

Taming Memory With Disaggregation

Pankaj Mehra, Elephance Memory

Tom Coughlin, Coughlin Associates

The Compute Express Link (CXL), a trademark of the CXL Consortium, protocol enables creating pools of memory and accelerators, allowing memory disaggregation, and composable virtual machines that make more efficient use of memory. New software will make CXL memory pools even more useful.

be reaggreated using software to configure virtual machines or containers for running various processes. The software-based combination of pooled computer resources is also known as *composable infrastructure*.

Storage pooling today focuses on using nonvolatile memory express (NVMe) running on fabrics (NVMe-oF), allowing arrays of solid-state drives (SSDs) in a storage pool that can then be assigned to provide storage for containers or virtual machines that can be spun up

Data centers, especially the large ones, are constantly seeking to optimize their resource utilization. With scale comes increasing pressure to get the most out of one's hardware. The requirement to use compute resources more efficiently, for instance, led to the widespread use of virtual machines running on servers and, more recently, to creating virtual machines or containers utilizing disaggregated (separated) storage and networking components. Disaggregation usually results in interconnected pools of computer resources, such as processors, networks, and storage, which can then

and down at will, resulting in a much higher utilization of storage resources. New memory networking standards are now making it possible to disaggregate memory beyond today's direct connection to a CPU toward memory pools that can be shared on an interconnection network and allocated as part of a data center's composable infrastructure. Let's examine these developments, which will help future data centers tame their memory needs.

In 2016, Rao and Porter¹ found memory disaggregation over traditional networks favorable for Apache Spark's memory-intensive and highly partitionable workloads. In 2017, Barroso et al.² anticipated the changing access characteristics of data in data centers and encouraged software developers to address a gap in their stacks when it came to



accessing data that was approximately 1 μ s away. A form of disaggregating memory was possible even before Rao and Porter's work. Hardware proposals for stand-alone memory blades⁴ anticipated many of the aspects of modern memory disaggregation fabrics.

In 2019, the Compute Express Link (CXL) Consortium was formed to create standards for disaggregating memory and creating memory pools indirectly connected to CPUs. In November 2020, the CXL Consortium released its 2.0 specification.³ The CXL 3.0 specification release is expected sometime in 2022. CXL runs on the Peripheral Component Interconnect Express (PCIe) bus and uses advances in serial link technology (such as high-speed SerDes) and the decades-old idea that a handful of serial links, each forming a lane of 4x-to-16x-wide serial links, can serve as a system-expansion interconnect. CXL-enabled systems are expected by the end of 2022 or early 2023, based upon the latest PCIe specification, generation 5.

CXL makes protocol-layer enhancements to PCIe that make it especially apt for memory attachment. First, it allows long input-output (I/O) packets and short cache-line grain accesses to share the same physical link by supporting arbitration at the flow-digit level so that load-store operations and I/O direct memory access (DMA) operations can share the same physical link without memory accesses incurring exorbitant latencies due to I/O Transport Layer packets crossing switch ports in front of memory data. Second, it specifies coherence protocols that allow caches and buffers to be coherently connected to processors inside a disaggregated heterogeneous system composed of both traditional elements, such as general-purpose CPUs with their tightly coupled memory devices, and novel elements, such as far memory and domain-specific accelerators (field-programmable gate

arrays, GPUs, and coarse-grained reconfigurable arrays with highly integrated static random-access memory or high-bandwidth memory dynamic random-access memory). Figure 1 shows some CXL pooling approaches.

FROM IN-SERVER AND DISTRIBUTED MEMORY TO DISAGGREGATED MEMORY

Each generation of CXL will allow memory to be deployed farther from the CPU with increasing flexibility in terms of the capacity deployed, the dynamic configuration of host memory capacity, and the number of hosts able to share and efficiently access fabric-attached memory. The benefits of this are best understood in contrast with the traditional bespoke deployment of dual in-line memory modules (DIMMs) on the double-data-rate (DDR) buses of CPU sockets, each socket exposing four, six, or even eight DDR channels and allowing two (lately just one because of capacitive loading) DIMMs per channel.

