SUMMARY

The server industry is in the middle of a once-every-20-years shift. What has happened in the past is a good indicator of what is about to happen. Given radical past transitions, it is very likely that many of today’s key technologies probably will fade out of the market by the year 2020.

This paper focuses on server primary system memory and secondary storage technologies spanning the past, present, and future. We begin with an overview of historical server architectures going all the way back to the 1960s. Then we outline the current state of server hardware, software, and how they are used today. We conclude with a description of potentially important technologies upcoming in the next decade.

HISTORY OF SERVER ARCHITECTURE

The start of modern datacenter history begins fifty years ago in 1965 with IBM’s System/360 mainframe server. It introduced the idea that a family of server products could be created with the same instruction set and programming model: software written for one model of servers would run on the other models in the product family. From 1965 through 2014, technology changed a lot. But high-level server systems architecture has had only a few big changes. Here is an overview of what has changed in 50 years, and why those changes are important.

HARDWARE TECHNOLOGIES

Starting in the late 1960s, the computer industry shifted from tubes to transistors, which are tiny building blocks used to create computer chips with different functionality and purpose. Progress in transistor advancements over time has been measured against Moore’s Law. Moore’s Law is an observation which forecasts that year after year, transistor sizes will shrink and the number of transistors that can be put onto a computer chip will multiply.

In the early 1960s, computer chip manufacturers could put one transistor on a chip. Now in 2015, they can put billions of transistors onto a single chip. Computer chip manufacturers have miniaturized things so much, that all those transistors can fit onto a chip that is pretty much the same size as a chip from 50 years ago.
INTEGRATION

The impact of this increased miniaturization has been technology integration. Manufacturers take circuits that were initially separate systems, condense those systems into a set of computer chips, and eventually consolidate those computer chips. They create a set of powerful and flexible building blocks that can be put together on one or more chips.

On the computing side, the effect of integration has been huge. In the early days, mainframe processing “cores” were built out of individual tubes and then transistors. As transistors shrunk, computer designers could put more functionality onto a chip. Eventually, they built their first complete computing core on a single chip. Intel’s 4004 processor was first to market in 1971. But fully integrated compute processors did not show up in servers until many years later.

There are four major parts to a server: compute, memory, network, and storage (Figure 1). They each perform separate, different, distinct functions in running applications.

- **Compute** is the hardware logic that runs applications - we call these independent logic units that run applications “cores,” and a processor may contain one or more cores.
- **Memory** is where compute keeps applications and the data those applications need while they are running.
- **Storage** is where applications and data are kept while they are not actively being run.
- **Network** is a server’s connection to other servers and the outside world.

**Figure 1: Major Parts of a Server**
Figure 2 is a generic diagram of some chips before integration. It shows the complex ways the individual chips connect together. Before integration, each chip had a separate interface (or external connections) to other parts of the server (shown with black arrows). The labels on the boxes are less important than the fact that most of the boxes are separate from each other.

**Figure 2: Late 1980s Separate Chips**
All these separate chips and connections were inefficient. So manufacturers started integrating chips together (combining multiple chips into one chip). After integration (Figure 3), all of these chips are combined into hubs. Each hub shares an interface (external connection) to the other chips. Again, the labels are relatively unimportant. What’s important are the blue boxes. They represent a step toward integration.

**FIGURE 3: 1990S X86 INTEGRATION**

Between the late 1980s and late 1990s, integration completely changed server chips. Three major integration hubs emerged:

- The **processor** chip,
- The “**northbridge**” chip, and
- The “**southbridge**” chip.
More recently, server chips are transitioning to “System-on-Chip” (SoC) designs. An SoC takes integration even further by combining the functionality of the three hubs (processor, northbridge, and southbridge) onto one chip. Figure 4 shows fewer blue boxes with more things inside them: more integration. And more integration means fewer lines connecting everything: more efficiency.

**Figure 4: Current Server SoC and Package Integration**
By 2017, products that integrate multiple chips in the same package, including system memory and compute offload accelerators, will be available. Figure 5 suggests even more functions will be integrated together.

**FIGURE 5: POSSIBLE FUTURE OF SERVER SOC INTEGRATION**

It is important to remember that both memory and storage connections do not directly connect memory or storage to a core. Data from both memory and storage travel through at least one memory controller or I/O hub before this data reaches a core. This means that neither memory nor storage communicate directly with a core.
Memory vs. Storage

Memory and storage have been at the center of integration throughout this evolution. In the early years, memory solutions and storage solutions were fairly simple and easy to tell apart (Figure 6).

**Figure 6: Mainframe Data Pyramid, circa 1965**

- **Discrete Registers**
- **Primary**
  - Immediately Available
    - Working Memory
    - Core Memory
- **Secondary**
  - Short-Term Readily Available
    - Online Storage
    - Drum moving to HDD
- **Tertiary**
  - Long-Term Permanent
    - Offline Storage
    - Paper Tape moving to Magnetic Tape
To understand how technology evolution and integration have affected server system performance, Figure 7 illustrates what various parts of current memory and storage systems do, starting with the current solution stack.

**FIGURE 7: UPDATED DATA PYRAMID**

Registers

Registers are part of the processor’s core instruction set. They are at the center of how a core runs the instructions that make up application software. Today, registers are buried deeply inside processor cores.

