OBJECTIVE ANALYSIS
Semiconductor Market Research

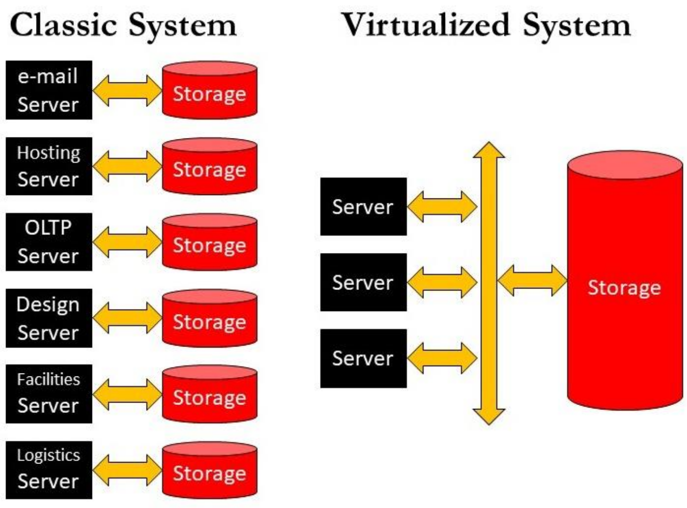
OBJECTIVE ANALYSIS BRIEF

THE FUTURE OF THE DATA CENTER

Memory and Storage Take on an Increasing Role

Data center architecture has undergone many changes over the decades. More recently, the advent of virtualization, as illustrated below, enabled improvements over existing solutions (left) in which users would own a number of servers, each spending maybe 50% of the time doing a specialized job and the other 50% of the time idle. The virtualized system (right) allowed users to deploy fewer servers and have less idle time while getting nearly the same throughput by moving tasks between unassigned servers which shared data sets via a common storage pool.

The success of this approach led to growing use of pooled resources, and disaggregation followed, where both server resources and storage resources were allowed to be dynamically allocated to different applications.



Memory & Storage Focus

Over time, memory has become increasingly important to data center workloads, and now memory – not the processor – is the limiting factor when it comes to bandwidth. Data center operators and customers are looking at different architectures and new ways of organizing memory. Innovations like processing in memory and composable memory pools are needed to meet the demands of data-intensive workloads and enable a more efficient data center. Computational storage is also garnering attention as a means of reducing data movement and offloading some processing from the server's workload.

Data-Intensive Workloads

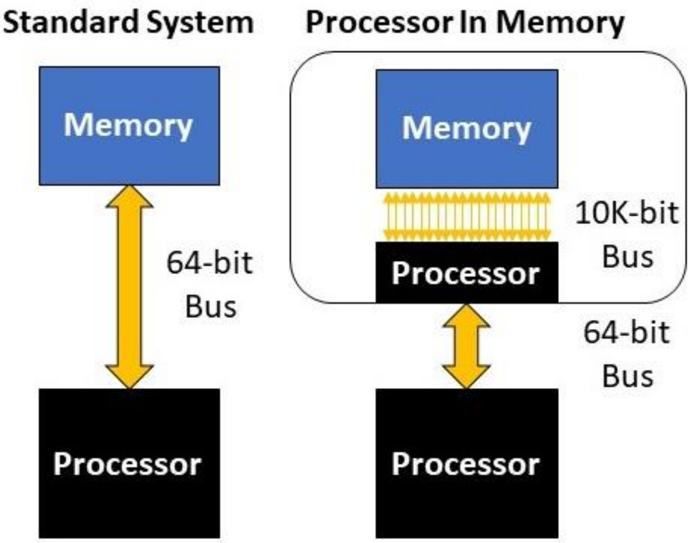
Just as virtualization placed new demands on storage and I/O, new workloads that are gaining importance are driving the use of innovative and scalable resources.

For example, the growth of data-centric workloads like artificial intelligence (AI) is forcing changes to the way the data center is architected. System memory bandwidth cannot keep pace with CPU core growth, especially in compute environments where accelerators are paired with CPUs to address these data-intensive workloads. As workloads continue evolving to incorporate even more complex AI-based tasks like machine learning or natural language processing, the balance of compute to memory or storage access and connectivity must shift. This means that some existing data center configurations may not be well suited for these new workloads. Today's data center investments must be flexibly managed.

In-Situ Processing

One idea that has been discussed for almost the entire history of memory chips has been the in-memory processor, often called "Processor in Memory," or PIM. Memory chips are internally organized with extremely wide data paths, and those wide paths can provide enormous bandwidth for problems that require high bandwidth. If the data need not move off the chip, then latency times can be slashed as well.