Those CPUs were interconnected via a switched or point-to-point symmetric coherency fabric that allowed uniform or nonuniform latency of load-store access to each other's memory. The lanes of PCIe emanated from CPU sockets separately, often with 96 or 128 lanes per socket, and were routed to I/O devices, such as network interface

cards (NICs) or SSDs, with or without switches and retimers on the backplane or midplane. In other words, the CPUs were attached to memory in one way and to I/O in another.

Because of the disaggregation of I/O, first providing access to storage over Fibre Channel and IP networks in the 1990s and subsequently using the more expensive NICs and SmartNICs (Xsigo, Virtensys, and Mellanox Multihost NICs) during the 2000s and 2010s, PCIe was created to meet the need for system-expansion fabrics capable of supporting remote DMA (RDMA). Although CPUs and their application software also adopted RDMA for efficient interprocessor communication, the heavy software path of setting up and tearing down the memory registrations required for safe, zero-copy RDMA and the heavy queue-pair-based issue and completion paths of RDMA read and write operations remind one more of storage protocols (such as NVMe) than of memory access. By contrast, it is expected that even the higher CXL latencies (compared to DIMMs) will be an order of magnitude lower than the lower RDMA read round-trip time.

DISAGGREGATION-RELATED TRENDS AND THEIR IMPLICATIONS

Some of the implications of memory disaggregation are similar to those

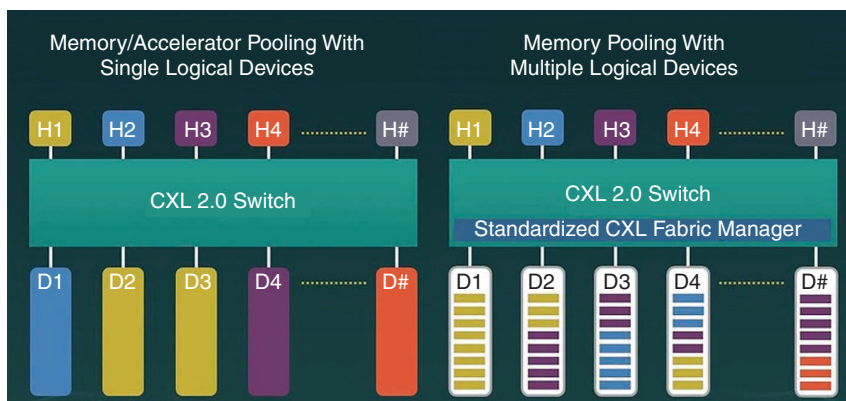


Figure 1. CXL memory/accelerator pooling approaches. (Courtesy of the CXL Consortium.)

of storage disaggregation in the late 1990s. When any resource decouples from a host server, it must be managed differently. Starting with power-up and boot, there are fewer ordering guarantees over the power-up sequence across disaggregated components. Because of the independence of procurement and decommissioning of resources and also because of independent failures, there are fewer assurances of coavailability.

On the positive side, one may now independently scale components that previously could not be. The independent manageability required of the freshly disaggregated components creates an opportunity for value-added services. For instance, storage arrays developed many new software-based capabilities not previously available in hard disk drives, such as snapshots, cloning, and thin provisioning, to name a few. We likewise expect disaggregated memory nodes to evolve from devices into subsystems with a growing list of novel software-based capabilities.

Independent scaling of computation and memory is to be contrasted with homogeneous scale-out, where the sins of bespoke memory deployment were compounded by eager overprovisioning and the inability to acquire more memory without the cost and latency of additional processors.

Moreover, the economic impact of bespoke memory deployment runs deep in today's data centers. First, memory has now become the costliest element of a data center server's bill of materials, accounting for as much as 50% of the overall cost compared with 25% in 2009.⁴ For this reason, as many as five to seven server stock keeping units (SKUs) are commonly found in a 100,000-server cloud data center, mainly differing in their memory capacity. The use of these fixed SKUs can result in up to 34% of memory capacity remaining idle.

Second, because of the inability to dynamically grow the memory capacity of a server to match demand, applications are forced to consider either tolerating out-of-memory errors or moving

their data to larger instances, just when the footprint of their state is at its peak, neither of which is particularly palatable to modern DevOps.