**Primary Memory**

Processors store the data they need to run in primary memory. The processor’s direct access to primary memory is called “random access memory” (RAM). Memory access is “random”, because a processor can read or write data or application instructions out of any part of it at any time. For servers and most other computers, primary memory is called “temporary” because the processor is constantly moving applications, and data each application needs, into and out of primary memory. This temporary quality is not dependent on specific memory technologies. The ability to access memory randomly to run an application and for the application to access its data is built into computer technology at a fundamental level.
• **Primary System Memory** To keep up with processor speeds, computer and server memory technologies historically have used “volatile” memory. In this context, “volatile” means data fades when power turns off—an acceptable trade for faster speed. In modern systems, “dynamic random access memory” (DRAM) cells have to be continually refreshed or their data fades. Over the decades, primary system memory has evolved its standard interfaces, so that memory suppliers and system vendors both can enjoy economies of scale. Today, most servers use “dual in-line memory modules” (DIMMs) based on connection standards called DDR3 and DDR4 (“double data rate” versions 3 and 4) for primary system memory. There are specialized kinds of DIMMs for different purposes. Figure 8 is an example of a DIMM, and Figure 9 is an example of a server system with several DIMMs installed. Primary system memory (DRAM) is required by server systems to boot normally and run, as server hypervisors, OSes, and applications are designed to run out of RAM memory.

• **Cache Memory** was invented because system memory historically has operated much slower than the processor itself. Cache sits between processor registers and system memory. Cache is more expensive and faster to read from and write to than main memory. Therefore, there is typically not very much of it. As the gap has increased between memory speed and compute speed, cache has evolved more layers to keep up without dramatically increasing costs. Each layer of cache becomes larger and less expensive down the hierarchy (L2 is bigger and cheaper than L1, L3 is bigger and cheaper than L2; etc.). Application data and instructions are loaded into cache as needed. The most-frequently-used parts of an application and their data are kept in cache for best performance. Today, many levels of (sometimes very large) primary memory caches are integrated onto processor chips and sometimes into northbridges as well.

**Figure 8: DIMM (Dual In-Line Memory Module)**
Secondary Storage

A computer stores applications and data when they are not being run in secondary storage. Secondary storage is “persistent” which means application instructions and data kept in secondary storage will not fade; they will always be there (except during failures, etc.). “Non-volatile” is the technical term for persistent. Secondary storage is organized using formats like “block”, “file”, and “object”. The exact definitions of these formats are not relevant here, except to say that secondary storage is not randomly accessible at a level that a running application can use. Cores cannot run applications directly from secondary storage. Applications and data in secondary storage must be written into primary memory before they can be processed. Methods for attaching secondary storage to servers include…

- **Direct Attach Storage (DAS)** devices, such as hard disk drives (HDD) and solid state drives (SSD), load data directly into primary memory and basic input/output system (BIOS) memory. DAS is directly attached to a northbridge or SoC, mostly using standard storage connections. Some DAS manufacturers now integrate network interfaces (like Ethernet) with their HDDs and SSDs instead of storage interfaces (like SAS, SATA, or traditional storage standards). Figure 10 shows a storage server motherboard with both HDDs and SSDs, as well as DIMMs installed. Secondary storage is not required by a server to boot an OS and run,
as at the time a server boots a hypervisor, OS, and applications may be loaded over a network from secondary or tertiary storage on a different system.

- **Networked Storage** systems consolidate DAS into a single storage system attached to the server via the network. The two primary types of network storage systems are “network attached storage” (NAS) and “storage area networks” (SAN). As the name “networked storage” implies, applications and data have to be loaded from and stored to a location elsewhere on a network. NAS and SAN usually have a single network attach point to the server where all data flows into and out of. This single network attach point tends to limit NAS and SAN performance and can become a single-point-of-failure.

- **Distributed Storage** is a newer form of network-based storage where secondary storage systems are dispersed throughout a datacenter. Distributed storage systems do not have a single interface to access data through. They are accessible to any servers in their vicinity. This accessibility avoids SAN and NAS weak points. It also reduces the number of networks data needs to cross to reach primary memory.

**Figure 10: A Storage Server Motherboard with Many DAS Devices**

![A Storage Server Motherboard with Many DAS Devices](image)
Tertiary Storage

Tertiary Storage is long-term persistent storage. Applications and data in tertiary storage typically are written into some kind of secondary storage before they can be used; if not, they must be written into primary memory.

- **Archive Storage** Historically, tertiary storage was called “offline”. Punch cards, paper tape, and magnetic tape took a lot of time for a person to mount them on a reader and have their data sequentially read into an online secondary storage system. We now have robotically managed libraries of magnetic tapes to archive large amounts of data. These are datasets that are too big to move conveniently over networks and too expensive to store on secondary storage systems. Archive storage often contains data that will be used infrequently-to-never (for regulatory compliance, etc.). Trends such as Big Data analytics and increased use of Business Intelligence (BI) are driving long term storage away from offline technologies and toward cold storage.

- **Cold Storage** has emerged recently. It is based on low-cost enterprise-quality HDD technology. HDDs remain off until they are needed, then they are rapidly turned on for data reads and writes, and then they are shut off again after they have not been used for a period of time. This model eliminates physically moving physical objects to mount them on a reader. Cold storage saves a lot of time versus offline storage. For many applications (Big Data, social media, and increasingly enterprise BI) cold storage is now cheaper than archive storage. So HDDs have become a replacement for tape. HDDs also solve a long-term problem with tape storage: when tape drive technology becomes obsolete, tapes can no longer be read and are lost to history. HDDs have highly-standardized interfaces that are likely to be accessible for a much longer time. In addition to HDDs, optical technologies like Blu-Ray and DVD are being considered as potential low-cost high-capacity solutions for cold storage.
SOFTWARE TECHNOLOGIES

The ways that operating systems (OSes) and software support memory and storage have evolved very little in the past few decades. OSes use a number of different models to support them. Several are described in this section.

