Background

【存在的问题】

1. 新兴行业和场景（人工智能、机器学习和云计算等技术）的出现催生了海量的数据。随着数据驱动型技术的兴起，对更强大计算机硬件架构的需求也随之而生。为了创造强大的处理器，在单个处理器芯片上集成越来越多的内核，以满足数据密集型应用的处理和性能需求。但是存储器的带宽和容量未能跟上 CPU 内核数量的增长步伐，使处理器和存储器的性能之间存在鸿沟。由于现有常规 DRAM 设计的局限，使存储器容量的扩展难以突破既定量级，需要全新的存储器接口技术。
2. 人工智能和大数据的兴起推动了异构计算和分解的潮流，多个不同类型的处理器能够并行处理大量数据，鉴于这样的趋势，必须针对异构计算和组合基础架构开发下一代互联技术，以实现高效的资源利用。传统计算机内部互连总线已经不能满足发展要求。比如为链接慢速设备所采用的 PCIe 协议，没有制定一致性支持的机制，不能高效的管理隔离的内存池且链路延迟过高。
3. 服务器市场市场在内存上面临的一大挑战就是成本，在选择云服务容器时，内存依然占了很大一部分，往往内存用量扩大一倍之后，价格也随之上涨了一倍，这其实与内存本身的扩展性有关，目前服务器上更大的存储已经成为常态，但是要想实现更大的内存始终存在瓶颈。
4. 现有的内存容量没有得到有效利用。Google、Facebook 和 Alibaba 的报告称，数据中心中多达 40%的服务器内存未使用。【Redy：Remote Dynamic Memory Cache [Extended Report]】其中一些内存在管理程序级别未分配，有些在应用程序/操作系统级别也未使用。在云规模上实际上是数十亿美元的闲置硬件。

Interconnect Protocols

【具体介绍几种新兴互连协议】

【PCIe 总线】

端到端：发送端和接收端种都含有 TX（发送逻辑）和 RX（接收逻辑），一个数据 Lane 中，有 2 组差分信号

树状结构：RC->HOST 主桥 在其上增加了许多功能，PCIe 总线控制器；使用 Switch 进行 PCIe 链路扩展

三层架构：事务层——接收数据请求转换成总线事务；数据链路层——保证事务层 TLP 正确传递、容错和重传机制保证数据传输的完整性和一致性以及队 PCIe 链路进行管理和监控；物理层——从逻辑层和电气层，差分信号有关的模拟电路知识。

【Gen-Z】

Gen-Z 被设计成一个内存语义架构，可以有很多不同的计算引擎挂载在它上面共享各种内存。

【NVLink】

【CCIX】

【OpenCAPI】

【CXL】

【PCIe 和 CXL】

PCIe 无法成为 CPU、GPU、FPGA 以及其他 AI 计算设备之间沟通的最佳语言。主要是因为

PCIe 不支持 cache 一致性，这会导致每次 Device 去访问 Host 上的内存的时候，即便*已经访问了多次而且内存也没有变化*的情况下，都要重新访问，这样会性能很差。
另外因为人工智能和机器学习的兴起，FPGA/GPU 卡上会有大量的内存，在不进行运算时就闲置在那里浪费资源，可是使用 PCIe 的接口的设备，Host 没法直接访问设备上的内存。设备上的内存和 Host 端的内存没法统一编址，而且同样是因为 PCIe 不支持 Cache 的一致性，Host 访问设备上的内存也会非常的慢。