Third, as if that weren't enough trouble, the capacity needs of applications vary wildly.⁴ Speaking at the Fifth International Symposium on Heterogeneous Integration, John Shalf, the CTO of the National Energy Research Scientific Computing Center, has observed that server workloads use less than 25% of their memory, 75% of the time.⁵ So wasteful is bespoke deployment of memory in the data center that a resource that is procured by data center operators at approximately US\$4/gigabyte is then rented out to cloud service operators at approximately US\$22-US\$30/gigabyte/year, probably to make up for the losses in a poorly architected value chain.

In their 2022 Architectural Support for Programming Languages and Operating Systems conference article, Microsoft Azure researchers⁶ estimate that they can save approximately 10% of overall memory cost by placing just the cold pages (infrequently accessed provisioned memory) in a CXL-based far-memory tier shared between 16 and 32 servers.

INDUSTRY'S ROAD MAP OF MEMORY DISAGGREGATION

Given that the demand for memory keeps rising because of the growth of memory-intensive workloads, architects will need to get much more aggressive about leveraging memory as a far, fungible, and shared resource. There has been some recognition that bottom-up hardware developments, such as CXL, are merely a first step in the right direction. The guidance of Barroso et al.² is that software needs to evolve for more workloads (than just Spark) to take advantage of memory that is cost-effectively deployed but may incur higher latency.

There are unique software requirements for disaggregated memory. The first of these is the friction of using rich data in disaggregated memory

from independently scaled CPUs. The second is an enhanced need for leveraging hardware mechanisms to raise the level of security for data in CXL memory, which is technically located outside the CPU and thus may outlive processors and processes. A related final issue is state consistency in the face of decoupled CPU and memory failures.

The principal difficulty of multiple hosts accessing data in disaggregated memory is that the virtual-to-physical-address translation context of those data is a property of the process that is managed for the process by using microprocessor hardware mechanisms, such as page table entries and memory management units.

New device-side software is drawing upon the analogy between memory and storage and building for disaggregated memory what services such as S3 built for cloud storage: a foundation based on self-contained objects.⁷ In these new products, memory objects rescue the translation context required by graph-structured data and compute and embed the necessary information in the form of a foreign object table that resides at a known location in every memory object.

Memory-efficient pointers take advantage of properly constructed objects (mostly intraobject pointers) to store unique 128-bit global object identifiers within the foreign object table for resolving extraobject pointers. Intraobject pointers can avoid the overhead by storing just the intraobject offsets [Figure 2(a)]. Such techniques allow an Elephance MemOS to expose global references [Figure 2(b)] that can be used in describing computations and data that 1) can be placed flexibly within the disaggregated system and 2) can use the more efficient parameter passing by reference to communicate pointers to data between services⁸ rather than the relatively inefficient parameter passing by value used in current Remote Procedure Call (RPC) mechanisms employed by existing data-rich microservices.

