

Messaging Overview

Introduction

Gen-Z is a new data access technology that not only enhances memory and data storage solutions, but also provides a framework for both optimized and traditional messaging solutions. The following sections unveil the features and capabilities of the Gen-Z architecture by first describing the foundational features and building on these to illustrate the next generation components and system solutions possible.

Gen-Z Messaging

The *Gen-Z DRAM and Storage-Class Memory (SCM) Overview* and *Gen-Z Block Storage Overview* documents describe the enhanced features and capabilities of Gen-Z as a data access interface and interconnect. Figure 1 illustrates an example infrastructure solution built on using Gen-Z for fabric-attached memory data access, but using an existing network solution for messaging between a pair of servers. The Media Controller in the memory module and the two CPUs have integrated Gen-Z logic and interfaces to support low-latency, high-bandwidth, resilient, and scalable byte-addressable data access to volatile or storage class (persistent) memory. But this configuration has deployment challenges. First, messaging and data access have their own independent fabrics and components including NICs (for networking), switches, cabling, management software, and mechanical packaging. Second, the message latency between the CPUs will be higher than the data access latencies because of the additional protocol translations at the PCIe® and network interfaces. Finally, this configuration incurs higher cost, power consumption, and operational management overhead.

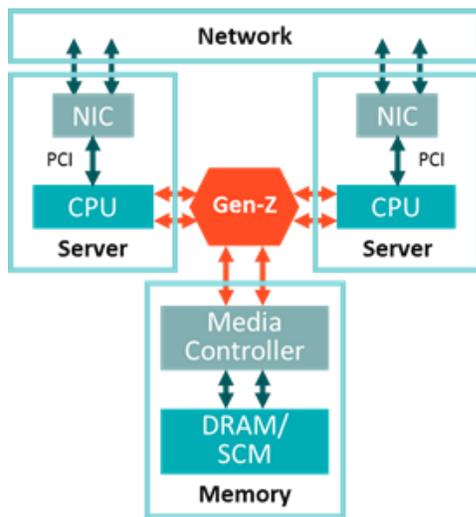


Figure 1: Gen-Z with Existing Networking

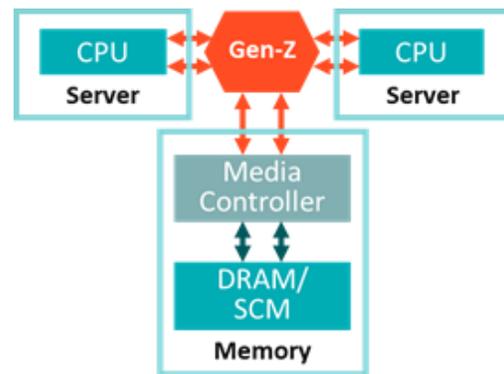


Figure 2: Gen-Z for Messaging & Data Access

Figure 2 demonstrates an alternative that consolidates messaging with data access onto Gen-Z. The Gen-Z architecture includes protocol and packet handling mechanisms to support both reliable and unreliable messaging seamlessly with byte-addressable and block data access operations. This configuration offers many benefits including:

- Sub-250 ns one-way, application-to-application latency
- Bandwidth scaling
- Simple, optimized, system designs
- Reduced power consumption

- Integrated management framework
- Reduced capital and operating costs

Gen-Z messaging can be used for efficient, memory semantic communication between a variety of Gen-Z components, including compute components (e.g. CPUs, SoCs, GPUs, FPGAs, etc.), or communication-specific components (e.g. gateways to other network technologies like Ethernet, InfiniBand, etc.). Gen-Z enables legacy protocols and/or messaging APIs to be tunneled/mapped over Gen-Z infrastructure, as well as enabling next generation workloads that have native Gen-Z library support. Gen-Z supports messaging in deployments as small as a single enclosure (rack mount server, blade enclosure, etc.), to rack-scale, and even data center scale deployments if the components implement the optional Gen-Z features that support this capability. Factors such as component/solution features, cost, complexity, power, performance (latency/bandwidth), management models, etc. all affect the particular workloads, target markets, and deployment models that can be deployed using Gen-Z technology.

