NANTERO

White Paper

Memory Class Storage for Current and Future Servers

Memory Class Storage is Permanently Changing Server Architectures

By Bill Gervasi

Abstract: The industry is preparing for another inflection point as non-volatile memories continue to move closer to the system memory controller. Initially on the I/O channel as SSDs then optimized as NVMes, subsequently they were inserted into the memory channel in the form of NVDIMMs which brought storage class memory to the architecture. The next logical transition is non-volatile memories with full DRAM capabilities, or memory class storage. Nantero is a non-volatile memory technology based on carbon nanotubes that is capable of directly replacing DRAM. NRAM technology may be applied advantageously to vastly improve the quality and efficiency of distributed processing in a server environment.

About the Author: Mr. Gervasi is Principal Systems Architect at Nantero, Inc. He has been working with memory devices and subsystems since 1Kb DRAM and EPROM were the leading edge of technology. He has been a JEDEC chairman since 1996 and responsible for key introductions including DDR SDRAM, the integrated Registering Clock Driver and RDIMM architecture, the formation of the JEDEC committee on SSDs, and actively involved in the definition of NVDIMM protocols. He is chairman of the JEDEC emerging memories committee.

Introduction

For decades, computer memory hierarchies have been designed around the limitations of DRAM. When power fails, massive amounts of data are lost permanently. As a result, systems have employed sophisticated methods to checkpoint essential data, adding noticeably to systems cost and complexity. Over time this checkpoint storage has moved closer to the CPU, from hard drives to SSDs and NVMes to storage class memory on the DRAM channel, but each transition still required massive data movement for this checkpointing which consumes significant system time and burns a phenomenal amount of power.

Finally, this is about to change. Memory class storage (MCS) is an emerging non-volatile technology poised to replace DRAM as the primary storage for applications and data. MCS has the speed of DRAM coupled with data persistence to retain all data in the case of power failure or other system glitches. NRAM® is Nantero's entry into MCS, offering the world's first non-volatile DRAM replacement technology.

At first glance this change seems more evolutionary than revolutionary, however the changes in system architecture enabled by MCS enable new ways to envision server systems including the elimination of external storage. Fabric computing and artificial intelligence solutions can also achieve orders of magnitude improvement from a high performance persistent memory option.

NRAM also improves server applications by offering lower power, higher performance, lower cost, and a plan for device densities up to 16 times the DRAM roadmaps.

NVRAM: Memory Class Storage

What if DRAM could be replaced with a non-volatile (persistent) memory alternative? Since 1971, this was not a question taken very seriously. That is finally about to change.

A revolutionary new standard in development is the NVRAM device specification which enables a new class of non-volatile memories called Memory Class Storage (MCS)[1]. These devices operate as direct drop-in replacements for DRAM which implies that they must be fully deterministic. Every read or write command must be handled by an MCS device in essentially the same number of clocks that a DRAM would do the same operation. In order to achieve determinism, MCS devices must have no wear-out characteristics that would impact DRAM controller timing.

[1] For now, Memory Class Storage and NVRAM are interchangeable terms, however it is likely that future MCS interfaces will break away from simple DRAM superset protocols and establish newer optimized interfaces to exploit all non-volatile features. NRAM® is Nantero's trademarked implementation of NVRAM.

The DDR5 NVRAM specification in development in JEDEC is an annex to the DDR5 SDRAM specification. The DDR5 NVRAM specification documents the compatibility with DRAM, and also any additional features of these MCS devices, such as the ability to eliminate Refresh commands, Precharge commands, and to leave pages open indefinitely.

With DDR5 NVRAM, a DRAM replacement memory module can be constructed using any of the standard memory module variations such as unbuffered (UDIMMs and SODIMMs), registered (RDIMMs), load reduced (LRDIMMs), multiplexed rank (MRDIMM), or differential interface modules (DDIMMs). No sideband signals are required, therefore DDR5 NVRAM modules are plug and play in an unmodified DRAM channel.

Data Tiers, Performance vs Capacity

For decades computer architectures have relied on a tier of storage elements from processor registers to local scratch pads, from on-chip caches to separate caches. The memory channel has typically provided temporary storage for von Neumann style processors, and on a separate I/O channel were persistent storage devices such as hard drives or tape backups. Each of these tiers tended to be slower and have greater capacity than the tier above. Figure 1 is a common expression of this hierarchy.