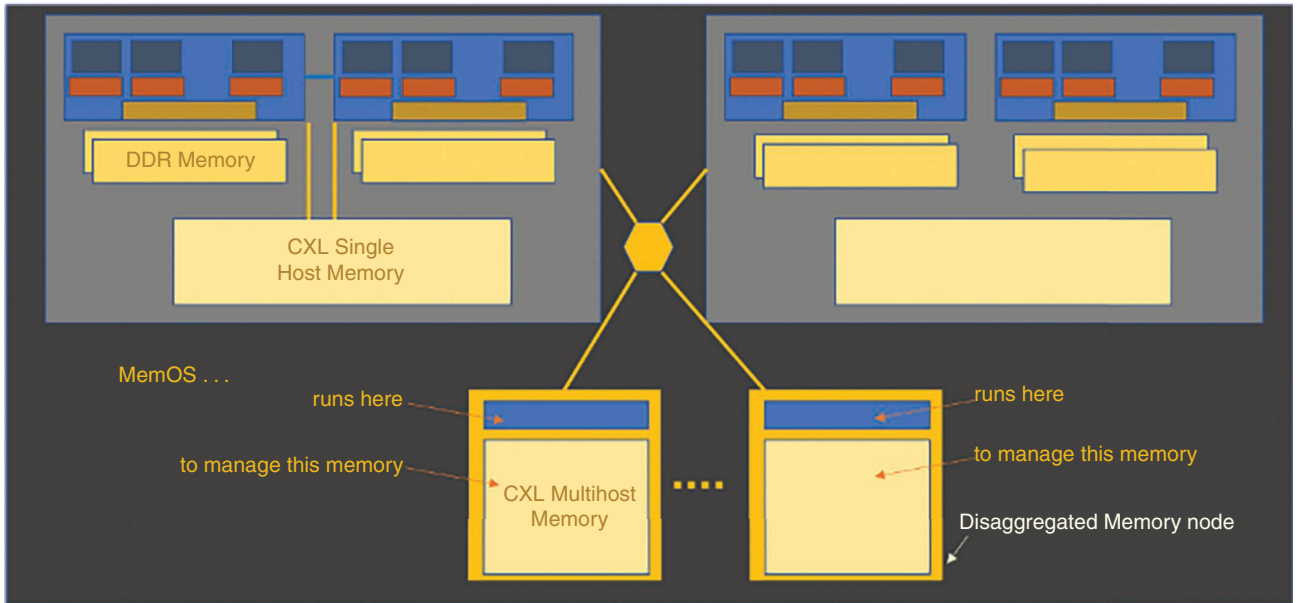
New OS software for disaggregated memory nodes knows how to keep out of the hardware data path except in the

events of memory allocation, deallocation, or pointer dereferencing. However, there is also an enhanced need to protect the data held in far memory even after the failure of a process, OS, or server hosting the computation that last wrote the data. Elephance MemOSs

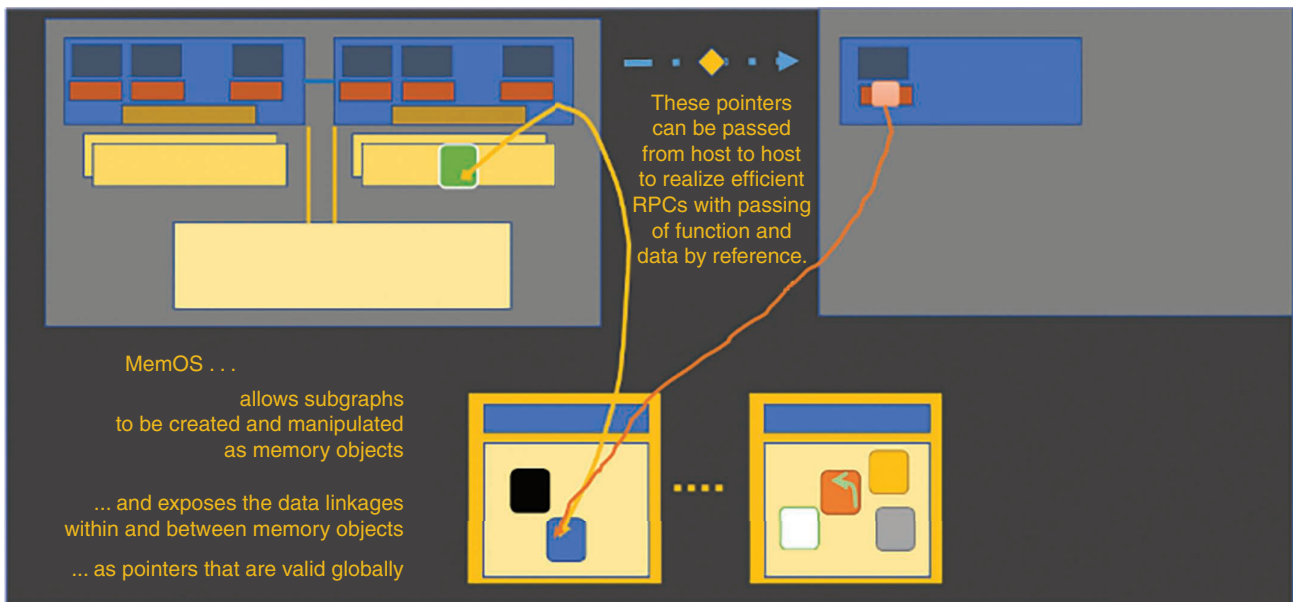
will evolve to exploit architectural capabilities,⁹ which are hardware-enforced permission mechanisms that deliver spatial, temporal, and referential safety even to memory-unsafe languages.