【Gen-Z 和 CXL】

开始 Gen-Z 和 CXL 都属于数据中心、HPC、AI 等领域全新数据设备互连协议的领导者，分割着数据中心互连和内部连接。

- Gen-Z 最初被设计成一个内存结构，可以有很多不同的计算引擎挂在它上面，共享各种内存，而 CXL 协议设计成将处理器与它们的加速器和系统内的内存、存储连接起来；
- 而后 Gen-Z 开始负责机架间互联，CXL 负责 CPU 与 GPU、FPGA 等加速器的互连；
- Gen-Z Fabric 直连 SCM 或 CPU/FPGA 加速器模块需要在 CPU 一端提供 Gen-Z Logic 的支持，但是 80%以上的服务器市场被 Intel 占据，因为 Intel 没有加入 Gen-Z 联盟，所以 Gen-Z 的推广并不顺利。2021 年 11 月，所有 Gen-Z 规范和资产转移给 CXL 联盟。

【CCIX 和 CXL】

CCIX在x86上并不能像CXL那样真正融入CPU Memory体系，只能做成PCIe的IO Memory。
CCIX的interface latency比CXL差很多，因为CCIX无法更改被Intel严格把控的Link Layer。
CCIX尽管得到了AMD、Xilinx、ARM和Ampere Computing的支持，但是从未真正起飞，因为缺乏关键行业的支持。

【NVlink、Infinity Fabric 和 CXL】

主要是做GPU-GPU互联，GPU集群越来越多，GPU和GPU之间怎么连接就是自己的事情了。NVlink和 Infinity Fabric，没有任何行业生态系统可以围绕这些专有协议发展

【OpenCAPI 和 CXL】

OpenCAPI 推出的 OMI 重点关注内存扩展，IBM Power10 处理器边缘集成 2 个 OMI 内存单元，其思路是通过对内存接口进行序列化来增加处理器中可安装的内存量。但是只有 Power9/10 的兼容平台，CXL 可以同时支持 ARM 和 x86 等，因此有更多厂商选择 CXL。OpenCAPI 标准和财团的资产将转移到 CXL 财团。

【总结】打不过就加入

- 任何用于处理器与DRAM、FPGA以及其他专用处理器互连的后PCIe总线技术都必须得到服务器CPU供应商的支持，这是一个必要条件。CXL联盟不仅聚集了内存厂商、IP厂商、加速器厂商等，更重要的是，它有AMD、ARM、IBM以及Intel所有四个主要的CPU供应商的加入。Intel制定了新的标准，且能够利用现有的PCIe的物理层和电气层已经构建生态系统，建立了行业中大多数主要参与者都支持的行业标准协议，CXL使像异构计算的过度的成为可能.（主要是Intel占有较大的CPU是市场份额，没有解决方案会在没有Intel支持的情况下实现。目的基本上都是一致的，希望实现处理器及内存、加速器之间的快速连接，但是CXL有更好的生态支持，Intel的生态控制力。）
- CXL 较晚的推出时间反而成了它的 “利器”。随着 PCIe 摆脱了 3.0 和 4.0 代之间七年的停滞状态，并进入两年带宽翻倍的性能节奏 ，给基于 PCIe 协议的 CXL 带来了更大的优势。相比 CXL，Gen-Z 等其他协议充其量只能降级为 CPU 到 CPU 的互连，而 CXL 作为兼容的 CPU 一致性协议，将允许跨 CPU 架构到标准，可以说 CXL 在 PCIe5.0 上的性能就是为此而存在的。
- 未来的 CXL 总线版本会将 OpenCAPI 和 Gen-Z 的功能集成

CXL 1.1 2.0 3.0 协议及其特点分析

【CXL 1.1】

相对简单的主机到设备的连接标准，CXL 可在主机 CPU 和互连设备之间提供高效连接。

【3 层架构】

CXL 设计规范三层架构：事务层、数据链路层、物理层

【3 种子协议】

CXL 事务层由汇总到单个链路的 3 种动态多路通信子协议组成：

CXL.io：用来发现、配置、寄存器访问、错误报告、主机物理地址查找、中断等。CXL 协议在 I/O 模式下运行时，它本质上与用于 I/O 设备的 PCI-Express 外围协议相同。CXL 设备也必须支持 CXL.io。