Physical Memory Model

Using this model, the physical memory present in the system is all the software has to work with (remember from a prior section that this is DRAM).

**Figure 11: Physical Memory Model**

- All primary system memory is directly attached to the processor and is assumed to have the same electrical and performance characteristics.
- If the memory characteristics are not identical, then all primary memory is configured by the operating system (OS) to the lowest-common-denominator of features and performance.
- Software programmers work hard to keep their application software’s memory requirements within physical memory constraints.
- Many embedded systems still use a physical memory model, as they do not have a secondary storage system available.
- **Hardware Implications** In a modern implementation of a physical memory model, the memory controller is integrated into the processor, and data is only moved on a memory bus.
Virtual Memory Model

Using this model, the OS “tricks” software applications into thinking that they have much more memory than the physical memory actually present in the server.

This model makes the same first two assumptions as physical memory regarding electrical and performance characteristics.

When an application asks for more memory from the memory controller than is physically present in DRAM, the data in a portion of physical memory (DRAM) is sent to secondary storage, usually DAS, such as an HDD or SSD. When this occurs, that physical memory now can be freed up and reused by the software application (the exchange is called a “swap”). When that data is needed again, another portion of physical memory is swapped out to secondary storage, and the original data is swapped back into memory. Applications and data swapped to secondary storage are not accessible by the processor; that portion of memory must be swapped back into primary memory to be accessible to the processor. An OS typically can specify how much secondary storage to allocate to its “swap space”, so applications still will be limited on the amount of memory they can access. However, even with this limitation on secondary storage allocation, the virtual memory model provides a much bigger usable memory space than a physical memory model.

Hardware Implications In most server systems sold today, a southbridge has not yet been integrated into the processor. When a “swap” occurs, data is transferred across a dedicated memory bus, then across a shared I/O bus (like PCI or PCI-Express) to a southbridge and from there across a dedicated storage channel to DAS (like SATA or SAS). When a block is swapped back into memory, the process is reversed. At no point does a processor’s memory controller directly access data in the DAS (storage) swap space as it would DRAM physical memory.
Non-Uniform Memory Access (NUMA) Memory Model

A group of processors running the same OS image share their local primary system memories with each other. It is called “non-uniform” because memory accesses now can take different amounts of time depending on which processor the memory is connected to in the system.

**FIGURE 13: NUMA MEMORY MODEL**

- The NUMA memory model builds on top of the virtual memory model.
- Two processors directly connected in the same system can request direct access to each other’s memory through linked memory controllers.
- **Hardware implications** In addition to the data transfers in the standard virtual memory model, memory data may be transferred over a shared or dedicated high speed link between processors’ memory controllers.
RDMA (Remote Direct Memory Access) Memory Model

This model allows processors to have access to remotely-distributed memory.

**Figure 14: RDMA Memory Model**

- The RDMA model builds on top of the NUMA model.
- Two processors with local network access to each other can request direct access to each other’s memory through the network.
- Different systems may have differently-configured physical memory pools. Each system’s OS does not know anything about the other systems’ memory attributes, but they can still share the contents of their memory.
- **Hardware Implications** In addition to the data transfers in the NUMA memory model, memory data may be transferred over networks (like InfiniBand and Ethernet) between servers in a rack and perhaps in different racks of servers.
Virtual Machines (VM) Model

This model virtualizes all of the hardware in a system, not just memory.

- Most large companies today have standardized on virtual machines from VMWare which use the memory models mentioned above.
- Software applications running in virtual machines do not have direct access to any of the hardware. The virtual machine uses software layers to hide the actual hardware. It lets an application "pretend" that it has sole access to shared hardware resources. Software applications running in a virtual machine model are completely unaware of which hardware is being used for their memory and storage requests.

In a purely physical hardware environment (a non-virtualized environment), the OS tells a software application precisely what physical hardware is available (Figure 15).

**Figure 15: A Physical Server, Not Virtualized**

In a virtualized operating environment, software applications run on a "guest OS" inside a virtual machine (VM). The VM tells the guest OS what server resources will be supported—which may be completely different from the physical hardware present. Also, a guest OS believes it has sole access to physical resources. All resource sharing between VMs—including memory and storage resources—is hidden (Figure 16).

**Figure 16: A Virtualized Server**

Virtual memory is used within a VM environment. But virtual memory is only a small part of a fully virtualized server. They are not interchangeable terms or technologies.
Maturing markets fragment as vendors seek to differentiate from one another. We are in a late maturity phase for our current generation of server technologies. Many hardware-based solutions that previously were built by specialty system manufacturers are becoming obsolete as those functions can now be performed by software.

Hardware solutions that can’t easily be moved to software are being converted from full systems to component level solutions that can be integrated into standard server platforms—eliminating the need for specialty systems. This integration is largely possible by using a small number of standard hardware interfaces.

Hardware vendors seeking to differentiate with new storage, network, and compute acceleration options must leverage the system interfaces available to them. Their choice is to pick the option that has the best performance at an acceptable cost. Devices with different functions can use different physical interfaces such as PCI-Express, SATA, SAS, USB, and others. Also, the same physical interface can support different functions. For example, PCI-Express can be used for compute, memory, network, and storage devices.

The three devices in Figure 17 all have different functions, but they all use PCI-Express.

**Figure 17:** A. PCI-EXPRESS PROCESSOR CARD, B. PCI-EXPRESS STORAGE CARD, & C. PCI-EXPRESS NETWORK CARD
CURRENT STATE OF SERVER ARCHITECTURE

Server technology evolution is at a crossover point in 2015-2017. We are in the gap between major architecture generations. This situation is very similar to the leap from mainframes and minicomputers to today’s Unix and x86 servers.