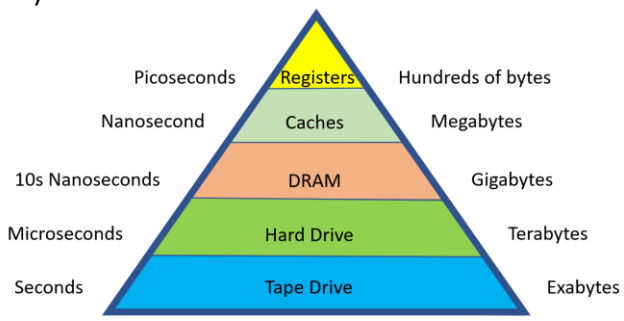

Figure 1: Traditional Memory Hierarchy

This model has been consistent for so many decades that it is almost redundant to copy it here, except that it does serve as a reference point for the migration of non-volatile memory (NVM) into the hierarchy initially as solid state drives (SSDs) on standard I/O ports like SATA or SCSI, then later to higher performance variations such as non-volatile memory express (NVMe), but it still remained outside the directly addressed space of the main processor as an I/O resource.

HOW THE TRADITIONAL MODEL DRIVES ACCESS MODES

This tiered structure led to two fundamentally distinct methods for accessing data: direct byte-oriented access, and block oriented file access, each with its unique semantics for allocating and accessing, as well as vastly different performance profiles. Fundamentally, every time an access must go through the file access mechanism, it is significantly slowed by the overhead of context

switching through the operating system. Figure 2 shows the distinction between these access mechanisms and some of the programmer functions that distinguish them.

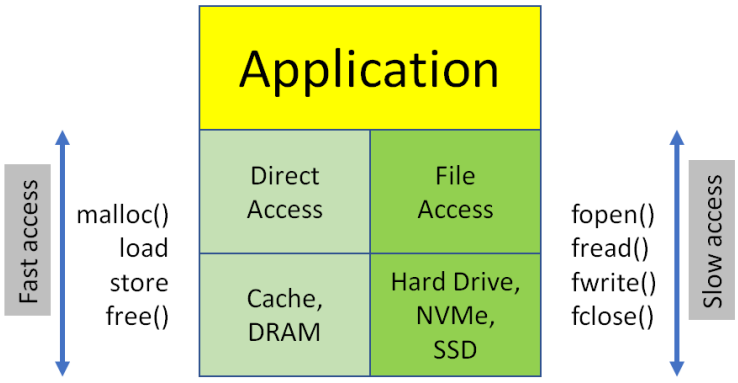

Figure 2: Access Mechanisms for Data

The direct access tier is expected to be essentially instantaneous, therefore context switching is not needed even for a single byte transfer. A program reads or write a variable without suspending the task that is running and just keeps going.

On the other hand, using the file access side of the programmer interface requires the overhead of context switching, plus the long data access latencies of the slower media types. To justify the latency hit, block accesses are required so that the latency becomes a smaller percentage of the total time for data movement.

HOW MEMORY CLASS STORAGE CHANGES THE ACCESS MODES

Eliminating the context switch through the operating system is a key improvement offered by Memory Class Storage. When tasks can perform all required work without an OS escape, orders of magnitude improvements in performance are realized. A load/store access mechanism allows tasks to remain on the active queue and not be switched out. Any size access is enabled without penalties, and block transfers operate at the full speed of the bus with essentially no latency penalty.

Fear of Flying

The common mechanism to deal with system frailties is checkpointing where executing tasks periodically save the critical state of the process to non-volatile media before continuing processing. In this way, if the system crashes before the next checkpoint, the system can restore the machine to its state prior to the crash and reload the checkpointed data and resume processing from that point forward.

Traditionally, as stated previously, this checkpointing had to be to I/O bus based persistent storage, such as tape drives, hard drives. This required a context switch through the filesystem

driver, long access latency, then a dump of the checkpoint data. Figure 3 is a graphic example of the interruption of flow caused by checkpointing.

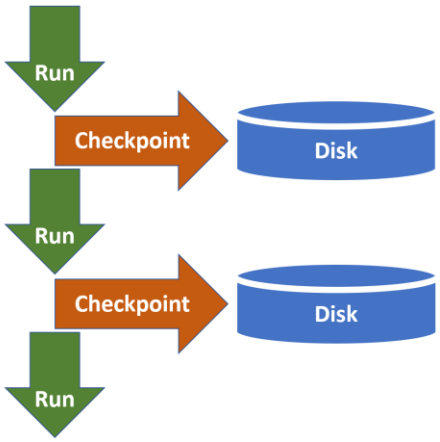

Figure 3: Checkpointing During Application Flow

It's worth noting that checkpointing burns a lot of power to achieve this enhanced data reliability. The checkpoint data is transferred through the memory controller from DRAM to the I/O channel, into the buffer device of the drive, and finally written to the non-volatile memory media.