CXL.cache: 用来扩展系统缓存。CXL.cache 是定义主机（通常是 CPU）和设备（例如 CXL 内存模块或加速器）之间交互的协议。这允许 CXL 设备以低延迟访问缓存在主机内存的数据。可以将其理解为为 GPU 直接缓存数据在 CPU 的内存中。

CXL.memory:用来扩展系统存储。CXL.memory / CXL.mem 是为主机处理器（通常是 CPU）提供使用加载/存储命令直接访问设备内存的协议。将其理解 CPU 可以直接使用 GPU 或加速器上的内存。

【3 种设备】

CXL.io 可以和 CXL.cache 或 CXL.mem 任意组合。比如 Type 1 (CXL.io + CXL.cache)、Type 2（所有三个）和 Type 3 (CXL.io + CXL.mem)。

- Type 1 CXL 设备是一种缓存设备，例如加速器和 SmartNICs（智能网卡）。Type 1 设备可通过 CXL.cache 事务访问主机内存，并维护与主机内存一致的本地缓存。
- Type 2 CXL 设备是 GPU 和 FPGA（现场可编程逻辑门阵列，AI 芯片的一种），具有挂载到设备的 DDR 和 HBM 等存储器。Type 2 CXL 设备可以像 Type 1 CXL 设备一样直接访问主机挂载的存储器。除此之外，Type 2 CXL 设备具有本地地址空间，主机 CPU 可以通过 CXL.mem 事务查看和访问该地址空间。
- Type 3 CXL 设备是内存扩展设备，支持主机处理器通过 cxl.mem 事务一致地访问 CXL 设备存储器缓存。Type 3 CXL 设备可用于实现存储器容量和带宽的扩展，无需增加主 CPU 存储器通道的数量，即可扩展存储器容量和带宽。(过去，要提升系统的存储器容量和带宽，就必须增加原生 CPU 存储器通道的数量。但是，增加存储器通道数会增加 CPU 工程复杂性，并推高了成本。)

【CXL 2.0】

CXL 是服务器的主要设备互连协议，它需要扩展其功能以适应更高级的设备，并最终适应更大的用例。

【内存池化和 CXL switch 带来的内存扩展】

CXL 1.1 解决了内存扩展的问题，而 CXL 2.0 引入的资源共享技术，让不同 XPU 之间实现内存共享成为可能。CXL 2.0 中新增了 CXL switch 与多逻辑设备功能，如此一来每个主机都能用到他们所需的内存，而且可以动态分配。如此一来，不仅解决了系统中 DRAM 没有得到充分利用的问题，也让内存的扩展不再只跟着 XPU 走。这一技术再加上内存扩展，意味着 CXL 与主内存之间共享大型内存池可以将服务器内存容量和带宽得到提高

内存池——工作负载的内存需求与内存池中的可用容量精确匹配 。CXL2.0 支持交换机 switch，通过 switch 的连接，主机可以访问池中的一个或者多个设备 ，主机必须使用 CXL2.0

才能启用这个功能，但构成内存池的设备可以是 CXL1.0、1.1、2.0 的硬件混合。在 1.0/1.1 中，设备被限制一次只能由一台主机访问的单个逻辑设备，但 2.0 级别的设备可以分区成多个逻辑设备，最多允许 16 台主机同时访问内存的不同部分。多个主机可以访问设备的内存，但必须为每个主机分配自己的专用内存段。

支持多扇出 Fan-out 单级交换，以及跨多个虚拟层次结构共享设备的能力，包括对内存设备的多域支持。

【持久性内存标准化管理】

整个业界正在向更为便宜、有效的方向前进，而非傲腾这种造价高昂的独占硬件解决问题。放弃傲腾持久内存并不意味着性能倒退。Intel 的官方策略是转向 CXL 内存技术，这项技术允许通过 CXL 的 PCIe 总线将非易失性存储（比如 SSD）和易失性存储（比如 DRAM）直接连接到 CPU，从而实现与傲腾相似的表现，因此也无需开发完全独立的内存和固态硬盘技术，与现在的主流趋势融合。

