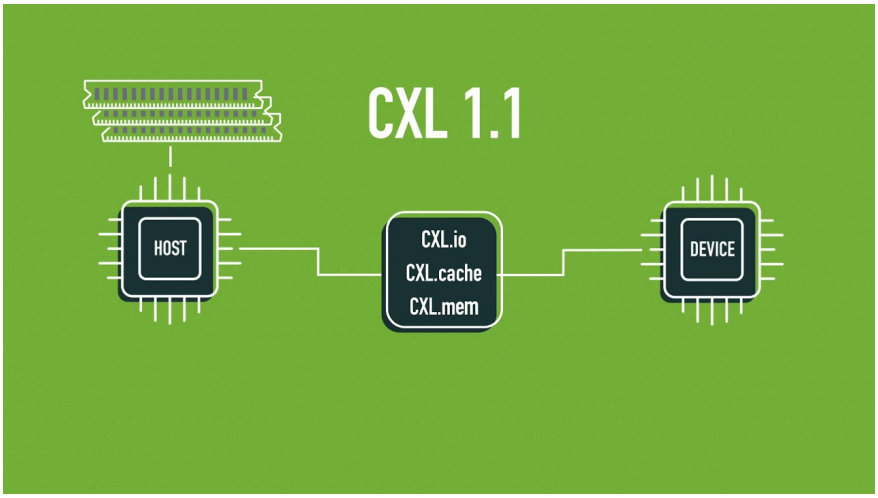
# What is CXL? But what you need to know about CXL

CXL is expected to be a coherent interconnect interface for caching to connect any number of CPUs, memories, processing accelerators (especially FPGAs and GPUs), and other peripherals.




Compute Express Link (CXL), a new open connectivity standard, promises to bring outstanding efficiency to the use of data center resources. This article ***TipsMake*** will help you better understand CXL and its potential applications in the future.

## What is CXL?

If you buy a server in the next few months that has a Sapphire Rapids generation Intel Xeon Scalable processor or a Genoa generation AMD Epyc processor, they will come with a notable new feature called Compute Express Link (CXL), an open connectivity standard that you may find useful, especially in future iterations.


What is CXL?

CXL is supported by almost every hardware vendor and builds on top of PCI Express to provide coherent memory access between the CPU and devices, such as hardware accelerators or CPUs and memory.

PCIe is designed for point-to-point communications like SSDs to memory, but CXL will eventually support one-to-many communications by passing through coherent protocols. So far, CXL is only capable of simple point-to-point communications.

CXL is currently at version 1.1, and the 2.0 and 3.0 specifications have been released. Because CXL is coupled with PCIe, new versions of CXL are dependent on new versions of PCIe. There is a two-year gap between PCIe releases, and then an even longer gap between the release of a new specification and a product hitting the market. Currently, CXL 1.1 and 2.0 devices are in what are called engineering samples for testing.

## CXL Protocols

There are three protocols that CXL supports:

1. CXL.io: Enhanced version of the PCIe 5.0 protocol for device initialization, discovery, and connection to devices.
2. CXL.cache: This protocol defines host-device interactions, allowing mounted CXL devices to efficiently cache host memory with extremely low latency using a request-and-response approach.
3. CXL.mem: This protocol provides the host processor access to the memory of the mounted device, including both volatile and persistent memory architectures.

CXL.mem is the most important protocol, starting with CXL 1.1. If a server needs more RAM, a CXL memory module in an empty PCIe 5.0 slot can provide it. There will be a slight performance drop and a slight increase in latency, but the trade-off is that it provides more memory in the server without having to buy more. Of course, you have to buy the CXL module.

CXL 2.0 supports memory pooling, using the memory of multiple systems instead of just one. Microsoft says that about 50% of all virtual machines never reach 50% of their leased memory. CXL 2.0 can find that memory and put it to use. Microsoft says that decoupling through CXL can achieve a 9-10% reduction in overall DRAM requirements.

Ultimately, CXL is expected to be a coherent interconnect interface for caching to connect any number of CPUs, memories, processing accelerators (especially FPGAs and GPUs), and other peripherals.

The CXL 3.0 specification, announced at the Flash Memory Summit (FMS) last week, takes that decoupling even further by allowing other parts of the architecture—processors, memory, networking, and other accelerators—to be pooled and dynamically addressed by multiple hosts and accelerators just like memory in 2.0.

The 3.0 spec also provides direct peer-to-peer communication over a switch or even over a switch fabric, so theoretically two GPUs could talk to each other without using the network or involving the host CPU and memory.

'It's going to be everywhere,' said Kurt Lender, co-chair of the CXL marketing working group and senior ecosystem manager at Intel. 'It's not just IT people who are embracing it. Everyone is embracing it. So this is going to be a standard feature in every new server in the next few years.'

So how will applications running in enterprise data centers benefit? Lender says that most applications won't need to change because CXL operates at the system level, but they will still benefit from CXL functionality. For example, in-memory databases can take advantage of memory pooling.

Pooling components can help provide the resources needed for AI. With CPUs, GPUs, FPGAs, and network ports all pooled together, the entire data center can function as a single system.

But let's not get ahead of ourselves just yet. We're still waiting for CXL 2.0 products, but demos at the recent FMS show suggest they're coming.