Flexible, High Bandwidth Serial Interfaces

Gen-Z utilizes high-speed serial physical layer technology supporting 16, 25, 28, 56, and 112 GT/s rates on each serial signal. Gen-Z supports link interfaces that support 1-256 serial signals for transmit and receive (the most common link widths will be 1, 2, 4, 8, or 16 serial signals). Gen-Z interfaces support both electrical and optical connectivity between components. Gen-Z supports symmetric link interfaces where there are an equal number of transmit and receive serial signals, as well as asymmetric link interfaces where the number of transmit and receive serial signals are not the same. Although symmetric links are most common, asymmetric links enable solutions to tailor transmit or receive bandwidths to application-specific needs (e.g., at link initialization, configuring six receive and two transmit lanes to increase memory read and/or receive message bandwidth by 50%). Figure 3 illustrates aggregate unidirectional transmit or receive bandwidths of 4, 8, or 16 links. Each link is independently configured and can be dedicated to a specific workload or shared by multiple workloads.

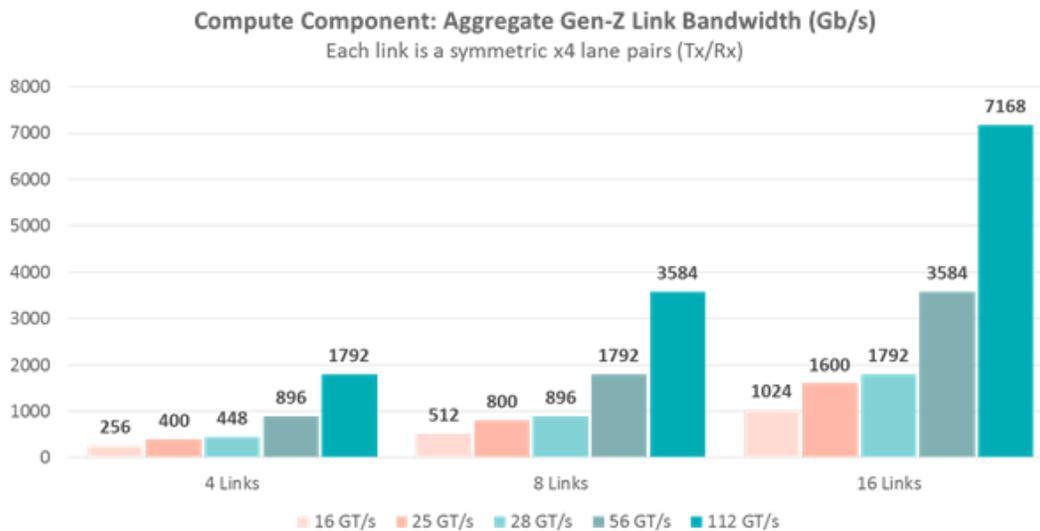


Figure 3: Examples of Gen-Z Compute Component Link Bandwidth

Enhanced Multipath for Messaging

Multi-link Gen-Z components increase aggregate bandwidth, system resiliency, and topology flexibility. This enables Gen-Z solutions to deliver optimal performance and availability without sacrificing performance. Figure 4 illustrates a pair of dual-processor systems with four Gen-Z links per processor connected to two Gen-Z switches. Each system can isolate and load-balance traffic to meet application performance needs. In the event of switch or link failure, Gen-Z transparently rebalances traffic to the remaining links and switches to minimize application performance impacts. In this example, the loss of a single link represents a 25% performance loss (the more links provisioned, the lower the performance loss due to link failure).



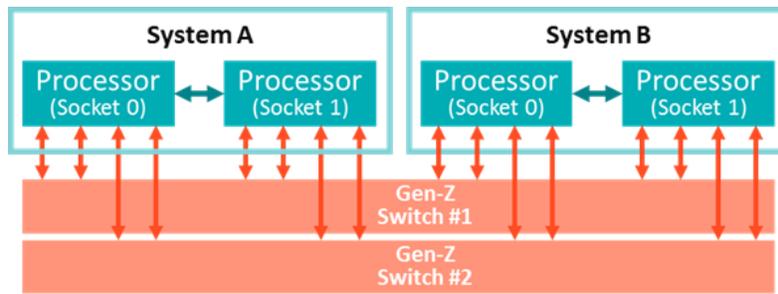


Figure 4: Gen-Z Enhanced Multipath Capabilities

Flexible Topologies and Advanced Switching Capabilities

Gen-Z architecture enables solutions to support a wide range of topologies from simple point-to-point to single-enclosure fan-out, to scale-out multi-rack solutions. Gen-Z also enables switch logic to be integrated into any type of component (e.g. CPUs, GPUs, memory, network gateways, etc.) to reduce costs, improve peer-to-peer latencies and support fabric topologies like 2D/3D Torus, Mesh, and Daisy-Chains. Using standalone switches enables more scalable topologies, such as Clos, Fat Trees, Hypercube, Dragonfly, etc.