【安全增强】

通过使用 CXL 控制器中的硬件加速器来支持队任何通信加密。

【兼容 CXL 1.x】

【CXL 3.0】

【PCIe 6.0】这里就是 PCIe 标准的一些变化

PCIe 6.0 标准变化很大，它将总线上的可用带宽提升了一倍，也就是 64GT/s，这意味着 PCIe 6.0 x16 的带宽可以达到 128GB/s，PCIe 的信号也由原来的二进制 NRZ 信号变成了四态的 PAM4 信号，并结合固定数据包 FLIT 接口实现传输，以避免速率翻倍之后不提升频率的情况。

【CXL 3.0】

CXL 3.0 响应了设备供应商的需求，将比 CXL 1.X 版本提升了带宽，并将一些原本复杂的标准设计简单化，确保易用性。这是 2020 年 CXL 2.0 标准引入内存池和 CXL 开关功能之后较大的改动，CXL 3.0 将侧重于物理和逻辑层面的升级。

【物理层面】

在物理层面，CXL 将每通道吞吐量提升了一倍，达到 64GT/s。相对于 CXL 1.X 和 CXL 2.0 建立在 PCIe 5.0 之上，CXL 3.0 与 PCIe 6.0 规范进行合并，这也使得 CXL 3.0 成为标准建立以来第一次物理层更新。

在 PCIe6.0 传输之上运行 CXL3.0 协议，所有三种类型的驱动程序的带宽都增加了 1 倍，而延迟没有任何增加。由于 256 字节流控制单元或 flit 固定数据包大小（大于 PCIe 5.0 传输中使用的 64 字节数据包），带宽增加到 256GB/sx16 通道（包括两个方向）和 PAM-4 脉冲幅度调制编码，可将 PCIe 可将 PCI-Express 传输上的每个信号的比特数加倍。PCI-Express 协议结合使用循环冗余校验（CRC：cyclic redundancy check）和三向前向纠错（FEC ：three-way forward error correction）算法来保护通过线路传输的数据，这是一种比以前的 PCI-Express 协议更好的方法因此为什么选择 PCI-Express 6.0 和 CXL 3。

CXL 3.0 协议确实具有低延迟 CRC 算法，该算法将 256 B flit 分成 128 B 半 flit，并在这些子 flit 上进行 CRC 检查和传输，这可以将传输延迟减少 2 纳秒到 5 纳秒之间。