**Figure 18: Server Processor Shipments Over Time**

The next generation of server technologies, called “hyperscale”, will be based largely on the new cloud and network technologies that many customers are adopting to address emerging software applications such as the Internet of Things and Big Data analytics.

Hyperscale computing allows large datacenters to scale their compute, memory, storage, and network resources quickly, efficiently, and cost effectively as their demands increase. Hyperscale hardware is purchased and deployed in large quantities and is designed to be more power efficient and dense than typical enterprise datacenter equipment. Hyperscale computing is often associated with the infrastructure required to run large websites such as Amazon, Facebook, Google, and Microsoft.

Software defined datacenters (SDDC) will drive the next generation of server, storage, and network infrastructure. In a software defined datacenter, all infrastructure elements—network, storage, compute, and security—are virtualized and delivered “as a service”.

**HARDWARE TECHNOLOGIES**

There will be many alternatives for memory and storage access approaches in the market until the industry climbs out of the current transitional phase. Having many options is normal after a market fragments and before the next dominant design provides a stable platform for the next decade or two.

**Application Specific Acceleration and Shared Primary System Memory**

Integration marches on. Server processors are reaching “diminishing returns” limits with general-purpose cores. Performance-per-watt improvements have slowed down, and applications are becoming more specialized. Some applications can be sped up by compute offload or alternative compute architectures. While these accelerators historically have been attached to processors via interfaces like PCIe, they increasingly are being integrated onto system-on-chip (SoC) designs. In future SoC designs, these accelerators will access primary memory directly along with the processor cores. This access can happen across processor on-chip and off-chip interconnects.

Intel, AMD, and IBM each are using direct processor or northbridge links to grant high bandwidth memory access to compute accelerators. Micron took a different approach: it put a compute accelerator, the Micron Automata Processor (AP), directly on the memory bus.

**Figure 19: Micron Automata Processor Attaches to DIMM Connector**
Memory & Storage Access over Networks

PCIe and Ethernet both are being used as DAS interfaces within server chassis as demonstrated by Dell DCS, Dell FX2 systems, HP Moonshot, HP Apollo, and others. These interfaces muddy the definition of DAS, because the storage is no longer directly attached to the northbridge or to the processor SoC; it is a local network hop away.

Not only is DAS becoming more remote, RoCE (RDMA over Converged Ethernet) enables using the network for memory access to other processors (see Figure 14 and the RDMA description above). Data can be moved directly from one processor’s primary memory into another processor’s primary memory across a local network without being written to secondary storage at all. Primary memory and secondary DAS storage are both extending their reach across the network.

Non-Volatile Chip Technologies Move onto Primary Memory Bus

One of the most common technologies used to create non-volatile storage is called a flash chip. Figure 20 shows the familiar enclosure of a “flash drive”, a similar flash drive with the casing removed, and a flash storage chip. Flash chips can use any physical interface or connection in the system, including USB, SATA, SAS, or PCI-Express. Flash chips are even being used on DIMM modules. Whether they use USB, SATA, SAS, PCI-Express, or DIMM modules to connect flash to the system, **flash chips and DRAM do different things**. Flash chips and DRAM cannot substitute for each other.

**Figure 20: A. COMMON “FLASH DRIVE”, B. FLASH DRIVE WITH EXPOSED FLASH STORAGE CHIP, & C. FLASH STORAGE CHIP**

The Storage Networking Industry Association (SNIA) formed a non-volatile DIMM special interest group (NVDIMM SIG) to standardize functionality and create awareness...
of use cases for non-volatile DIMMs. These NVDIMM specifications are not yet completed, are not publicly available yet, and the names of the different NVDIMM specifications are not fully agreed upon, but memory and storage vendors have designed and shipped a first generation of products under the general name of “NVDIMM”. While the purpose of these NVDIMM’s will differ significantly from one another, they allow various new devices to connect to the memory controller.

The first use case to market used NVDIMM technology in battery backed up (BBU) DRAM-based DIMMs in the case of a power failure (Figure 21 and further discussion in Usage Models below). In the BBU DRAM-based DIMM model, immediately after a power failure, the DRAM and flash storage controller write the contents of DRAM to the flash storage using power stored in the battery. The processor does not see the flash storage in a BBU DIMM configuration, so the flash chips on the DIMM are not used by the processor for computing; it is not part of primary system memory. When system power is restored, the data that was backed up in the flash storage is loaded back into DRAM. Thus, the processor can continue with the application it was running as if the power interruption had never happened.

**Figure 21: Battery Backed Up DRAM General Architecture**

SNIA’s NVDIMM SIG is developing a set of specifications that further define the differences between different types of NVDIMM. The first three specifications are NVDIMM-F, NVDIMM-N and a couple of as yet unnamed variants still under development. (For thoroughness, hybrids NVDIMM-N2 and NVDIMM-P also exist.) The ‘F’ refers to “all flash”, and the ‘N’ refers to “NAND” flash (a type of flash technology used in hybrid DRAM and flash modules: the next generation evolution of BBU DRAM technologies). NVDIMM-N is memory and NVDIMM-F is storage.
- **NVDIMM-N modules contain both DRAM and flash chips.** NVDIMM-N is designed to preserve data in DRAM when power is lost, just like BBU DIMMs. The processor does not see the flash storage in an NVDIMM-N configuration, so the flash portion of NVDIMM-N is not used by the processor for computing; it is not part of primary system memory (Figure 22). The DRAM is used and seen by the processor as part of primary system memory, while the flash is not. Immediately after a power failure, the NVDIMM-N controller writes the contents of its DRAM to its flash storage on the DIMM using power stored in capacitors. When system power is restored, the data that was backed up in flash storage is automatically loaded back into DRAM without the processors knowledge or intervention. Thus the processor can continue with the application it was running as if the power interruption had never happened. This new specification will use capacitor technology instead of batteries, because capacitors do not have to be replaced like batteries, and they are friendlier to the environment when disposed. Because NVDIMM-N devices are seen as memory, they can be used in place of DRAM DIMMs (but they are more expensive). There are many memory vendors competing for attention in this market. Although it is a niche market (see Usage Models section below), profit margins are attractive.

