OmniBUS Fabric: Turning PCIe Into a Composable Resource Layer

Executive Summary

Modern rack-scale systems are increasingly built from PCIe-native components: CPUs, NVMe storage, GPUs, FPGAs, and other accelerators all depend on PCIe as a primary interconnect. The challenge is no longer simply connecting those devices, but allocating them in a way that matches application needs without adding unnecessary protocol translation, latency, power, or operational complexity.

OmniBUS® Fabric addresses this by using PCIe as the system fabric and by presenting resources as logical constructs rather than only as physical endpoints. In storage, that means a server can be given a logical volume instead of a raw drive list; in other cases, the same model can apply to memory capacity, GPU resources, or other PCIe-attached assets. The result is a fabric that behaves more like composable infrastructure than conventional device attachment.

Compared with Ethernet-based architectures, OmniBUS® can reduce the need for bridging, protocol conversion, and software workarounds. It also preserves native PCIe semantics, allowing standard enumeration, familiar drivers where appropriate, and direct hardware-level data movement for workloads that benefit from it.

The Problem With Conventional Fabrics

Data center racks often rely on multiple fabrics at once: Ethernet for general communication, Fibre Channel for storage, and InfiniBand for high-performance clustered traffic. That creates redundant infrastructure, adds power draw, and increases system complexity because each fabric must be designed, managed, and troubleshot separately.

A conventional PCIe switch fabric solves only part of the problem. It connects endpoints, but it does not by itself turn those endpoints into higher-level logical resources. In practical terms, a standard fabric can expose devices attached to the bus, but it does not inherently provide the kind of resource abstraction needed for composable storage, shared memory, or flexible accelerator assignment.

This distinction is especially important in storage systems. If a server sees a shelf of physical drives directly, the storage layer is still largely behaving like attachment hardware. If the fabric can instead present logical volumes, the system can allocate, isolate, and reassign storage in a way that is much closer to a composable infrastructure model.

PCIe Background

PCIe, or Peripheral Component Interconnect Express, is the high-speed interconnect used to connect CPUs, storage devices, accelerators, and other endpoints inside modern systems. It uses point-to-point links organized into lanes, so aggregate bandwidth increases with both lane count and PCIe generation. That makes PCIe especially important when the devices in a rack are already PCIe-native and do not need a network protocol conversion layer.

For this reason, PCIe v5 is a meaningful design point for rack-scale systems. It offers much higher bandwidth than earlier generations and supports architectures that need low latency and high data movement efficiency. When a fabric can preserve PCIe semantics end-to-end, it can often deliver the same devices and drivers with less overhead than a translated Ethernet-based design.

OmniBUS Architecture

OmniBUS® Fabric uses PCIe as the underlying transport and extends it with a software-defined control plane. The fabric can initialize routing, manage errors, handle hot-plug events, and synthesize PCIe hierarchies so that hosts see normal PCIe enumeration while only being exposed to the resources assigned to them. This approach allows multiple hosts to reside on the same PCIe-based fabric without requiring the hosts to understand the fabric's internal topology.

This architecture is important because it separates connectivity from presentation. Conventional PCIe switching connects devices. OmniBUS® goes further by deciding what a host is allowed to see and by presenting that host with a PCIe-compliant view of assigned resources. That is what makes the platform suitable for composable infrastructure rather than simple device interconnect.

The system also supports direct hardware pathways for performance-sensitive operations such as sharing and DMA. For host-to-host communication, OmniBUS® can present an Ethernet-like DMA model for application compatibility and a lower-latency path for short messages where performance is critical.

Storage As Composable Infrastructure

Storage is where the distinction between attachment and composability becomes easiest to explain. In a traditional PCIe-attached storage shelf, the host typically sees the physical drives or device functions that are directly attached to the bus. That is useful, but it still leaves volume management, partitioning, isolation, and assignment to higher layers of software or to separate storage protocols.