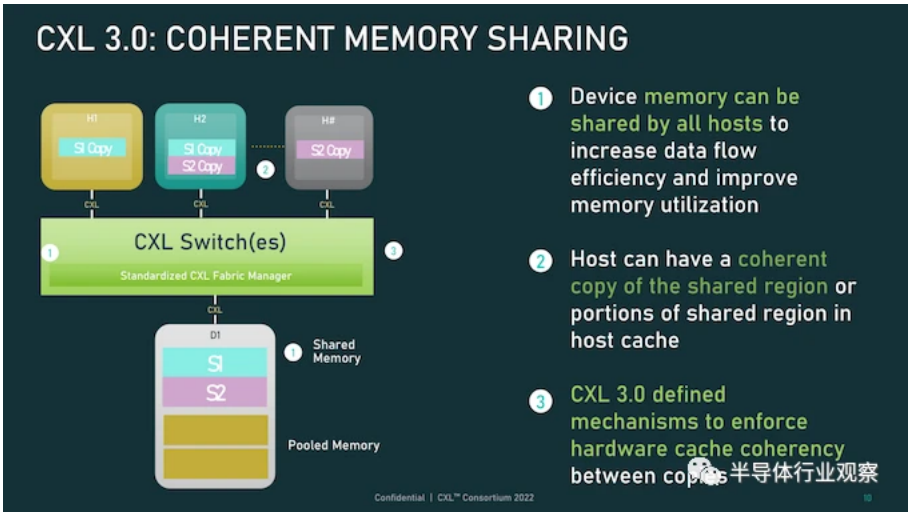
【逻辑层面】逻辑方面的一些升级

【增强的一致性，引入真正的内存共享】

- CXL 最初提供了一种非对称一致性（主机偏向性和设备偏向性，主机偏向性，是指挂载在设备下的存储就好像挂载在主机下面一样，意味着如果想要访问，必须先向主机发送请求，由主机解决 cacheline 的一致性问题），增强的一致性允许设备支持使主机缓存的

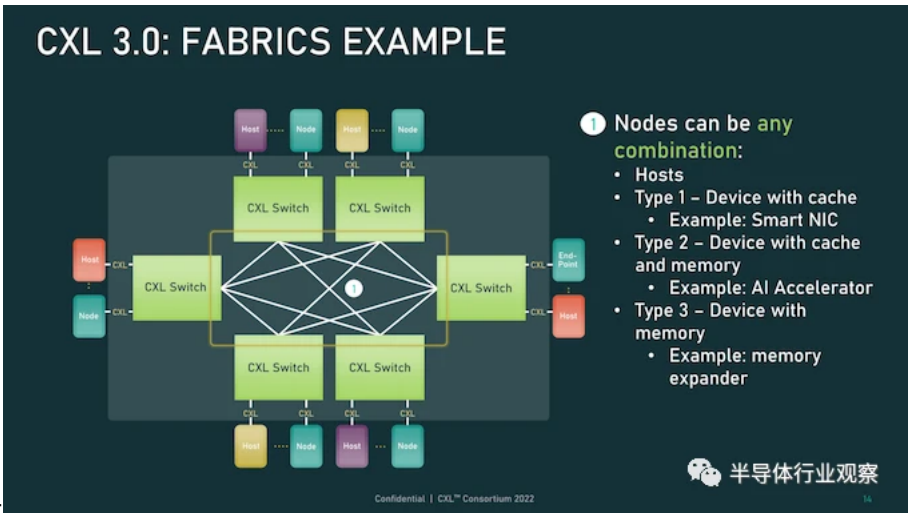
数据无效，设备独占主机存储访问权。这取代了 CXL 早期版本中使用的基于偏差的一致性方法，为了保持简洁，保持一致性不是通过共享内存空间的限制，而是通过让主机或设备负责控制访问。允许 CXL 设备在设备进行更改时通过 Back Invalidation 通知主机。

- CXL2.0 的提供的内存池，是共享设备级别的，设备内部还是要对内存进行划分，为每个主机分配自己的专用内存段。CXL 3.0 引入真正的内存共享，允许多个主机同时访问给定的内存区域，并且仍然保证每个主机都能看到该位置的最新数据，而无需软件管理的协调。利用新的增强一致性语义，多个主机可以拥有一个共享段的一致副本，如果设备级别发生变化，可以使用反向失效来保持所有主机同步。（ 提高数据流效率和内存利用率）但是这并不能完全取代池化。在某些用例中，CXL2.0 风格的池更可取（保持一致性需要权衡取舍，什么用例更可取呢？）CXL3.0 支持根据需要混合和匹配两种模式。



