Technical Brief: Why Traditional LAN Architectures Are a Mismatch for Modern Real-Time Systems — and Long-Reach PCIe Solves the Problem

Overview

The architecture of modern computing systems is undergoing a fundamental transition. Historically, distributed computing environments were designed around **node-centric architectures**, where independent servers communicated through Ethernet networks. These architectures worked well for batch processing workloads where throughput mattered more than timing precision.

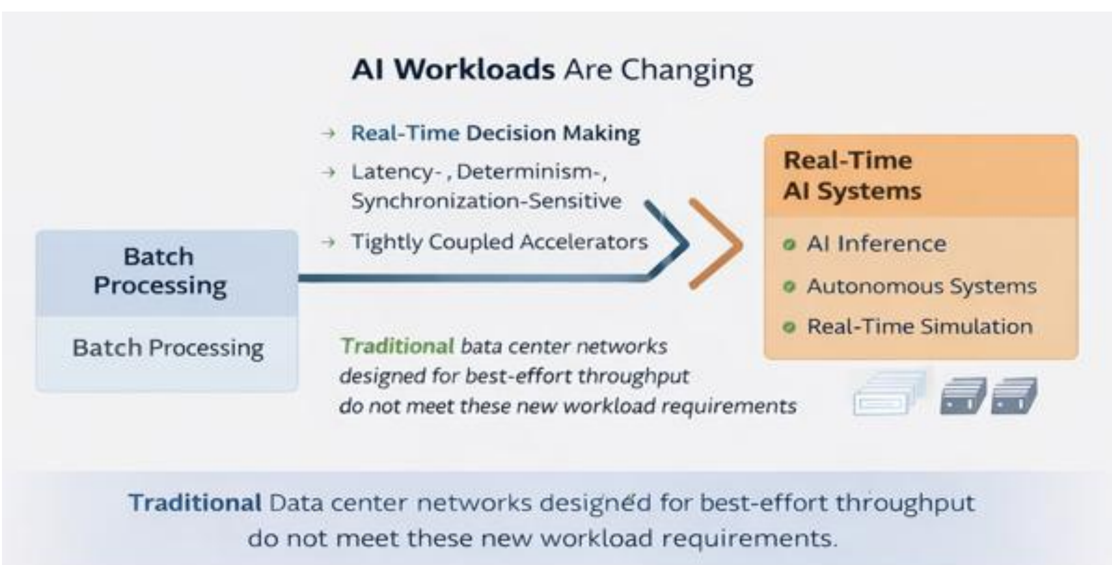
Today's emerging applications—particularly **AI inference at the edge, real-time simulation, robotics, autonomous systems, and distributed sensor processing—have very different requirements**. These systems require tightly synchronized compute resources that must exchange data with **predictable latency, deterministic timing, and direct access to shared memory resources**.

Traditional LAN-based architectures were never designed for these constraints.

A new approach is emerging: **long-reach PCI Express (PCIe) fabrics**, which extend the native interconnect used inside computers across multiple systems. This approach allows distributed compute resources to behave like a single coherent machine, enabling deterministic performance at scale.

The Shift in AI and Real-Time Workloads

Modern compute workloads are evolving rapidly, particularly with the growth of Real-Time and autonomous systems. Historically, workloads were designed around large centralized systems performing batch processing tasks. Data was collected, processed asynchronously, and results were generated later.



Today, many systems must make decisions in real time.

AI workloads have evolved from:

- Batch processing → Real-time, distributed decision-making
- Throughput-first processing → Latency-, determinism-, and synchronization-sensitive execution
- Isolated server nodes → Tightly coupled multi-accelerator systems

These new workloads increasingly rely on combinations of CPUs, GPUs, FPGAs, and specialized accelerators that must exchange data continuously while operating in tight synchronization. Examples include:

- Real-time AI inference systems
- Autonomous vehicles and robotics
- Defense and aerospace sensor fusion
- High-performance simulation environments
- Financial trading platforms
- Distributed AI training clusters

In these environments, **predictable timing matters as much as raw throughput**.

