驾驭内存：分离式方法*

作者：潘卡伊·梅赫拉（Pankaj Mehra）
汤姆·考夫林（Tom Coughlin）
译者：王　靖　王鑫凯　李　超

关键词：分离式内存

译者按：计算互联（Compute Express Link，CXL）标准是 CXL 联盟的主打产品。CXL 标准支持创建内存池和加速器池的互联，支持分离式内存和可组合式虚拟机的构建，从而更加高效地使用内存资源。新的软件环境将赋予 CXL 的内存池更强的可用性。

数据中心，尤其是大型数据中心，一直在寻找优化资源利用率的机会。随着集群规模的扩大，充分利用硬件资源的压力也越来越大。为更高效地使用计算资源，人们发明并推广了虚拟化技术，即在物理服务器上搭建多个虚拟机的技术。最近，在传统虚拟机和容器技术基础之上，人们开始利用分离式的（即解耦的）存储和网络部件进一步提升资源使用效率。分离式架构通常会引发池化的计算资源（如处理器、网络、存储设备）间的频繁互访，而系统通过配置，运行了多种进程的虚拟机或者容器，使用软件的方式使这些分离的计算部件重新组合起来使用。这种基于软件的计算资源池化组合方式，又称作可组合式架构（composable infrastructure）。

现如今，存储资源的池化主要使用专用网络（fabric）上的非易失性内存传输协议（如 NVMe-oF），支持存储池中的固态硬盘（SSD），可提供任意寻址的存储空间并分配给容器或虚拟机，从而提高资源的利用率。当前，新的内存网络标准已经可以将内存从以往直接连接的 CPU 设备上解耦，构成内存池。这些内存池可以通过互联网络共享，并作为数据中心的可组合基础设施的一部分进行分配。这种技术有助于数据中心驾驭其内存需求，本文具体介绍该技术当前的发展。

2016 年，Rao 等人 [1] 发现，基于传统网络的分离式内存能够优化 Apache Spark 框架中内存密集且高度可分区的工作负载。2017 年，Barroso 等人 [2] 预测了数据中心中数据访问特性的变化，并鼓励软件开发人员优化软件栈的性能，尤其是当数据访问延时达到 1 微秒量级的情况下。事实上，在 Rao 等人的工作之前，分离式内存架构已初具形态，Lim 等人 [4] 提出从硬件角度分离内存部件，形成独立的内存刀片服务器，这已经呈现出现代分离式内存结构的许多特征。

2019 年，CXL 联盟成立，其目的是创建分离式内存的标准，并构建间接连接到 CPU 的内存池。2020 年 11 月，CXL 联盟发布了 2.0 规范版本 [3]。CXL 3.0 规范版本预计在 2022 年发布（译者注：CXL 3.0 版本已在 2022 年 8 月发布 [11]）。CXL 在 PCIe（Peripheral Component Interconnect express）总线上运

* 本文译自 *Computer*, “Taming Memory with Disaggregation”, 2022(9): 94-98 一文。

行，融合了串行链路的最新技术（例如高速串行总线 SerDes）和几十年前使用少数串行链路的旧想法，在每个链路形成一个 4 路到 16 路位宽的串行链路，可用作系统扩展互连。支持 CXL 的系统基于最新的第 5 代 PCIe 规范，预计将在 2022 年底或 2023 年初推出。

CXL 在 PCIe 上进行了协议层增强，因此特别适合内存互联。首先，它支持流动数位（flow-digit）级的判断，使得长的 I/O 数据包访问和短的缓存行访问能共享相同的物理链路。这样，“加载–存储”（load-store）操作和基于 I/O 的直接内存访问（Direct Memory Access，DMA）操作可以共享相同的物理链路，避免 I/O 传输层数据包在内存数据访问之前穿过交换机端口，从而降低内存访问延时。

其次，CXL 指定了一致性协议，允许缓存和缓冲区在保证数据一致性的情况下，连接到由传统部件和新部件组成的分离式异构系统中的处理器。新部件主要指的是远内存和特定领域加速器，例如高度集成了 SRAM、HBM、DRAM 等内存器件的 FPGA、GPU 和 CGRA 加速器等。图 1 展示了一些 CXL 池化方法。