【支持多级 switch，引入 Fabrics 功能】

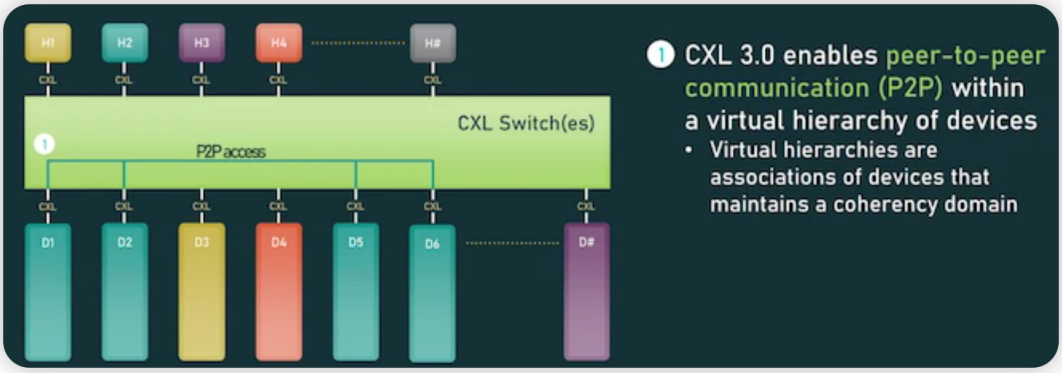
- CXL2.0 引入了对 CXL switch 的支持，但仅允许单个交换机驻留在主机及其设备之间。CXL3.0 将允许多层交换机（2 层）——这极大地增加了所支持的网络拓扑的种类和复杂性。即使只有 2 层交换机，也足以实现非树状拓扑结构，例如环形、网状结构和其他结构设置。并且各个节点可以是主机或设备，对类型没有任何限制。同时 CXL3.0 甚至可以支持 Spine/Leaf 架构，其中流量通过顶级主干节点路由，其唯一工作是将流量进一步路由回包含实际主机的低级（叶）节点/设备。
- 引入 Fabrics 功能。CXL Fabric 可以支持多达 4096 个节点，这些节点可以使用成为给予端口的路由 (PBR,Port Based Routing) 的新的可扩展寻址机制相互通信。节点类型不限，可以获得支持 CXL 的设备的软件定义的动态网络，而不是使用链接特定 CXL 设备的特定拓扑设置的静态网络。



.....

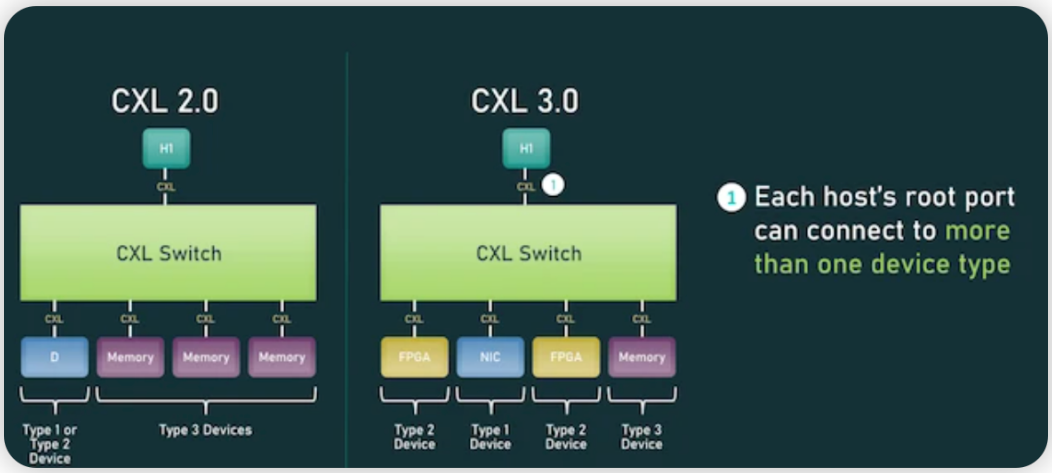
【支持点对点通信】

- 在 CXL3.0 中，设备现在可以直接访问彼此的内存，而无需通过主机，使用增强的一致性语义来通知彼此它们的状态。从延迟的角度来看，跳过主机不仅速度更快，而且在涉及交换机的设置中，这意味着设备不会通过请求占用宝贵的主机到交换机带宽。
- 更大的拓扑，允许将设备组织成虚拟层次结构，其中层次结构中的所有设备共享一个一致性域。CXL Swtich 连接的设备和设备之间进行通信。