Why Traditional LAN Architectures Fall Short

Most distributed systems today rely on Ethernet networks and the TCP/IP protocol stack for communication between compute nodes. While Ethernet is ubiquitous and well understood, it was designed as a **best-effort packet delivery system**, not as a deterministic compute fabric. This design introduces several limitations for real-time and tightly coupled applications.

1. Best-Effort Networking

Ethernet networks are optimized for maximizing throughput across many independent users. However, real-time systems require:

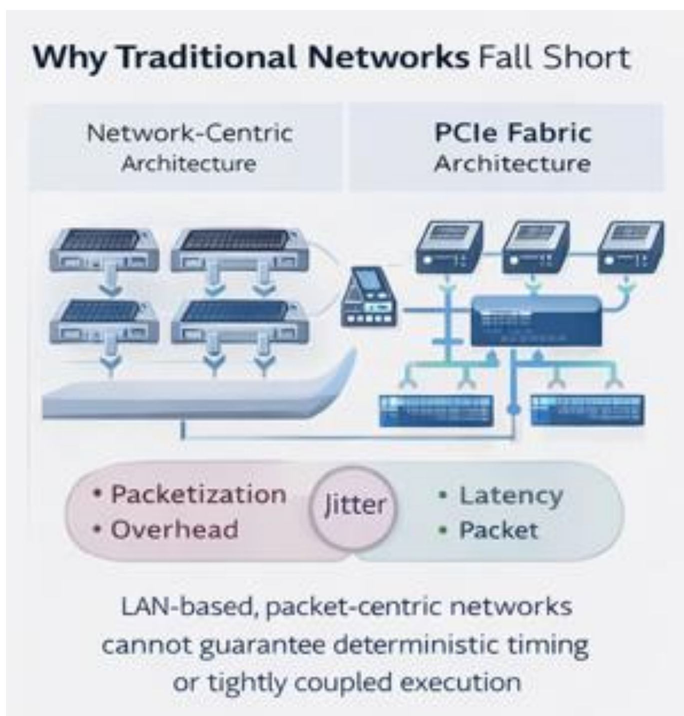
- Deterministic latency
- Guaranteed delivery timing
- Predictable system behavior under load

Traditional networks cannot guarantee these properties.

2. Packetization Overhead

Ethernet communication requires data to be broken into packets, transmitted, routed through switches, and reassembled at the destination. This introduces:

- Latency
- Jitter (timing variation)
- Buffering delays
- Software processing overhead



Even high-speed networks such as **100G or 400G Ethernet** cannot eliminate these architectural inefficiencies.

3. No Shared Memory Model
Traditional network architectures treat each node as an independent system. This means:

- No shared memory space
- No direct device-to-device communication
- All communication must be explicitly packaged and transmitted through network stacks

This dramatically increases complexity and limits system performance.


PCIe on Motherboard

4. Performance Degrades Under Load
As systems scale, network congestion and switching delays introduce unpredictable behavior. This results in:

- Increased latency
- Greater jitter
- Reduced synchronization across compute nodes

For real-time systems, these effects can make architectures **unusable at scale**.

Scaling Network Bandwidth Does Not Solve the Problem

A common approach to improving performance is simply increasing network bandwidth. Over the past decade, Ethernet speeds have progressed rapidly: 25G → 100G → 200G → 400G → 800G
While higher bandwidth improves throughput, it does **not address the fundamental architectural limitations** of packet-based networking. Key issues remain:

- Latency remains relatively high
- Packet overhead remains unavoidable
- Synchronization challenges remain
- Deterministic performance is still not guaranteed

As a result, increasing network speed often leads to **higher cost, greater power consumption, and increased system complexity without solving the underlying architectural mismatch.**

PCI Express: The Native Interconnect of Modern Compute

Inside every modern computer system, high-performance devices communicate through **PCI Express (PCIe)**.
PCIe is the industry-standard interconnect used to connect:

- CPUs
- GPUs
- FPGAs
- AI accelerators
- High-speed storage
- Network interfaces
- Specialized hardware devices

Unlike Ethernet, PCIe was designed specifically for **direct device-to-device communication with deterministic performance**. Key characteristics include:

- Extremely low latency
- Direct memory access (DMA)
- Peer-to-peer communication
- High bandwidth density
- Deterministic performance behavior

PCIe is governed by the PCI-SIG industry consortium, ensuring interoperability across thousands of vendors and products.