OmniBUS® changes that model by allowing a storage subsystem to present logical volumes over PCIe. The server enumerates the assigned resource as part of its local infrastructure, even though the capacity may physically reside elsewhere in the system. That is what makes the architecture feel like composable storage rather than a collection of raw drives.

A simple way to describe this is:

1. A storage subsystem creates a logical volume.
2. OmniBUS® presents that volume to a server over PCIe.
3. The server enumerates the volume as a PCIe-visible resource.
4. The operating system treats it as locally assigned infrastructure rather than as a network-attached block device.

That model can be extended beyond storage. The same composable principle can apply to a volume of RAM, a GPU resource, or another PCIe-native device class, provided the fabric and platform software present it through the appropriate PCIe-visible abstraction.

SR-IOV Explained

SR-IOV, or Single Root I/O Virtualization, is a PCIe extension that allows one physical device to expose a Physical Function, or PF, and multiple Virtual Functions, or VFs. The PF is the management and control interface, while the VFs are lighter-weight PCIe functions that can be assigned to virtual machines or isolated workloads. This reduces virtualization overhead because the workload can access hardware resources more directly.

In a standard environment, SR-IOV is usually used within a single host. A single physical adapter exposes multiple VFs, and the hypervisor or operating system assigns them to guest workloads. OmniBUS® extends that concept by allowing SR-IOV-capable endpoints to participate in a larger fabric where multiple hosts can share resources while still using standard drivers and standard PCIe enumeration semantics.

That makes SR-IOV an important supporting concept in the paper, because it helps readers understand how hardware resources can be partitioned without treating everything as a monolithic physical device. In the OmniBUS® model, SR-IOV is part of the broader story of resource sharing, isolation, and composability.

Host-to-Host Communication

Many data center applications already assume Ethernet-style communication patterns, so OmniBUS® provides a virtual Ethernet NIC model for compatibility. This lets existing software run without major changes while still taking advantage of the PCIe fabric underneath. That is useful for general-purpose workloads where compatibility matters more than squeezing out every last microsecond.

For clustering and other latency-sensitive applications, OmniBUS® also supports a dedicated NIC DMA path. This reduces software copying overhead and enables very fast host-to-host transfer without specialized application changes. For short messages, Tunneled Window Connection, or TWC, provides an even lower-latency option. Together, these mechanisms give the system flexibility across a wide range of workloads.

Error Handling And Reliability

Storage and composable infrastructure require robust failure handling. If a device disappears unexpectedly, the fabric must avoid cascading errors that disrupt the rest of the system. OmniBUS® supports Downstream Port Containment, or DPC/eDPC, which helps isolate faults, disable problematic links, and make recovery more manageable in storage and hot-plug scenarios.

That matters because storage systems cannot assume that every failure is graceful. If a drive is removed, if a cable is pulled, or if an endpoint fails, the fabric must contain the impact and preserve system integrity as much as possible. OmniBUS® adds logic to track outstanding reads and synthesize completions so the host is not left waiting on a timeout when an endpoint disappears.

Topology And Clocking

Traditional PCIe systems are typically constrained by hierarchy, with a relatively fixed path from one endpoint to another. OmniBUS® relaxes that restriction and can support alternative topologies

such as mesh and fat-tree styles while keeping PCIe compatibility at the architectural level. This gives system designers more freedom in how they allocate resources across a rack or enclosure.

The platform also supports multi-clock-domain operation and spread spectrum clocking isolation, which helps with backplane and system integration. For designers, this reduces the need to force every component into a single rigid clocking model and makes the overall system easier to adapt to different hardware sources and board layouts.

Legacy Workload Impact

SP7K storage platform. Please see the white paper on the SP7K for more in-depth detail about the GateStor Storage platform.

The SP7K platform makes the OmniBUS story especially relevant for legacy applications that need high throughput without being redesigned. In this model, the storage subsystem parallelizes access across many drives, typically 10 to 12 or more, with striping aligned to page size and the operating system using VMP-style fetch behavior so data is accessed through a memory-like model. That combination is especially effective for workloads that need sustained read/write performance and predictable access patterns.