从单机内存和分布式内存到分离式内存

每一代 CXL 都会支持内存部件部署到离 CPU 更远的地方，并提高系统在部署容量、主机内存动态配置、能够共享和高效访问网络化内存（fabric-attached memory）的主机数量等方面的灵活性。

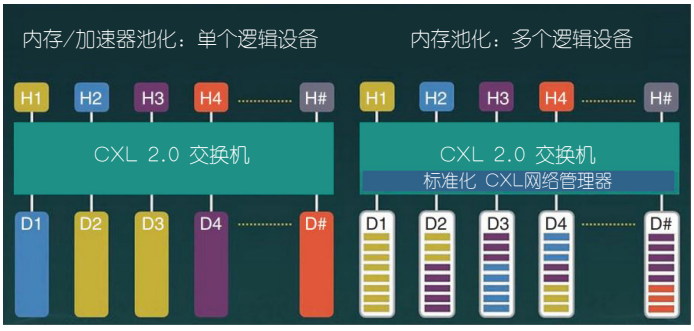


图1　CXL内存/加速器资源池化方法（由CXL联盟提供）

传统定制化的双列直插式存储模块（Dual Inline Memory Modules，DIMM）主要集成于 CPU 插槽的双倍数据速率（Double-Data-Rate，DDR）总线上；与之相比，分离式内存的优势更加明显。在分离式架构下，每个 CPU 插槽可以互联 4 个、6 个甚至 8 个 DDR 通道，并且每个通道支持 2 个 DIMM 插口（目前因为电容负载限制只支持 2 个插口）。

这些 CPU 通过交换式或点对点的对称一致性网络互连，允许使用统一或不统一的延时来读写彼此的内存。PCIe 通道分别从 CPU 插槽出发，通常每个插槽有 96 或 128 个通道，并被路由到 I/O 设备，例如网卡或固态硬盘，在背板或中间板上有或没有 PCIe 交换机和 PCIe 重定时器都是可以支持的。换句话说，CPU 能够同时以一个通路连接到内存，而以另一个通路连接到 I/O 设备。

在 20 世纪 90 年代，随着 I/O 设备的分离，人们第一次尝试通过光纤通道和 IP 网络访问存储器。在本世纪第一和第二个十年，人们开始使用昂贵的网卡乃至智能网卡（SmartNIC，如 Xsigo、VirtenSys 和 Mellanox 多主机网卡），并开始利用 PCIe 满足支持远程直接内存访问（RDMA）的系统对网络通道扩展的需求。尽管 CPU 和其应用软件也能够采用 RDMA 实现高效的处理器间通信，但由于当前系统中内存访问机制存在冗长的软件路径，人们更愿意使用存储协议（如 NVMe）。内存访问机制中存在冗长的软件路径有两方面的原因：(1) 为了满足数据安全和零拷贝的需求，RDMA 保留了复杂的内存注册（memory registration）的软件路径，建立和拆除这些软件路径会产生很大开销；(2) 基于队列对的 RDMA 读取的性能开销较大，RDMA 读写操作的完成路径也较长。相比之下，即使是高估的 CXL 访问延时（与 DIMM 相比）也比保守估计下的 RDMA 数据读取往返时间低一个数量级。

分离式架构的趋势与内涵

分离式内存的含义在一定程度上和 20 世纪 90 年代后期提出的分离式存储（storage

disaggregation）类似。当任何资源与主机服务器解耦时，必须以不同方式对其进行管理。从上电和启动开始，分离式部件的上电顺序难以得到保证。并且，由于分离式设备的采购、退役和故障是相互独立的，分离式系统中各个设备的共同可用性更加难以保障。

乐观来讲，现在我们可以独立地扩展系统的组件，而这在之前是不行的。新的分离式设备的独立管理需求为其增值服务提供了机会。例如，存储阵列开发了许多过去硬件驱动不支持的软件新功能，比如快照、克隆以及精简配置等。我们同样希望分离式内存节点能够从硬件设备发展为一个基于软件的具有越来越多新功能的子系统。