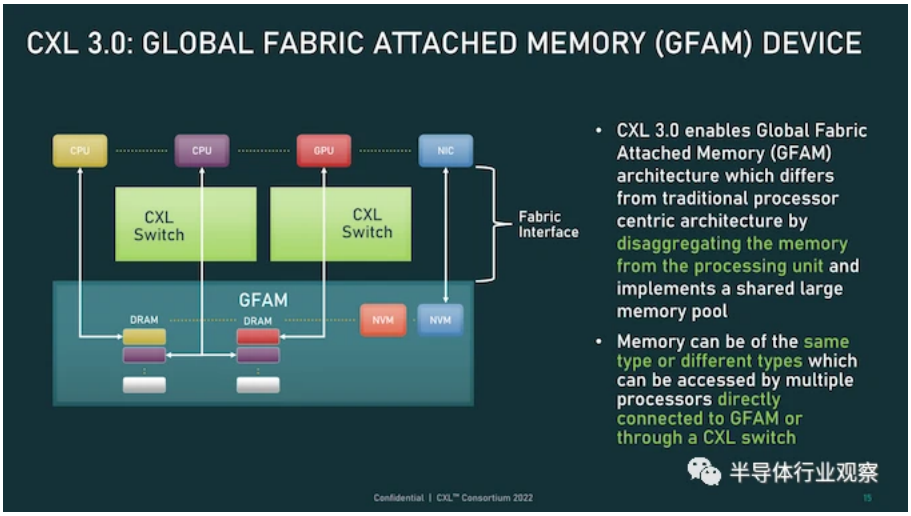
【改进的主机设备功能，允许不同类型和数量的设备连接到主机上的给定根端口】

消除了之前对可以连接到单个 CXL 根端口下游 Type1/2 设备数量的限制。CXL3.0 完全解除了这些限制。现在 CXL 根端口可以至此 Type123 设备的完全混合匹配设置，具体取决于系统构建者的目标。值得注意的是，这意味着能够将多个加速器连接到单个交换机，提高密度（每个主机更多的加速器），并使新的点对点传输功能更加有用。



【GFAM 全局结构附加内存】

- 上面提到的所有这些新的内存和拓扑/结构功能都可以在 GFAM 中一起使用。GFAM 通过进一步分解来自给定主机的内存，将处理单元的内存分解出来构建成一个大的共享内存池，将 CXL 的内存扩展 Type3 理念提升到了一个新的水平。GFAM 设备在功能上是它自己的共享内存池，主机和设备可以根据需要访问它。GFAM 设备内存可以是相同类型的也可以是不同类型的，可以同时包含易失性存储和非易失性存储器，如 DRAM 和 Flash。GFAM 将使 CXL 能够有效地支持大型多节点设置。多种处理器可以直接访问 GFAM 也可以通过 CXL Switch 进行访问。



CXL 用例和产品介绍

单独列出一些产品，剩下的一些可以和上面的不同版本的协议一起分析

三星新推出的 CXL 存储器扩展设备模组配备高达 512GB 的 DDR5 DRAM 存储器，可将服务器的存储器容量扩展到几十 TB，同时还将存储器带宽扩大到每秒几 TB。CXL 存储器扩展设备使用 x8 PCIe 5.0 接口连接到 CPU，每条通道的最大传输速率为 32GT/s。 此外，三星还推出了开源 CXL 软件解决方案，即可扩展存储器开发套件 (SMDK)。它包含了一系列软件工具和 API（应用程序编程接口），支持存储器和 CXL 存储器扩展设备在异构存储器系统中无缝协作。借助 SMDK，系统开发人员无需修改现有的应用环境，即可将 CXL 存储器轻松集成到高级系统中，从而加速 CXL 生态系统的普及。