The diagram on this page illustrates this. The system on the left might have a very powerful processor communicating over a 64-bit bus with a standard memory. Sometimes the processor will be performing at its maximum capability, while at other times it will be idle. The diagram on the right shows a system with an additional processor incorporated within the memory chip that takes advantage of a very wide (10K bit) data path internal to the memory chip. These processors are typically very limited, yet their capabilities are matched to a very wide data path. By offloading certain tasks to this in-memory processor, the main processor at the bottom might not need to be as powerful. The goal is to use this approach to improve the overall cost/performance of the system.



In a similar vein, numerous companies are working to bring processing and memory together within the SSD[*]. These "Computational Storage Devices" harness the large internal bandwidth of an array of NAND flash chips by putting a processor either within or near the SSD itself.

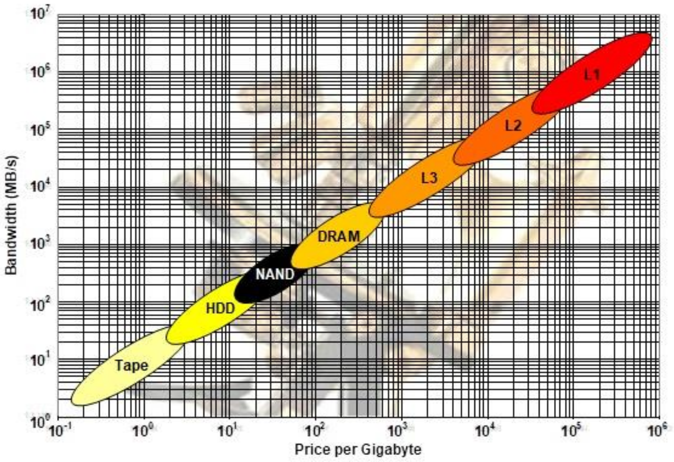
Still, changes like PIM and computational storage require software support, generally provided by stripping functions out of popular programs to be delegated to the new smart memory or storage devices. Although this can provide significant performance improvements, most software companies are hesitant to break up a field-proven product to support a hardware technology that may not be around in a decade (or less). In this way, business considerations are delaying the widespread adoption of both PIM and computational storage.

Cache vs. Memory

One popular way to hide the slow speed of the memory-processor interface is to put a cache memory within the processor chip itself. This is a sort of shell game in which the application program doesn't realize exactly where (physically) the code or data resides. That application program is able to access the code or data rapidly, though, thanks to hardware or software

[*] Eideticom, NGD, Samsung, ScaleFlux, Xilinx, and others.

that manages to keep the most frequently accessed data within fast, local memory while directing seldom-used items to slower/cheaper storage or memory.



Computers have been using techniques like this since the 1960s, with the invention of virtual memory, a system that combines memory (at that time core, but today it's DRAM) and disk storage to fool the application program into seeing a space that is as large as the disk and almost as fast as the memory. The invention of virtual memory was the first in a series of additions to the memory/storage hierarchy, which today includes not only HDDs and DRAM, but also processor caches, SSDs, and archival storage either on tape, in the cloud, on optical drives, or in spun-down HDDs.

The basic mechanisms of virtual memory are borrowed and applied today to processor cache memories as well as to storage management programs that manage data between SSDs and HDDs.

Memory/Storage Hierarchy

This mix of slower/cheaper with faster/costlier memory and storage has been named the memory/storage hierarchy. The illustration on this page graphically depicts this hierarchy.

Each level of the hierarchy, indicated by an orb, is faster than the next-cheaper orb and slower than the next-more-expensive orb. As long as each level fits into this scheme then it can be used to reduce the cost of the system as a whole.

Difficult decisions must be made about the correct way to arrange each of these levels. There is no exact science to determine how much memory or storage capacity should be used at each level, and these requirements depend on the exact set of application software that is being run on the system. Because of this, each system design will have different amounts of cache memory, DRAM, and SSD and HDD storage, and some systems might leave out certain layers altogether.

New Interfaces

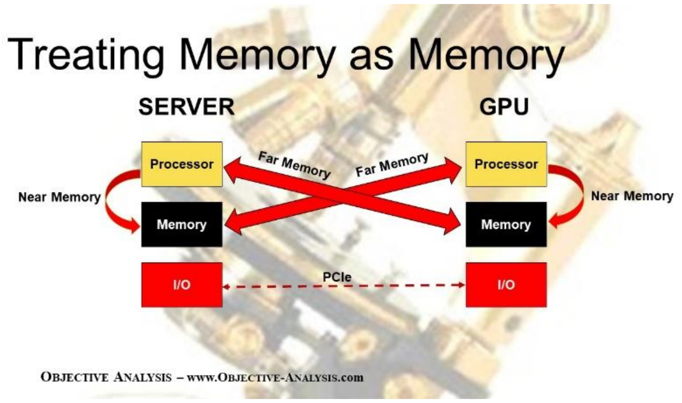
In the memory-storage hierarchy chart, all of the semiconductor technologies, those from NAND up to the L1 cache, accelerate over time. This drives the need for new interfaces, as does the insertion of a new layer, like the advent of SSDs fifteen years ago.