Why PCIe Matters Now

Several major trends are driving increased adoption of PCIe-based architectures.

1. Predictable Performance Scaling

PCIe has one of the most predictable performance roadmaps in the industry. Each generation roughly doubles bandwidth while maintaining backward compatibility. Example: PCIe Gen 4 → Gen 5 → Gen 6 → Gen 7. This ensures long-term infrastructure investment protection.

			Performance demanded today					
PCIe Generation	**Gen 1**	**Gen 2**	**Gen 3**	**Gen 4**	**Gen 5**	**Gen 6**	**Gen 7**	**Gen 8**
Specification Released	2003	2007	2010	2017	2019	2022	2025	2028
Total Bandwidth (x16)	4.0 GB/s	8.0 GB/s	16.0 GB/s	32.0 GB/s	64.0 GB/s	128.0 GB/s	256.0 GB/s	512.0 GB/s

2. Accelerator-Driven Computing

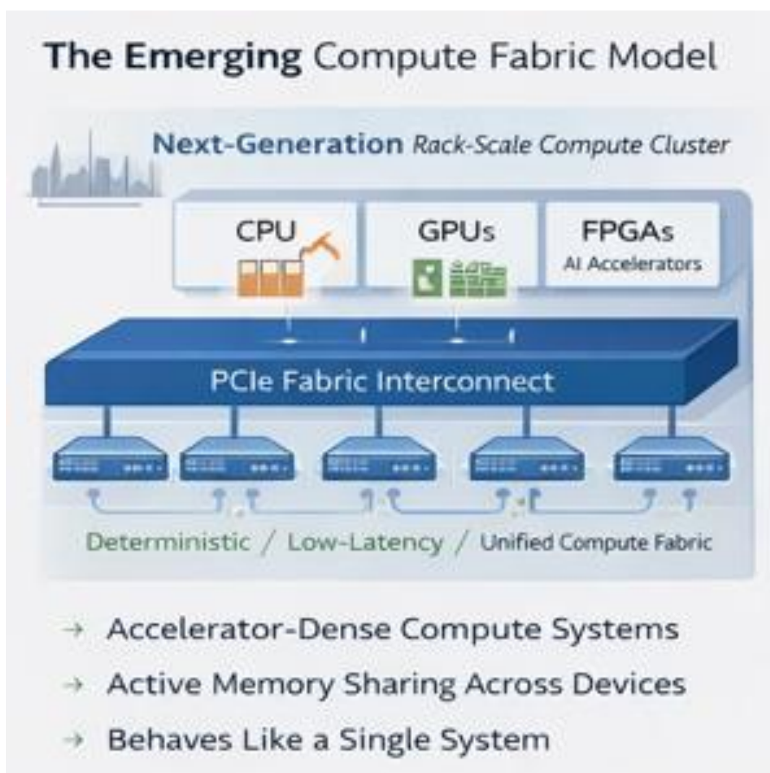
AI systems increasingly rely on multiple specialized accelerators. These devices must exchange large volumes of data quickly and deterministically. PCIe provides the **native interface used by these accelerators**, making it the natural backbone for next-generation architectures.

3. Increasing System Density

Modern compute systems are packing more GPUs and accelerators into smaller spaces. As density increases, traditional networking architectures become inefficient. PCIe fabrics enable **high-density compute clusters with minimal communication overhead.**

Introducing Long-Reach PCIe Fabrics

Traditionally, PCIe has been limited to connections within a single server motherboard.
Recent advances in switching, signaling, and system architecture now allow PCIe to be extended beyond a single system. This approach is known as **long-reach PCIe**.



Long-reach PCIe fabrics extend the native PCIe interconnect across multiple systems while preserving PCIe's core characteristics. This enables distributed compute resources to function as a **single coherent compute environment**. Key capabilities include:

- Extending PCIe beyond the server chassis
- Maintaining native PCIe semantics across distance
- Enabling direct device-to-device communication
- Allowing shared memory access across systems

The result is a distributed infrastructure that behaves much like **a single large computer rather than a collection of loosely connected servers.**

Architectural Benefits

Long-reach PCIe architectures offer several important advantages over traditional LAN-based systems.