Gen-Z components and switches (internal or standalone) incorporate novel, compact relay table configurations that enable nearly any routing algorithm to be supported without sacrificing performance. Further, Gen-Z supports adaptive routing to dynamically bypass congested or failed paths to deliver reduce jitter and ensure application availability.

To reduce message latency jitter, Gen-Z supports up to 32 virtual channels (VCs) and up to 256-byte data payloads. VCs enables Gen-Z to separate traffic to target independent buffer resources. When combined with VC remapping, packets from limited VC components can be transparently distributed across multiple switch paths to maximize fabric utilization and efficiency. A 256-byte data payload enables Gen-Z to achieve ~90% protocol efficiency while minimizing jitter due to head-lo-line blocking (a modest data payload size ensures different priority packets can be efficiently interleaved with minimal latency and complexity).

Message Models

Any processor, GPU, FPGA, DSP, etc. component can utilize Gen-Z messaging to improve solution performance, capability, and availability. Such components can support software-based or hardware-assisted messaging. Figure 5 illustrates two processors using software-based messaging by exchanging Gen-Z read, write, and atomic operations (these operations can be transparently generated by components with an integrated memory management unit used to translate load / store / atomic instructions and operations into Gen-Z read, write, interrupt, and atomic operations). Software-based messaging can be directly used by applications to deliver optimal latency and power efficiency. For example, a thread in Processor B can access another thread's data in Processor A using load-store (read-write) semantics.

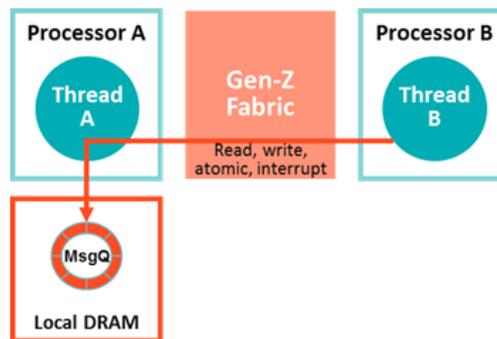


Figure 5: Gen-Z Software-based Messaging

Hardware-assisted messaging uses lightweight, component-integrated data movers to move data with minimum software involvement. Data movers can use Gen-Z write message operations to send and receive up to 64 KB of message data that targets destination-managed memory. For messaging that requires large data movement, data movers support either Gen-Z put and get operations to move up to 4 GB of data movement, or large read operations to move up to 1 MB of data, both of which require little software involvement. In all of these modes, data movers can support queue-based programming models. All of these operations and models can be used directly by applications or integrated within existing network and storage stacks or application middleware. This enables Gen-Z solutions to reduce the cost, complexity, and overheads to deliver highly-efficient and high-performance messaging solutions.

Optimized but Flexible Software Architecture

One of the most important benefits of Gen-Z is that it delivers messaging services with software that is a fraction of the size of modern protocol stacks. Gen-Z enables a new generation of applications and protocols that can directly access Gen-Z hardware interfaces to facilitate high-performance data path messaging to other components. Yet, Gen-Z software modules can be created that fit into exiting protocol stacks to enable today’s applications to maintain backward compatibility and still benefit from many of the capabilities of Gen-Z.