计算和内存设备的独立扩展与同构扩展形成了鲜明的对比。在同构扩展的情况下，定制化内存部署的问题来自急切的超额配置。同时，如果不支付额外的处理器费用并且接受附加的性能延时，就难以获得更大量的内存。

定制化内存部署带来的影响在当今的数据中心中根深蒂固。第一，内存开销已成为数据中心服务商的物料清单中成本最高的一项支出，高达总成本的 50%，而这一开销在 2009 年仅占 25%。在拥有 10 万个服务器的云数据中心，有多达 5~7 种内存容量不同的服务器库存单位（Stock Keeping Units，SKUs），而使用这些固定的库存单位会导致高达 34% 的内存闲置率。第二，由于服务器的内存容量不能动态增长以满足需求，服务器上的应用在它们的内存占用状态到达顶峰时，不得不忍受内存不足的问题，或者不得不将数据移动到更大的存储实体中，这两种方式都有悖于现代的开发运维一体化（DevOps）的概念。第三，应用的内存需求变化非常大，也会带来很多问题。在第五届异构集成国际研讨会上，美国能源研究科学计算中心的首席技术官约翰·沙夫（John Shalf）观察到，服务器工作负载在 75% 的时间里使用不到 25% 的内存。在数据中心定制化地部署内存是非常浪费的，以至于运营商以大约 4 美元 /GB 的价格采购资源，却以每年 22~30 美元 /GB 的价格出租给云服务商，这可能是为了填补存在架构瑕疵的价值链中的损失。

发表于 2022 年国际编程语言和操作系统体系结构支持大会（ASPLOS）的一篇文章中，微软云（Microsoft Azure）的研究人员 [6] 估计，通过仅将冷页面（cold pages，不经常访问的预置内存）放置在 16~32 个服务器共享的基于 CXL 的远内存中，可以节省大约 10% 的内存开销。

分离式内存行业路线图

由于内存密集型工作负载的不断增长，对内存的需求也在增加。在这种情况下，架构师需要更加积极地将内存作为远程的、可替代的、共享的资源来使用。人们已经认识到，自下而上的硬件开发只是朝着正确方向迈出的第一步，例如 CXL。巴罗佐（Barroso）等人 [2] 指出，软件需要针对更多工作负载（而不仅仅是 Spark）进行发展革新，以利用高效能部署的分离式内存资源，虽然可能会带来更高的延时。

分离式内存有特殊的软件需求。首先，需要解决在独立于 CPU 扩展的分离式内存中使用大量数据带来的冲突。其次，需要利用硬件机制提高 CXL 内存中数据的安全级别，这些数据在技术上位于 CPU 之外，生命周期比访问它的进程更长，甚至比处理器设备的寿命更长。最后是当出现故障时，需要解决分离式 CPU 和内存的状态一致性问题。

多个主机访问分离式内存中数据的主要困难在于，这些数据从虚拟地址到物理地址的上下文切换是进程的一个属性，需要由进程管理。进程通过使用微处理器硬件机制，例如页表项以及内存管理单元，来管理数据从虚拟地址到物理地址的上下文切换。

最新的设备端软件正在借鉴内存和存储之间的类比关系，并为分离式内存构建类似 S3 等为云存储构建的服务 ：构建于自包含对象之上的基础。在这些新产品中，内存对象保全了图结构数据和计算的上下文切换信息，并以驻留在每个内存对象已知位置的外部对象表的形式，嵌入其他必要信息。

内存高效的指针可以利用正确构造的对象（主要指构建对象内指针），通过在外部对象表中存储

唯一的128位全局对象标识符解析外部对象指针。对象内指针可以通过仅存储对象内的偏移量避免开销。此类技术支持Elephance内存操作系统（MemOS）公开全局引用，可用于描述计算和数据。这些数据可以灵活地放置在分离式系统中，也可以使用更高效的引用传递参数，在服务之间传递指向数据的指针[8]，而不是相对低效的值传递。例如，现有的大数据微服务应用，通常采用远程过程调用（RPC）机制，这种机制属于值传递。