**FIGURE 22: NVDIMM-N GENERAL ARCHITECTURE**
- **NVDIMM-F** modules are **entirely flash-based (no DRAM)**. NVDIMM-F devices are not random access (they use a format called “block I/O” only) and are functionally the same as a PCIe connected SSD. They do connect to a DDR3 memory connection (Figure 23), but NVDIMM-F bypasses high levels of memory control, and so it does not appear to the processor as part of primary system memory. Without the DRAM, NVDIMM-F devices will provide very different capabilities than NVDIMM-N devices. DDR3 has higher data transfer rates than PCIe and traditional SAS and SATA storage connections, which enables NVDIMM-F modules to have higher performance than SSD in a wide range of applications. Because NVDIMM-F devices are seen as storage, they compete directly with HDDs and SSDs. NVDIMM-F cannot replace NVDIMM-N and DRAM devices.

**Figure 23: NVDIMM-F General Architecture**

The first (and so far only) released NVDIMM-F products are the Lenovo eXFlash DIMM and SanDisk ULLtraDIMM based on Diablo’s Memory Channel Storage (MCS) architecture. A secondary benefit to locating storage in each processor’s memory space is that each processor can use NVDIMM-F modules as a local virtual memory storage cache, instead of sharing virtual memory storage cache with other processors in a system. In a NUMA design this configuration will directly improve performance by keeping virtual memory data traffic local to each processor’s memory controller, plus it will reduce data traffic through the southbridge storage hub, which also contributes to improved system performance.
• **NVDIMM-unnamed** modules are *entirely flash-based (no DRAM)*. All of the DIMM is seen by the system as part of primary system memory, and therefore it all is used directly by the processor and applications running on the processor. NVDIMM-unnamed is designed to fit into DRAM-based DIMM connectors (Figure 24). With newer flash chip technologies, data can be read out of NVDIMM-unnamed at or near DRAM-based DIMM speeds. However, writing data into an NVDIMM-unnamed is much slower, so usage models will be somewhat different than a standard DRAM-based DIMM. Entirely flash-based NVDIMMs will need to use a technique called “execute-in-place”. Execute-in-place refers to running an application directly from non-volatile memory rather than copying into random access primary system memory. Execute-in-place also allows an application to use data that is stored in non-volatile memory—data that the application does not expect to change or that minimal changes will occur during processing. Embedded systems use this technique today, but it is not currently used in servers. No one has entered the market with this type of NVDIMM product yet.

**Figure 24: NVDIMM-unnamed General Architecture**

• **NVDIMM-N2** or **NVDIMM-P** modules will have the same components as NVIDIA-N modules, but they have more flash chips than are needed to back up the DRAM on the DIMM. The extra flash chips will be accessible to the system as a block-oriented storage device—they will behave like an SSD—and so this type of device extends NVDIMM-N to also behave like NVDIMM-F, albeit with a large capacitor attached. The system will separately use this DIMM’s DRAM as memory and the extra flash chip capacity as storage. We ignore these future hybrid devices for the rest of the paper because their behavior will be just like either NVDIMM-N or NVDIMM-F depending on which function a system is using.
SOFTWARE TECHNOLOGIES

Over the next couple of years, new primary memory system hardware technology will not demand huge changes from software.

Solid State Drives (SSD) will replace Hard Disk Drives (HDD)

SSDs eventually will replace HDDs for secondary storage, but OS memory model changes will not be required to support them. SSDs are newer, higher performance, lower power replacements for HDDs. HDD performance has been reaching diminishing returns limits for several years. SSD is not architecturally different than HDD, so the major server OS vendors like Microsoft, Linux, and VMware will not be required to alter their memory support models as this transition occurs.

Modern operating systems have been written to a model that assumes:

- System memory does not retain its state when powered off, and
- System memory is a lot more expensive than mass storage per bit stored.

Both of these assumptions are still true with SSDs. Primary system memory is treated as a scarce resource. SSD-based secondary storage can improve system performance over HDD, but SSDs do not enable new architecture. However, SSDs are not the same thing as non-volatile primary system memory. SSDs, like HDDs, store data as blocks, files, or objects. They are not suitable for the kind of random access required to act as primary system memory. Because SSDs tend to have better performance than HDDs, they are useful for hosting virtual memory swap spaces and as fast cache for larger capacity but slower HDDs.

Impact of NVDIMM on the Operating System

Each of the different NVDIMM technologies will be handled differently by the OS.

- **NVDIMM-N** devices do not require any changes to an OS, but they do require changes to BIOS so that a server can restore its memory properly during a reboot. Because NVDIMMs are built with DRAM, they behave as normal DRAM-based DIMMs. The OS sees and NVDIMM-N devices as memory.
- **NVDIMM-F** devices are not part of primary system memory and do not require changes to an OS, but they do require installing a driver that knows how to talk to block I/O flash devices located on memory connections. The OS sees and treats NVDIMM-F devices as storage.
• **NVDIMM-unnamed** devices have the biggest potential impact on their OS. To support them near term, software programmers will borrow a method used by embedded products. Many products (printers, scanners, smartphones, *etc.*) cannot afford the cost or space to use secondary storage. They design their primary system memory with a combination of read-only memory and DRAM to run their OS and/or applications without loading them into the DRAM portion. This method is called execute-in-place. Manufacturers pre-load the OS and software applications directly into a ROM or flash chip that is part of primary system memory and then run their OS or applications out of that read-only memory. DRAM is used for portions of the OS and applications where data changes (ROM) or changes frequently (flash). It takes a specialized set of software design skills to program for this type of system; most application developers assume their application will run entirely in DRAM. The OS sees and treats NVDIMM-unnamed devices as memory.