In the case of SSDs, the HDD interfaces were slow enough that they bogged down the SSD, preventing systems from taking advantage of all of the speed that NAND flash provided. SSDs started to use the PCIe interface rather than HDD interfaces, and the NVMe protocol was developed to add SSD-specific commands to the PCIe interface. As time progressed, SSD designers and users gained greater insight into how SSDs were being used and devised new ways to improve system performance, leading to the creation of the recently-released NVMe 2.0 specification. Likewise, the PCIe interface has been refined and accelerated and is now in its fourth generation in mainstream server use: PCIe 4.0.

DRAM is also going through a similar transition. Today the DRAM market is beginning to transition from DDR4 to DDR5, bringing new features, error correction, and higher bandwidth to the processor/memory interface. This will support faster and more efficient servers, but large workloads have created another new opportunity that calls for an innovative approach.

Edge and Endpoint Processing

In the Internet of Things (IoT), steps are being taken to reduce data traffic, accelerate response times, and take advantage of Moore's Law price decreases, by pushing processing resources out of the data center and closer to the source of the data — or to the point where the data is "consumed." In certain cases, like image recognition cameras, the endpoint is given the resources to analyze the data, sending a simple "match" signal back to the cloud, rather than the video data itself. In instances of edge processing more sophisticated processes are performed at a small-scale computing installation that is close to a collection of endpoints. Such a system might be used to monitor all of the instrumentation in a manufacturing plant, sending only the compiled and pre-analyzed data on to company headquarters.



OBJECTIVE ANALYSIS – www.OBJECTIVE-ANALYSIS.com

Each of these systems use remote processing and memory to perform tasks that help to minimize communications traffic. The cheaper semiconductors become, the more reasonable it is to use this kind of approach.

The Changing Role of Memory

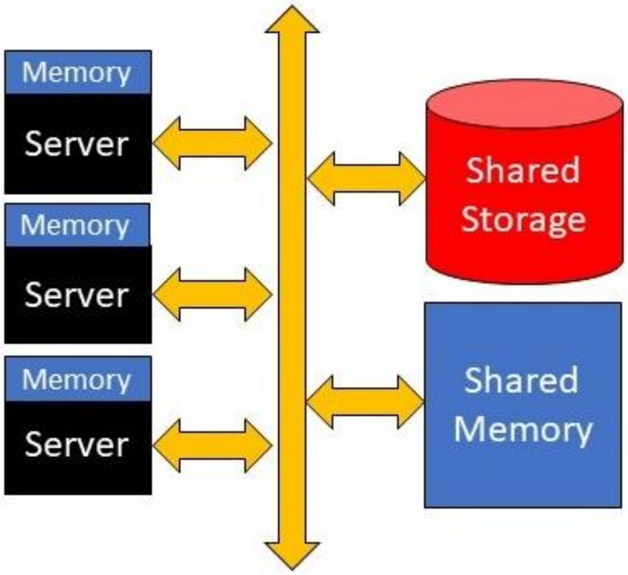
As computers share an increasing number of resources, memory has been given a new role. It is no longer simply a fast place to temporarily save code and data for a single processor, but is now blossoming into a role in which it also rapidly shares data between processors.

This has driven the adoption of new technologies. For example, with the growing adoption of GPUs for AI acceleration, system architects learned that it was cumbersome to move data into and out of the GPU's on-board DRAM. How could this be accelerated?

The problem stems from the fact that the DRAM on a GPU board was loaded by the host system processor through the PCIe I/O channel, the dotted line at the bottom of this page's graphic. Although PCIe hardware runs enormously fast, the software protocol was written around slow I/O devices causing the server's access to the GPU's memory to be much slower than necessary. This spawned the creation of several new protocols, CXL, OpenCAPI, CCIX, and Gen-Z, to allow memory data to be moved at near-memory speeds between the host and the GPU.

This changes the memory's role from fast temporary storage to one in which it serves as a communications medium. If we add *in situ* processing to that we find that memory begins to take on roles once devoted to processing and storage.

The figure on this page illustrates how this concept would work with a CXL channel, although it's applicable to the other protocols as well. In this diagram, which is strikingly similar to the one on the first page, each server has its own Near Memory (the memory that is directly attached to the server's processor), but just as there was shared storage in the earlier figure, there is shared Far Memory in this later figure. These terms also appear in the prior page's graphic. Near Memory for the GPU is the memory on the GPU's board, but to the server it is Far Memory, since it must be read and written through some sort of network. Likewise, for the GPU, the server's memory is Far Memory, even though to the server it's Near Memory.