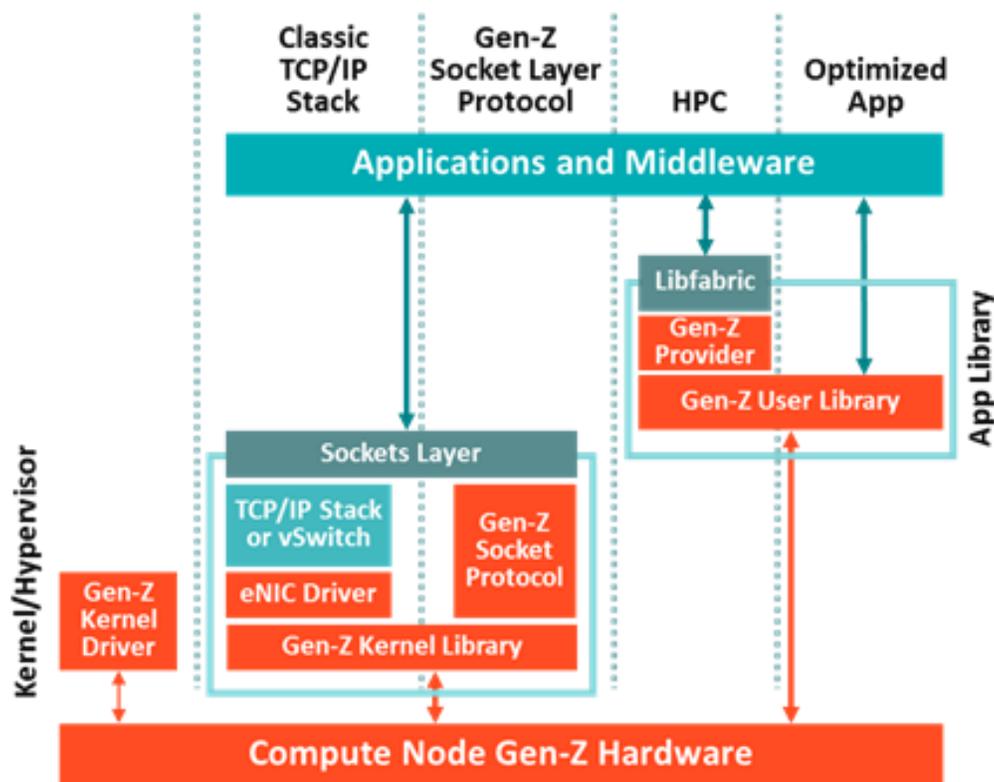


Figure 6: Gen-Z Messaging Software Architecture

Figure 6 illustrates a high-level view of the Gen-Z software architecture in a typical compute component. As indicated in the far right-hand side of Figure 6, next generation applications can be bound to a Gen-Z Library to have direct access to Gen-Z hardware for sending and receiving messages. This model has the thinnest software layer yielding high-bandwidth, low latency messaging. But the Gen-Z software model supports the creation of a Gen-Z Provider for the Open Fabric Interface (OFI) Libfabric environment to support existing High Performance Computing (HPC) middleware and workloads. Furthermore, Gen-Z supports the creation of either a native Gen-Z socket level transport protocol, or an emulated Ethernet driver that fits under the industry standard TCP/IP stack, both of which support existing applications written to the industry standard Sockets API. All of these

solutions can operate under unmodified operating systems (requiring no kernel changes) by simply installing the Gen-Z kernel drivers and libraries to support these models. Therefore, Gen-Z enables infrastructure solutions that gracefully transition to optimized Gen-Z messaging solutions while allowing existing applications to take advantage of Gen-Z performance, resiliency, and security features.

Security

Gen-Z offers a combination of hardware-enforced isolation techniques and full packet authentication to prevent errant or malicious component software or hardware from communicating with unauthorized components or accessing unauthorized resources, including Gen-Z message queue memory. In addition, for protection and network isolation, Gen-Z Fabric Management configures communications between SoCs on both a component and logical message sender and receiver basis. Gen-Z multicast mechanisms similarly restrict which components belong to various multicast groups.

Summary

Gen-Z technology is designed to supplement platforms and infrastructure with capabilities for a new generation of workloads that require messaging services, while delivering many of these capabilities even to existing applications and workloads via industry standard APIs and messaging stacks. These capabilities include:

- Next generation convergence with memory/storage data access and messaging on common infrastructure
 - Lower capital and operating costs with unified management
- High Performance
 - Very low latencies and high bandwidth
 - Robust, serial electrical/optical interface roadmap with selectable modes and link widths
- Hardware driven multipath for higher bandwidth and resiliency
 - Minimal degradation of performance in the event of failures
 - No redundant drivers or multipath software required
- Advanced switching technology facilitates messaging for enclosure and rack scale solutions
 - Reduces latency jitter and enables congestion avoidance using 32 virtual channels (VCs)
- Thin, light-weight software architecture that supports unmodified operating systems
 - Graceful transition from existing to next generation messaging APIs and protocol stacks
 - Optimized messaging for east-west traffic flows within clusters of Gen-Z infrastructure
- Security with authentication and features that protect and isolate an infrastructure's messaging resources

DISCLAIMER

This document is provided 'as is' with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Gen-Z Consortium disclaims all liability for infringement of proprietary rights, relating to use of information in this document. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Gen-Z is a trademark or registered trademark of the Gen-Z Consortium.

All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.