It is important to note the differences in functionality between the above NVDIMM options at the server level. DRAM primary memory is used to execute instructions from an OS and applications. Servers cannot run on secondary storage alone, because instructions cannot be executed directly out of storage. Because of this, NVDIMM-N devices can be used in a server without additional DRAM present. NVDIMM-F devices must be accompanied by DRAM and also perhaps NVDIMM-N in order to operate.

**Impact of NVDIMM on the BIOS**

Most modern OSes are designed so that a small “basic input-output system” (BIOS) is the first application that a server runs when all of its components are powered on. BIOS takes inventory inside the server. This inventory (along with some predetermined settings) determines the information that the BIOS gives to the OS (Windows, Linux, Unix, *etc*.). The OS uses this information to understand what hardware features are available. Figure 25 outlines how this procedure works step-by-step.

Features that BIOS identifies include the major parts of the server: compute, memory, network, and storage. Among other things, BIOS determines which DIMMs are built for memory and which DIMMs are built for storage (such as Diablo’s Memory Channel Storage-based DIMMs). BIOS recognizes the basic difference between them (hence, “basic input-output system”). BIOS tells the OS which parts are NVDIMM-F and which parts are standard DRAM DIMMs. After BIOS has taken inventory and informed the OS, then the OS takes over, and BIOS shuts itself down.
BIOS has to be aware of each memory DIMM’s capabilities.

- **DRAM** is assumed: it has fast reads and fast writes.
- **NVDIMM-N** has DRAM speed reads and writes, so it is functionally identical to DRAM. Specialized BIOS code is required for the OS to understand that applications and data may already be available in DRAM memory after booting, so processing can begin where it left off after a power failure.
- **NVDIMM-F** has slower performance than DRAM memory, but fast performance compared to other storage solutions. It is functionally very different from DRAM. NVDIMM-F requires BIOS or the OS to load a block I/O driver so that it will be seen as a storage device.
- **NVDIMM-unnamed** has DRAM speed reads, but is slower for writes. In different environments it may be treated as ROM, or it may be devoted to an application or data that is not expected to change frequently. Execute-in-place would be used by the OS to support NVDIMM-unnamed products.
- New or customized functionality (like the “write-only” model that Micron’s Automata Processor uses).
**Usage Models**

Below are some of the key storage and memory technologies usage models expected over the next several years.

**NVDIMM-F and SSDs** are starting to see use as caches in a wide range of applications for secondary storage, where HDDs are not fast enough. SSD is also starting to see use as cache for tertiary cold storage. As stated above, SSD will not see use as primary system memory, and NVDIMM-F is no different than an SSD or HDD.

**Figure 26: Competing Memory Technologies: Relative Performance**

---

**Non-Volatile Memory** will see more use as primary system memory and to enable application acceleration. For the first time, mainstream random access primary memory will not be equal in terms of functionality. Over the next couple of years, flash chips will enter primary system memory for read-only or light writing applications and will enable application specific acceleration via write-only memory channels (like Micron’s Automata Processor).

**BBU and NVDIMM-N** The use of BBU DIMMs in servers is a very small niche in a large memory market. Typical applications have been in embedded applications like automated teller machines (ATMs) and casino gaming, where regulatory compliance mandates zero data is lost. In these specific cases, the cost of BBU DIMMs can be justified by a higher price for the system. BBU DIMMs also have seen use in enterprise
secondary storage RAID caches. In these RAID cache use cases, BBU DIMMs are used when it is important to preserve the contents of data that has not yet been written to disk. Memory vendors believe that lower cost and more reliable NVDIMM-N modules will have a larger demand than BBU DIMMs.

**NVDIMM-unnamed** Applications that need to do a lot of processing on chunks of data that will not change (or remain “static”) while the application analyzes them are a good fit for NVDIMM-unnamed. Software applications that are good targets for NVDIMM-unnamed generally do so much processing on the data after it is loaded into primary system memory that they don’t care about the slow write speeds of flash in NVDIMM-unnamed. These software applications generally have large datasets that require DRAM like read speeds at lower cost per bit than DRAM. Based on this criteria, these software applications may be a good fit for NVDIMM-unnamed.

- Online Transaction Processing (OLTP) and database processing, such as credit card processing and airline reservations systems.
- Real-time analytics and “Big Data,” such as financial fraud detection and risk analysis, social media filtering and pattern recognition, and emerging Internet of Things data mining.

**Figure 27: Competing Storage Technologies: Relative Performance**
FUTURE DATACENTER ARCHITECTURE DIRECTIONS

Computer hardware evolves at a faster pace than software. Hardware is driven by physics and materials. Software is dependent on people writing code. Part of the reason why gaps between datacenter technology generations are so long is people. Software programmers have to integrate a new wave of hardware capabilities, and that takes time. So, if new hardware entering the market behaves just like old hardware and can be used for existing usage models and applications (like SSD vs. HDD), then it will have an advantage. It will be successful sooner than hardware that requires changes to OS software (like NVDIMM in its early stages). And it will be successful much sooner than hardware that requires changes to applications software (like web services written to future mixed capability memory pools).

