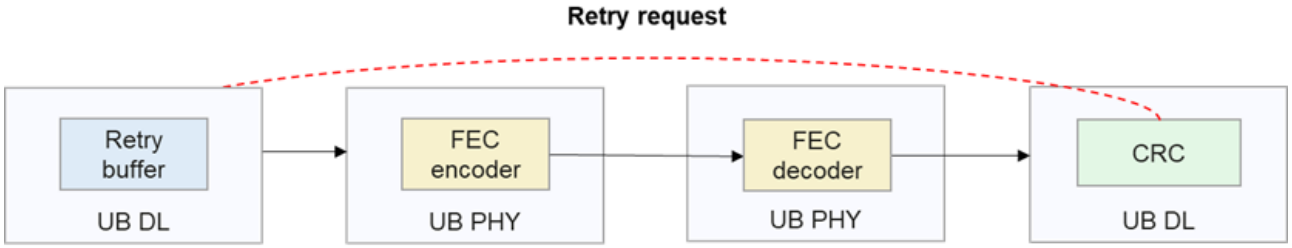


Figure 4-15 LLR

- **Dynamic lane adjustment and optical module backup**

  The UB physical layer supports dynamic lane increase/decrease, which works with LLR to implement 2+2 optical module backup, as shown in the following figure.

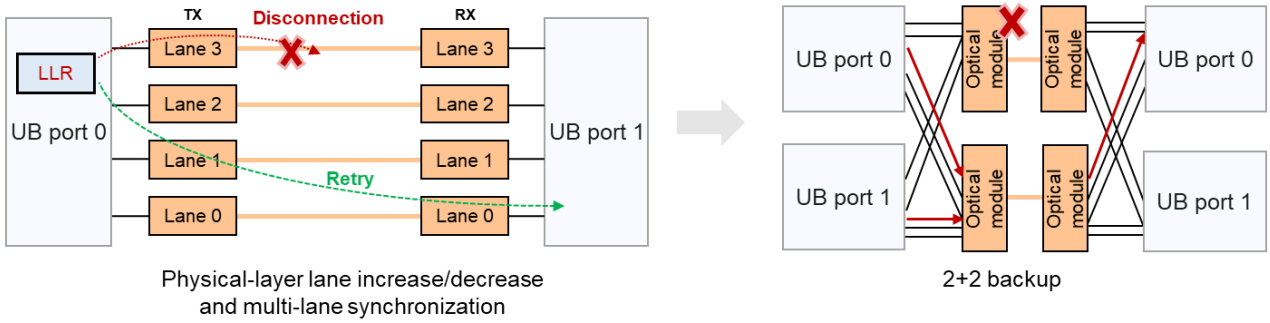


Figure 4-16 2+2 optical module backup

4.3 UB Software

Users only need to deploy the UB OS Component to support UB in the OS. The UB OS Component abstracts and decouples heterogeneous hardware, manages multi-level heterogeneous resource pools, schedules computing, memory, and interconnect resources, and establishes a unified memory address space. These designs help SuperPoDs achieve pooled memory access, global resource scheduling, dynamic combination and scaling of computing resources, and high-performance communication across devices.

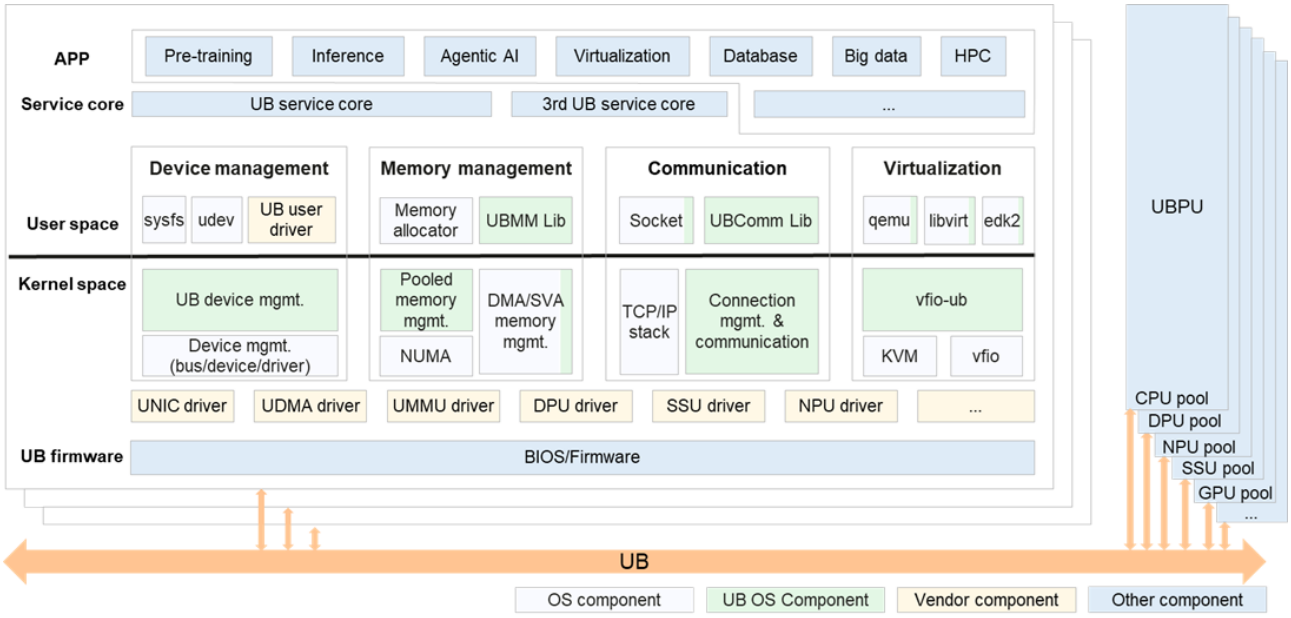


Figure 4-17 Software architecture of the UB OS Component

The UB OS Component provides the following functions:

- **Device management:** Manages UB interconnects and devices, and enables hot swapping and configuration of UB devices on compute nodes.
- **Memory management:** Offers memory semantics access within SuperPoDs to implement memory borrowing and sharing across compute nodes.
- **Communication:** Provides communication and remote calling across devices and compute nodes.
- **Virtualization:** Enables UB device passthrough to VMs.

5 Summary

UB has been adopted and verified by products such as Atlas 900 A3 SuperPoD, proving its value in driving AI and industry development. Next-generation UB-powered products will soon launch. To promote interconnect technology and industry development, Huawei has decided to open the UB specifications to the industry. Huawei invites customers and partners to build cutting-edge computing infrastructure using UB and the SuperPoD reference architecture, and work together to push the computing industry forward.

Find more details about UB in the *UnifiedBus™ (UB) Base Specification*, *UnifiedBus™ (UB) Firmware Specification*, and *UnifiedBus™ (UB) Software Reference Design for Operating Systems*, which are available at the UB official website: www.unifiedbus.com.