Finally, much as the work on Sinfonia¹⁰ did 15 years ago for network

distributed memory, the software work for disaggregated memory needs to offer a safe way to mutate data held in far memory without risking consistency should failure occur at either end of the remote operation. Fresh research is currently in progress to address that issue.



(a)



(b)

Figure 2. (a) Elephance MemOS runs on each disaggregated memory node and allows memory objects with internal and external persistent pointers to be created within its managed memory. (b) Pointers that are globally meaningful can lead to more efficient communication for data-heavy distributed applications. MemOS: memory operating system; RPC: Remote Procedure Call. (Courtesy of Elephance Memory.)

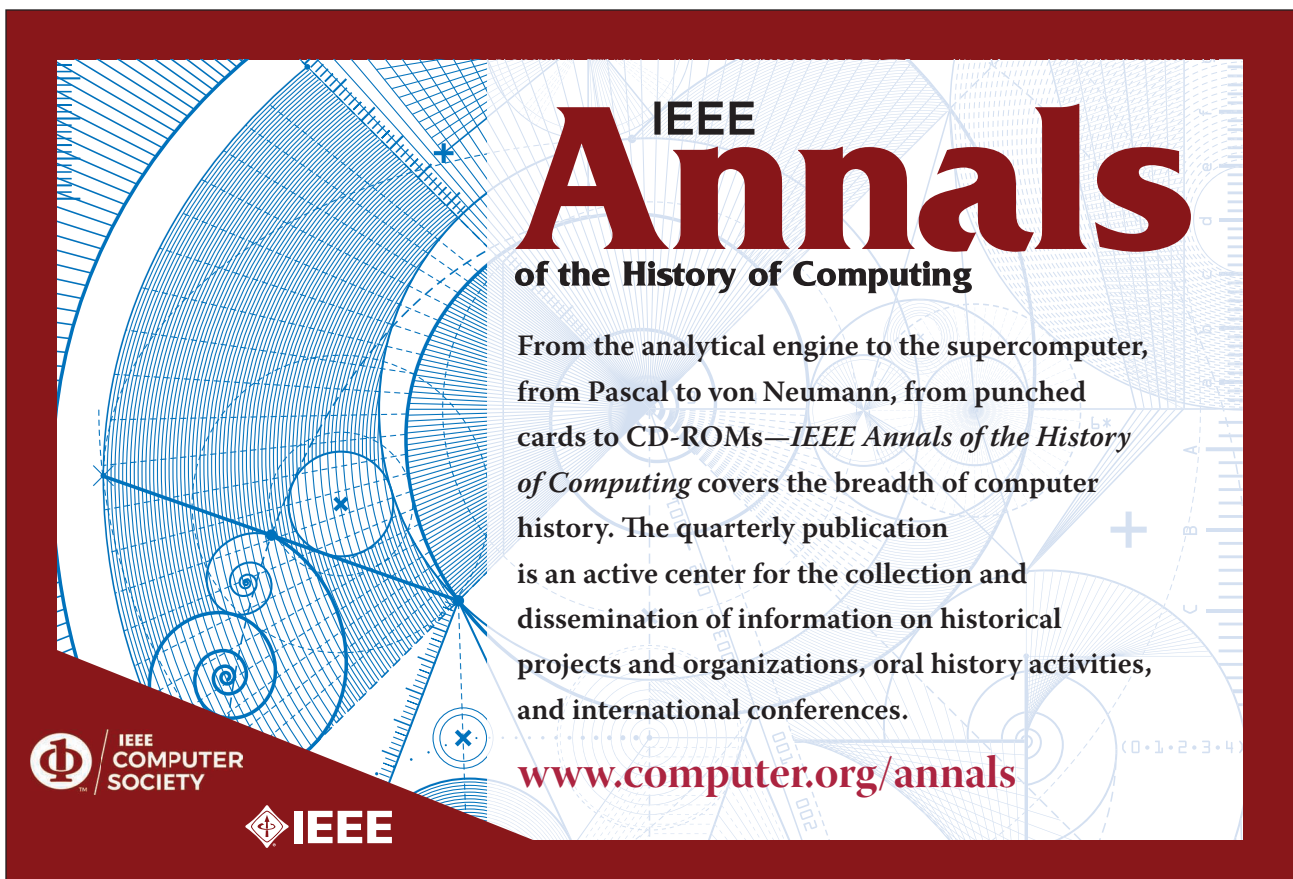
Memory disaggregation is addressing a problem with high economic impact in data center servers. To realize the full potential of this new technology, software will evolve to exploit far and fungible memory through safe, portable, and efficient mechanisms that enhance data sharing and respect data gravity. **□**