FIGURE 28: POTENTIAL FUTURE DATA PYRAMID
**Hardware Technologies**

Integration Continues: Packaging Multiple Chips

The expected trend for processor vendors is to integrate more memory and more application accelerators into the processor package. For example, Micron has developed Hybrid Memory Cube (HMC) technology. HMC connects logic chips and memory chips in a single package, one on top of each other, so they don’t take up more board space than a single chip, they are just taller. It’s not a stretch to envision Micron integrating their Automata Processor with primary system memory in a single package.

When memory is stacked with logic, the package’s internal memory connection has to change from today’s standard interface of DDRx DIMM connectors. This change will require a custom or semi-custom interface. There will still be an external memory connection, but it is unlikely to look like today’s standard DIMM connectors.

Rack Scale Architecture

Intel RSA is Intel’s attempt to provide value beyond the processor by enabling a solution at the server rack-level. The goal for RSA is to replace traditional server architectures with pooled and disaggregated computing, network, and storage resources. RSA uses modular hardware that can be swapped out as needed, which builds on the concept of VMs described above. Intel plans for RSA to separate processor from direct attach primary system memory later in the decade—which will need some of that OS magic we talked about above. RSA is also important because it will require completely different memory interface technology than today’s DDRx memory interface.

Non-Volatile Memories

New non-volatile memory technologies have the potential to impact the storage pyramid dramatically over the next decade. There still will be a set of memory hierarchies, but they will be based on memory functionality, bandwidth, and latency—not solely on differences in persistence and speed.

HP has a new system architecture concept they call The Machine. It is based on processors using special purpose cores, high speed communications links, and a non-volatile memory design to provide memory with the speed of DRAM but the capability of permanent data storage. HP expects this technology to span all the way from end points (mobile devices, IoT) to datacenters (servers, network, and storage equipment).
SOFTWARE TECHNOLOGIES

Mainstream operating systems historically have been written to a model that assumes these things about system memory:

- Processors require randomly accessible, byte addressable memory.
- All of system memory has the same performance level for how fast it can move data and how quickly it responds to requests for data.
- System memory does not retain its state when powered off.

System memory is a lot more expensive than mass storage per bit stored. There is a challenge moving forward with non-volatile memories in primary system memory: a lot of new software infrastructure must be created.

- New operating memory models for handling combined memory and storage and for dynamically composing pools of memory
- OS kernel modifications or replacement (requiring significant investment, time, and resources)
- New programming structures for software applications to take advantage of different primary system memory types
- Automated translation of existing software applications into the new programming and runtime models

To enable a heterogeneous memory universe, HP has invested in Machine OS (codenamed “Carbon”) as the key enabler for The Machine. Carbon offers a new OS concept that combines new memory and storage into one hierarchy—instead of separate entities like today’s OSes.

USAGE MODELS

Past the end of the decade, many tiers of primary systems memory will evolve. When the primary system memory pool becomes non-volatile, then it may merge with SSD-based DAS storage. And farther down the pyramid, when SSDs cross the cost curve with HDDs, they will be more widespread in secondary storage usage; then they may go into some tertiary storage applications. As standard server interconnects consolidate and networks become faster, the combination will enable secondary storage—and even future primary system memory—to be distributed across more distant networks. New memory architectures will not be function specific. As we approach the end of the decade, there will be no limits to non-volatile memory usage models.
CONCLUSION

Over the last 50 years, advancements in technology have allowed for servers to transition from large complicated systems made up of a large number of chips with customized connections and interfaces, to servers that today include a few chips integrated with a standard set of interfaces. While hardware has evolved at a relatively rapid pace, software takes more time to catch up to major technology advancements.

As new hardware technologies come to market, it is important to understand which will impact operating systems and software applications, so we can determine how quickly they can be accepted and used in the market. In looking at the memory and storage pyramid as an example, it is critical to separate the type of technology used from where it sits on the pyramid and how it connects to the rest of the pyramid, so we can evaluate its impact on software and determine its potential for use in other areas of the pyramid.

The industry megatrend toward Big Data will require analyzing ever-larger datasets at increasing speed, so that business and even personal insight is available exactly when we need it. New NVDIMM technologies, such as Diablo’s MCS, allow larger chunks of data to move closer to the processor. These new NVDIMM technologies enable processors to bring more data into each processor's memory and to speed up server system performance. NVDIMM is in many ways preparing server architecture for future memory chip technologies.

Lines continue to blur between buses, networks, and memory technologies. Many will not survive the upcoming server market transition. But there is much opportunity for new technologies as we enter the next datacenter era.
**Glossary**

**AP** - Automata Processor, an application accelerator that uses a DDR3 memory channel for data transfer with a processor, designed by Micron, Inc.

**Application** - a set of programs that runs on a server or other computer.

**BBU** - Battery Back Up, uses a battery to save volatile DRAM memory to non-volatile Flash storage when power goes out.

**BIOS** - Basic Input Output System, when a system powers up it loads the OS so that the computer can wake up & run applications.

**Cache** - a data resting place between faster & slower devices that makes some data available to a faster device while a slower device fetches more.

**Capacitors** - a passive electrical component used to store energy electrostatically. Capacitors are widely used as parts of electrical circuits in many common electrical devices.

**Compute** - the logic that runs applications, or the act of running an application.

**Core** - an independent unit of compute resources defined by its instruction set.

**DAS** - Direct Attach Storage, any secondary storage that communicates directly with a processor or a northbridge using one of the standard connections, such as SATA or SAS.

**Data** - information that applications use to figure things out & interact with people & the real world.

**DDR, DDR3, DDR4, DDRx** - Dual Data Rate & then a generation number, or an x to say it might be any generation; it is the most common interface between DIMMs & a northbridge or a processor.