Deterministic Performance

Communication occurs through direct memory access rather than packet routing, eliminating jitter and unpredictable network delays.

Ultra-Low Latency

PCIe communication occurs at nanosecond-scale latency, significantly faster than Ethernet-based communication.

Simplified System Architecture

By removing networking layers, PCIe fabrics reduce system complexity and improve operational efficiency.

Higher Accelerator Utilization

Direct communication between GPUs, FPGAs, and other accelerators allows workloads to be distributed more efficiently across available hardware.

Lower Power and Hardware Overhead

Reducing reliance on NICs, switches, and networking infrastructure can significantly lower system power consumption and infrastructure cost.

The Emerging Compute Architecture

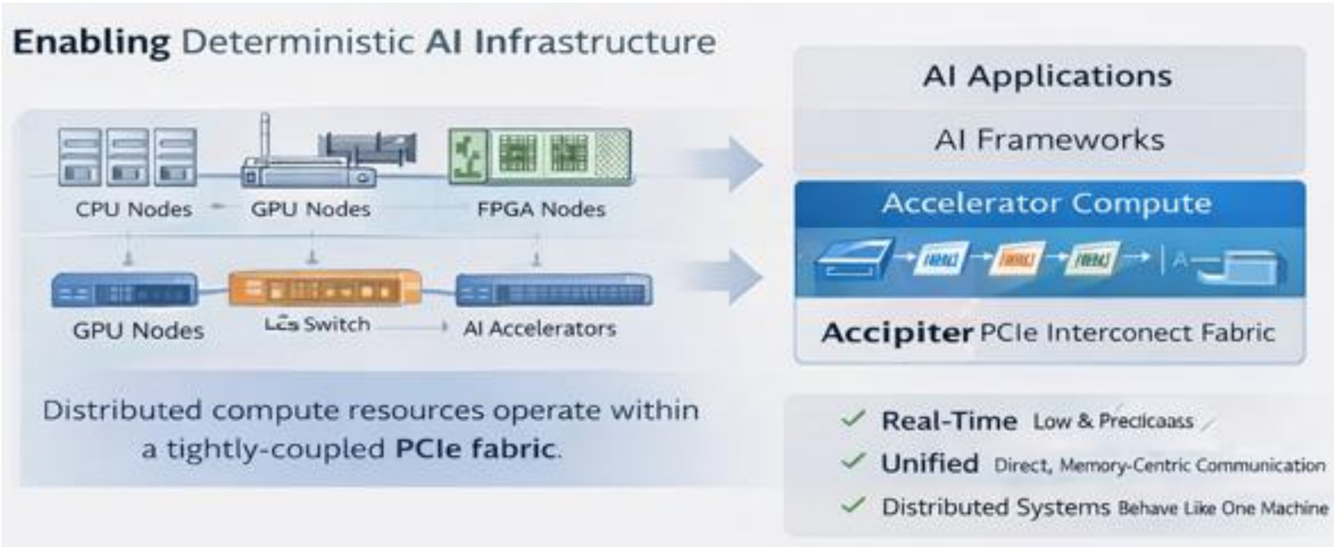
A hybrid architecture is increasingly common in advanced computing environments: **Inside the rack or compute cluster.**

Long-reach PCIe fabrics connect compute nodes, accelerators, and memory into a unified fabric.
Outside the rack

Ethernet remains useful for:

- Data center networking where batch processing offers acceptable performance
- Remote management
- External connectivity
- Communication between independent clusters

This hybrid approach combines the strengths of both technologies.



Conclusion

The evolution of AI, real-time analytics, and accelerator-driven computing is exposing the limitations of traditional LAN-based architectures.

Packet-based networking introduces latency, jitter, and complexity that limit performance in tightly coupled distributed systems.

PCI Express, the native interconnect of modern compute platforms, provides a fundamentally different approach—one based on **deterministic, memory-centric communication between devices.**

By extending PCIe beyond the server through long-reach fabrics, distributed computing systems can operate with the efficiency and predictability of a single coherent machine.

As AI systems continue to scale in complexity and performance requirements, **long-reach PCIe architectures are emerging as a critical infrastructure technology for next-generation computing systems.**