As with the figure on this page, a protocol like CXL is used to allow all of the servers to communicate with the shared memory on the right side of the figure.

Not only is this memory shared, but because it is not tied directly to the processor, it can be significantly larger than the largest memory supported by the processor. This approach is gaining a lot of momentum today, since AI techniques and in-memory databases both tend to manage enormous data sets that are often larger than the maximum memory supported by a single processor.

It's still too early to determine exactly where all of these protocols will settle out. It appears that CXL is generating more momentum than the others, but it is mainly being promoted as a means to manage composable memory pools, primarily of DRAM, but occasionally of persistent memory.

Disaggregation

Another benefit of CXL or an alternative is that it allows memory to be added to the list of disaggregated resources managed by software. This helps systems move past an issue that has plagued computer architects since the beginning of computing: how do you balance memory and compute resources? In a system with an uncommitted memory pool, different workloads can be assigned whatever fraction of the memory pool they need flexibly and efficiently, and that memory will be reassigned once the workload's job is complete.

A portion of the shared memory pool in this page's diagram can be dedicated to a task running on any one of the servers. In this way software can manage memory so that any server can run an enormous in-memory database and still have access to a sufficiently large memory.

Today servers and storage are managed this way, and future systems are expected to include disaggregated memory, but communication between the memory and the server should have a latency many orders of magnitude faster than that of storage, and this creates a conundrum. By its nature, shared memory must communicate over a network, and that network will add considerable latency to shared memory accesses. Will shared memory be a satisfactory solution?

If the shared memory approach catches on, then new tools will doubtlessly be developed to make it work well. Most likely something like caching will evolve, where hot data resides in the server's Near Memory (the memory actually within the server) and the cooler data will be allowed to reside in the shared memory, which is Far Memory from the server's perspective.

New disciplines will also evolve both in workload monitoring and in software structure to allow both system administrators and programmers to manage Far Memory allocation for highest performance.

Increasing Numbers of Tiers

This white paper has already named additional tiers in the memory/storage hierarchy that are likely to be added to commonly available computing systems. The pooled Far Memory just mentioned will fit between the server's Near Memory and shared storage. If the server has local storage, the shared Far Memory will have lower latency, so it will fit between Near Memory and local storage.

A second new tier that was mentioned a couple of pages back is the memory included in a PIM, or Processor in Memory. The very tight coupling between the on-chip processor and the memory gives the memory in a PIM chip a speed that is faster than Near Memory yet slower than the slowest SRAM cache. This means that the DRAM in a PIM chip should fit between the L3 processor cache and the Near Memory.

It adds complexity to manage an increasing number of tiers, and that might worry some people, but these techniques are very well understood and have been in use for decades, so they are unlikely to be problematic. Too, cache memory management logic or software tends to be relatively simple once all the policies have been thought through, although the process of determining cache policies can be a very difficult puzzle. With Moore's Law cost reductions, the cost of cache management logic has shrunk to the point that cost should never be used as a reason to reduce, rather than to expand, the number of tiers in the memory/storage hierarchy.

Noticing Trends

By this time the reader will have noticed a very strong trend: Memory and storage do not remain on one side of the communication channel, and the compute function does not remain on the other. Instead, the industry is moving towards a model with combined compute + memory/storage devices peppered everywhere, being dynamically allocated to any task that presently could use the particular resources each device can supply.

Future Data Centers

All of this points in a direction with three attributes:

- The data center's architecture will become less specific and more general, with all resources dynamically allocated
- Bandwidth considerations will bring compute and memory/storage resources together
- Computing and data resources will be scattered through the data center and into the Intelligent Edge

It makes sense, then, for computer architects, programmers, systems administrators, and others in the computing community to consider these trends when developing their roadmaps for future products, or their plans to meet the data center demands of tomorrow.

For those managing data centers, the first step might be to perform a thorough analysis of the current data center's performance running today's workloads. With this in hand, they can evaluate their data center's needs going forward: Can the current configuration keep up with tomorrow's needs and will it scale for the next five years? Will the data center's workload complexion change over the next five years, or will it simply grow along the same lines as today? What will it take to meet those growing demands in the future?

Plan of Action

Objective Analysis advises our clients that the next step is to reevaluate their business partners and determine how well both sides' paths align. Are your suppliers the right ones to get you where you plan to go?

Once a company has analyzed its current data center needs and considered its future direction, it is in a great position to chart out its future relationships, choosing vendors with well-aligned roadmaps. This will eliminate potential bumps in the road before they are encountered, simplifying the company's efforts to achieve its long-term goals.

This white paper was developed in collaboration with Micron.

***Jim Handy**, August 2022*