This matters for applications like video surveillance, large relational databases, and other high-performance legacy workloads. These systems often depend on speed, consistency, and direct access more than on modern cloud-native abstractions. SP7K provides a way to accelerate them while preserving the operational model they already understand.

A concise way to describe this is:

“SP7K is well suited to legacy applications that require high-throughput, page-aligned, parallel access. By striping data across many drives and presenting the data through a memory-style fetch model, the platform can accelerate workloads such as video surveillance, large relational databases, and other performance-sensitive enterprise applications without requiring application redesign.”

You can also note that SP7K supports traditional storage protocols in addition to its high-performance mode. That makes it easier to fit into existing environments while still providing a path to more advanced performance behavior.

AI Workload Impact

The same OmniBUS® matrix can also be used for AI workloads, and that is where the broader composability story becomes very powerful. A RAM-based volume can be added to stage RAG data, extended memory can be attached to a GPU, and an entire database can be run in memory when the application needs extreme speed. Because the fabric is PCIe-based, these resources can be presented in a way that feels native to the host rather than translated through a separate network stack.

This gives architects a way to compose the right mix of storage, memory, and accelerators for the workload in front of them. In one configuration, the system may emphasize storage capacity and protection. In another, it may prioritize GPU memory expansion and fast ingestion of retrieval data. In both cases, the system is built around the same PCIe-native resource model.

A good paper sentence here is:

"On the same OmniBUS® fabric, the system can present RAM-backed volumes for RAG data, extend memory resources to GPUs, or host an in-memory database, enabling AI workloads to scale dynamically within a PCIe-native composable infrastructure."

True Mirroring And High Availability

OmniBUS® also enables a true mirrored storage model within SP7K, similar to RAID 1. Both members of the mirror are active parts of the same local storage construct, so the host sees a single coherent volume and I/O is issued to both members simultaneously. If one member fails, I/O continues on the surviving member without interrupting the application.

This is an important distinction because it is not a passive backup or a secondary replica. It is a live mirror that behaves like one coherent protected volume from the host's point of view. In practical terms, that means the system can continue operating even if one member fails, while still presenting a local-device model to the application.

Because OmniBUS® extends PCIe across the fabric, the physical location of the mirrored members is not what matters to the host. SP7K (1) can see and use a mirrored volume in a way that appears local, even if the mirror member is physically distant or housed in another rack, because the fabric preserves the local infrastructure view. That is the magic of PCIe and OmniBUS®: the host experiences a coherent local storage service while the fabric handles composition underneath.

Benefits By Workload

OmniBUS and SP7K together create a platform that serves both legacy and modern AI workloads. For legacy applications, SP7K's parallelized storage architecture can accelerate video surveillance, large relational databases, and other throughput-intensive systems without requiring redesign. For AI workloads, the same PCIe-native fabric can present RAM-backed volumes for RAG data, extend memory resources to GPUs, or host an in-memory database.

In combination, SP7K provides the storage intelligence and performance model, while OmniBUS® provides the fabric that makes those resources composable across the rack. That combination is what makes the story unique. It is not just faster storage, and it is not just a better switch. It is a system architecture that treats storage, memory, and accelerators as allocatable PCIe resources.

Closing Position

OmniBUS® Fabric is not simply another PCIe switch fabric. It uses PCIe as the foundation for a more expressive architecture that can present logical resources, support multi-host sharing, and preserve compatibility with standard PCIe enumeration and common software models. Where conventional fabrics mainly connect physical devices, OmniBUS is designed to make those devices behave like assigned infrastructure.

The addition of SP7K strengthens that story significantly. SP7K gives the platform a practical, high-performance storage model for legacy applications, while OmniBUS® extends that same infrastructure into AI-era composability across storage, memory, and accelerator resources. Together they create a platform story that is broader and more differentiated than either technology alone.