REFERENCES

1. P. S. Rao and G. Porter, "Is memory disaggregation feasible? A case study with spark SQL," in *Proc. Symp. Architectures Netw. Commun. Syst. (ANCS '16)*, New York, NY, USA, 2016, pp. 75–80, doi: 10.1145/2881025.2881030.
2. L. Barroso, M. Marty, D. Patterson, and P. Ranganathan, "Attack of the killer microseconds," *Commun. ACM*, vol. 60, no. 4, pp. 48–54, Apr. 2017, doi: 10.1145/3015146.
3. "CXL 2.0 Specification," CXL Consortium, Beaverton, OR, USA, Nov. 10, 2020. Accessed: Jun. 20, 2022. [Online]. Available: <https://www.computeexpresslink.org/download-the-specification>
4. K. T. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch, "Disaggregated memory for expansion and sharing in blade servers," *ACM Sigarch Comput. Architecture News*, vol. 37, no. 3, pp. 267–278, 2009, doi: 10.1145/1555815.1555789.
5. J. Shalf *et al.*, "Photonic memory disaggregation in datacenters," in *Proc. Photon. Switching Comput.*, Optica Publishing Group, 2020, p. PsW1F-5.
6. H. Li *et al.*, "First-generation memory disaggregation for cloud platforms," 2022, arXiv:2203.00241.
7. D. Bittman, P. Alvaro, P. Mehra, D. D. Long, and E. L. Miller, "Twizzler: A data-centric OS for non-volatile memory," *ACM Trans. Storage*, vol. 17, no. 2, pp. 1–31, May 2021, doi: 10.1145/3454129.
8. D. Bittman *et al.*, "Don't let RPCs constrain your API," in *Proc. 20th ACM Workshop Hot Topics Netw.*, 2021, pp. 192–198, doi: 10.1145/3484266.3487389.
9. J. Woodruff *et al.*, "The CHERI capability model: Revisiting RISC in an age of risk," *ACM Sigarch Comput. Architecture News*, vol. 42, no. 3, pp. 457–468, 2014, doi: 10.1145/2678373.2665740.
10. M. Aguilera, A. Merchant, M. Shah, A. Veitch, and C. Karamanolis, "Sinfonia: A new paradigm for building scalable distributed systems," in *Proc. 21st ACM SIGOPS Symp. Oper. Syst. Principles*, 2007, pp. 159–174, doi: 10.1145/1323293.1294278.

PANKAJ MEHRA is the founder of Elephance Memory, San Jose, 95129, California, USA. Contact him at pankaj.mehra@ieee.org.


TOM COUGHLIN is president of Coughlin Associates, San Jose, California, 95124, USA. He is a Fellow of IEEE. Contact him at tom@tomcoughlin.com.




IEEE
Annals
of the History of Computing

From the analytical engine to the supercomputer, from Pascal to von Neumann, from punched cards to CD-ROMs—*IEEE Annals of the History of Computing* covers the breadth of computer history. The quarterly publication is an active center for the collection and dissemination of information on historical projects and organizations, oral history activities, and international conferences.

www.computer.org/annals

 IEEE COMPUTER SOCIETY

 IEEE