**DIMM** - Dual Inline Memory Module, a very old name for a small memory board containing memory chips, such as DRAM or flash chips.

**DMA** - Direct Memory Access, when another chip in a computer can access primary system memory without using a processor to do so, by only talking to a northbridge or a memory controller.

**DRAM** - Dynamic RAM, a RAM chip that loses its data when power goes off; some types of DRAM need periodic power refreshes to keep data in memory.

**Embedded System** - a computer-run device that people cannot load applications onto, or a device that people don’t know has a computer in it, such as a printer, thermostat, car engine, automated teller machine, etc.

**Ethernet** - a very common form of computer network.

**Execute in Place** - when an application is stored in a persistent memory & the processor runs it directly from that memory without first moving it into DRAM.

**Flash** - a type of non-volatile memory chip.

**HA** - High Availability, a type of computer architecture that is designed to recover from most or all possible failures.

**HDD** - Hard Disk Drive, a spinning platter of metal or ceramic coated in magnetic material that stores data.

**HMC** - Hybrid Memory Cube, a form of packaging a stack of processor & memory chips, designed by Micron, Inc.

**Hyperscale** - a style of computing often associated with the infrastructure required to run large websites such as Amazon, Facebook, Google, & Microsoft.

**Hypervisor** - is an OS-like software layer that can run lots of Virtual Machines (each with its own OS) on one set of physical hardware, mostly by telling each of the VMs what hardware they should have access to.

**I/O** - Input Output, data communications between chips and/or systems.

**InfiniBand** - a network like Ethernet, but faster & more expensive & used for higher performance applications.

**JEDEC** - Joint Electron Device Engineering Council, the organization that standardizes DDRx memory interfaces.

**L1, L2, L3** - levels of primary system memory cache, these days it is integrated into processor chips.

**Linux** - a frequently used server OS, especially in hyperscale data centers.

**Memory** - where applications & the data those applications need while running are kept when the applications are running.

**Memory Controller** - controls access to memory on a board, such as a motherboard or a DIMM; it translates memory requests from many communications channels into a form that memory can understand.

**Motherboard** - circuit board with the processor or several processors on it; the main board in a computer.

**NAS** - Network Attached Storage, a form of secondary storage that attaches all of the storage drives to a single network interface, similar to...
Network - a server’s connection to other servers & the outside world
Non-Volatile - memory that does not fade or lose its data when power is turned off, we also use “persistent” to mean the same thing
Northbridge - a chip that communicates directly with a processor & manages the processor’s data flow, it contains other system bus interfaces & the primary system memory controller
NUMA - Non-Uniform Memory Access, a way to divide up memory between processors so they can each share memory with the others; usually on one motherboard, but sometimes in the same system
NVDIMM - Non-Volatile DIMM, a DIMM that won’t lose data when power is turned off
OS - Operating System, a layer of software between applications & hardware that lets applications programmers not worry about exactly what hardware the application is running on, the OS hides detail & complexity
PCI - Peripheral Connect Interface, was used to attach smaller boards to a motherboard to add new features & functions, first shipped 20 years ago
PCIe - PCI-Express, a newer form of PCI that first shipped 10 years ago
Persistent - memory that does not fade or lose its data when power is turned off, we also use “non-volatile” to mean the same thing
Processor - the central part of a computer that runs the applications
Rack - a shelving unit for servers, about two feet wide, four feet deep, & six feet high
RAID - Redundant Array of Independent Disks where multiple disk drive components are combined into a logical unit for the purposes of data redundancy or performance improvement
RAM - Random Access Memory, a form of memory chip that data can be read from or written to any location, in any order
RDMA - Remote DMA, when DMA is done between systems connected over a network
RoCE - RDMA over Converged Ethernet, ignore the word “converged” & it simply means a well-defined method for doing RDMA memory sharing over Ethernet
RSA - Rack Scale Architecture, an Intel concept for a network connected server that can grow to fill a floor-to-ceiling rack of servers
SAN - Storage Area Network, a form of secondary storage that attaches all of the storage drives to a single network, similar to putting all of the drives in one building with several doors
SAS - Serial Attached SCSI (SCSI was an early storage connection used by Apple), a storage connection between one computer & one drive
SATA - Serial ATA (ATA was an early storage connection used by PCs), a storage connection between one computer & one drive
SDDC - Software Defined Data Center, a new hyperscale data center architecture where software hides the hardware from multiple OSes & forms virtual machines from the hardware components it needs
Server - a form of computer used by data centers, usually with one to four processors & taking up a few inches of rack height
SIG - Special Interest Group, a flock of geeks in an industry association, committee, consortium, etc.
SNIA - Storage Networking Industry Association, an industry association of storage manufacturers, customers & others interested in building storage products that work with each other & other data center products
Southbridge - a collection of I/O & other functions that are grouped together to save board space & chip costs, it connects to a northbridge so that it doesn’t slow down the processor
SSD - Solid State Disk, a module containing non-volatile memory chips, such as flash chips, built to act just like a HDD
Storage - where applications & the data applications need are kept when they are not in memory & running
TPM - Trusted Platform Module, a security device that knows when hardware in a system has been changed & has other security oriented functions
Unix - an older server OS mostly used in company data centers; it has been almost completely replaced by Linux
VM - Virtual Machine, an imaginary computer created by a hypervisor for a guest OS, in which an application running on the guest OS thinks it has only the memory & storage given to it by the hypervisor
Volatile - the data disappears when power is turned off
x86 - the type of Intel processor used in the original IBM PC & is now in every computer & most every server