除非发生内存分配、解除分配或指针取消引用的事件，用于分离式内存节点的新系统软件应当知道如何避开硬件数据路径。然而，我们更需要保护远内存中数据的安全，尤其是在进程、操作系统或保存最后写入数据的服务器发生故障之后。Elephance内存操作系统将不断改进以利用分离式架构的功能[9]，这些功能是硬件强制的许可机制，对内存不安全的语言也能提供空间、时间和引用上的安全性。就像15年前在Sinfonia[10]上为分布式内存所做的工作一样，用于分离式内存的软件工作需要提供一种安全的方式来更新远程内存中保存的数据，以防在任一端远程操作发生故障时遭受一致性失败的风险。目前新的研究有望解决该问题。

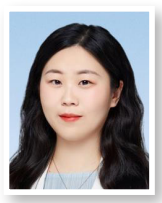
分离式内存的重要性正日益凸显。它将对数据中心服务商的业务产生极大的影响。为了发掘这项新技术的全部潜力，需要软件技术不断发展，需要通过安全、便携和高效的机制来利用远距离和可替代的内存资源，还需要加强数据共享、重视数据安全。 ■

作　者：

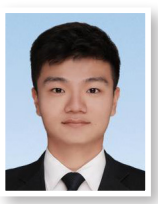
潘卡伊·梅赫拉（Pankaj Mehra）
Elephance Memory公司创始人，来自美国圣何塞。

汤姆·考夫林（Tom Coughlin）
Coughlin Associates总裁，IEEE Fellow，来自美国圣何塞。

译　者：



王　靖
CCF学生会员。上海交通大学博士生。主要研究方向为计算机体系结构、分离式内存、图数据处理。
jing618@sjtu.edu.cn



王鑫凯
CCF学生会员。上海交通大学博士生。主要研究方向为计算机体系结构、微服务架构优化、边缘智能系统。
unbreakablewxk@sjtu.edu.cn

李　超
CCF高级会员，CCF体系结构专委会副主任，*CCCF*译文编委。上海交通大学教授。主要研究方向为计算机体系结构、数据中心、云计算与大数据系统。
lichao@cs.sjtu.edu.cn

（本文责任编委：**李　超**）

参考文献

[1] Rao P S, Porter G. Is memory disaggregation feasible? A case study with Spark SQL[C]//*2016 ACM/IEEE Symposium on Architectures for Networking and Communications Systems(ANCS)*. IEEE, 2016: 75-80.

[2] Barroso L, Marty M, Patterson D, et al. Attack of the killer microseconds[J]. *Communications of the ACM*, 2017, 60(4): 48-54.

[3] CXL Consortium, “CXL 2.0 Specification”, 2020. https://www.computeexpresslink.org/download the-specification

[4] Lim K, Chang J, Mudge T, et al. Disaggregated memory for expansion and sharing in blade servers[J]. *ACM SIGARCH Computer Architecture News*, 2009, 37(3): 267-278.

[5] Shalf J, Michelogiannakis G, Austin B, et al. Photonic memory disaggregation in datacenters[C]//*Photonics in Switching and Computing*. Optica Publishing Group, 2020: PsW1F. 5.

[6] Li H, Berger D S, Novakovic S, et al. First-generation Memory Disaggregation for Cloud Platforms[OL]. arXiv preprint arXiv:2203.00241, 2022.

[7] Bittman D, Alvaro P, Mehra P, et al. Twizzler: a data-centric OS for non-volatile memory[J]. *ACM Transactions on Storage* (*TOS*), 2021, 17(2): 1-31.

[8] Bittman D, Soulé R, Miller E L, et al. Don't Let RPCs Constrain Your API[C]//*Proceedings of the Twentieth ACM Workshop on Hot Topics in Networks*. 2021: 192-198.

[9] Woodruff J, Watson R N M, Chisnall D, et al. The CHERI capability model: Revisiting RISC in an age of risk[C]//2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA). IEEE, 2014: 457-468.

[10]Aguilera M K, Merchant A, Shah M, et al. Sinfonia: a new paradigm for building scalable distributed systems[J]. ACM SIGOPS Operating Systems Review, 2007, 41(6): 159-174.