Intel 第四代可扩展处理器 Sapphire Rapids

CXL 面临的问题

> CXL与PCIe有着很强的绑定，这种结合不仅可以帮助CXL在发展过程中清除更多的风险，也可以确保性能的优异性，起到更好的协同作用。从数据中心应用的角度来说，这两者的目的有一定的不同之处。PCIe是一个非常普及的技术和标准，它更加适用于芯片到芯片之间的互连。CXL增加了一些额外的属性，延迟比较低，同时可以保证缓存一致性，更适用于在分布式计算架构体系之下，来进行内存资源的分配。

CXL 或许不会成为 HPC 和 AI 应用的宠儿

固然 CXL 对于云服务厂商和诸多数据中心拥有不错的吸引力，然而这种形式的内存可能并不适用于 HPC 与超算应用。“富岳之父”松冈聪教授表示 CXL 这种内存解构方案还存在不少技术问题，使其不能在主流的 HPC 甚至是 AI 负载中物尽其用。松冈聪教授并没有给出具体的细节，但他给出了一个例子，那就是多年前 SGI 的 NUMALINK 系统也是采用了分布式内存解构的方式，但我们也都知道如今市面上的 NUMALINK 产品基本已经销声匿迹了。

但他并没有彻底否认内存解构这种思路，就连富岳超算本身也用这一技术，从而将 MPI 进行 put/get 运算时的远程内存访问延迟降低至亚微秒级。但加入一个单独的 UMA 内存池，已经在历史中证明了这对 HPC 来说收效甚微。

首先，这需要更高硬件交换机成本，再者，在超算这种大型配置规模的系统上，缺乏对应的编程标准。因此，对于目前的 HPC 大型系统来说，CXL 内存或许会先出现在一小部分节点上，比如一些需要近存或存内计算 AI 负载，而不会普及到整个系统。

【Meta 和 AMD】

来自 Meta 和 AMD 的两位专家提出了一个概念，也就是对内存进行分层，分为用于实时分析等关键任务的“热”内存、访问不那么频繁的“暖”内存和用于庞大数据的“冷”内存。“热”内存页面放在原生 DDR 内存里，而“冷”内存页面则交给 CXL 内存。

然而在当前的软件眼里，它们才分不清楚什么是“热”内存和“冷”内存，原生内存用完后，就开始去占用 CXL 内存，如此一来原本作为“冷”内存的 CXL，也开始变成“热”内存。所以目前最大的挑战就是在操作系统和软件层面，如何检测到“冷”内存页面，将其主动转入 CXL 内存里，为原生内存留出空间。Meta 和 AMD 的两位专家表示，他们已经在开发相应的软硬件技术。

过去，要提升系统的存储器容量和带宽，就必须增加原生 CPU 存储器通道的数量。但是，增加存储器通道数会增加 CPU 工程复杂性并推高成本。Type 3 CXL 存储器扩展设备提供灵活而强大的方案，无需增加主 CPU 存储器通道的数量，即可扩展存储器容量和带宽。

- 对于服务器市场来说，低核心数的 CPU 依然会继续使用原生 DDR 通道来配置 DIMM 内存。到了高核心数 CPU 上，再根据系统成本、容量、功耗和带宽等参数来灵活应用 CXL 内存，而这才是 CXL 带来的最大优势，灵活性。也许引入 CXL 的延迟后，对性能的损失不会那么糟糕。
- AI 芯片上

use cases 闪存峰会上的：自动内存修复等

《PCIe 体系结构导读》
《大话存储》
CXL 标准看了一部分
gem5 学了一部分
disaggregated memory 方面的论文看了一部分

SNIA：Advancing Storage and Information Technology
PM+computational storage summit 2021
OpenFabric alliance 开放结构联盟
Storage Developer Conference SDC 2021 (2022.09.12-15)
NALLASWAY Inc.
Sumsung
synopsys(新思科技)
闪存峰会