Most data processing systems incorporate the disks used for checkpointing directly into the same chassis, keeping latency as low as possible and helping scale the ratio of total memory to the required checkpoint disk resources for the installed memory capacity. Because of the extreme sensitivity to heat that Flash memory in SSDs suffer from, the drives are placed near the intake of cool air. The SSDs restrict airflow to other components, including CPUs and memory, impacting the overall system design significantly. Clearly, eliminating checkpoint drives from system architectures would simplify system design and cooling significantly.

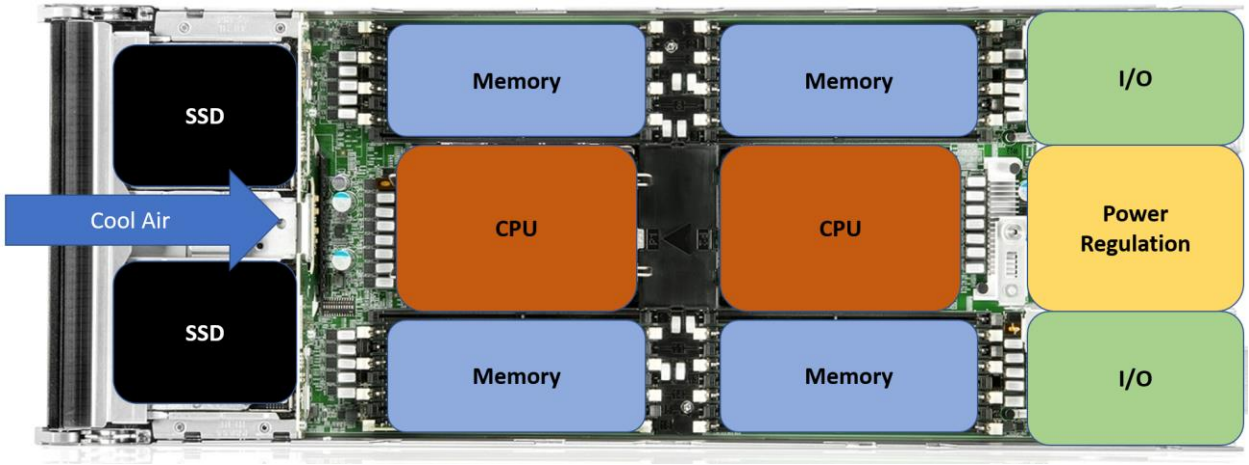

Figure 4: Typical Server Blade Architecture

Non-Volatile Memory Moves In

Non-volatile memory is moving closer to the memory controller, providing superior methods for data persistence. The DRAM bus initially only allowed DRAM on it, however many methods are deployed to abstract the DRAM bus to offer new functionality including various versions of data persistence. One key factor to consider is that the unmodified DRAM bus is fully deterministic: when a read or write command is issued, it is required that the bus respond in an exact number of clocks. No accommodation is made for a device that requires additional time to process a command. As a result, recent methods to add non-volatile memory to the DRAM channel have required some number of sideband signals not in the original DRAM bus definition.

CURRENT STORAGE CLASS MEMORY MODULE SOLUTIONS

NVDIMM-N	Optane & 3DXpoint
♦ Full DRAM performance ♦ Requires battery or supercapacitor backup power ♦ Half the total capacity of a DRAM only module ♦ 1-2 minutes for data backup and restore ♦ Only sideband is SAVE_n which need not come from memory controller	♦ Two modes of operation: memory mode and direct ♦ Neither mode runs at full DRAM speed ♦ Data persistence only in slower direct mode ♦ Proprietary DDRT protocol limits to Intel servers only ♦ Much higher module capacity than DRAM ♦ Memory mode requires separate DRAM module which does not increase total capacity

Table 1: Persistent Memory Module Options

System performance for the storage class memory module types varies widely. Each of the persistent memory module types described in Table 1 has significant limitations when compared to DRAM, from reduced capacity to reduced performance, making none of today's NVM solutions ideal. As a result, a majority of systems use these memories sparingly, mounting them as disk drives to partition them from the higher performance DRAM. As stated before, once a media is mounted as a drive, performance suffers significantly since all accesses must trap through the OS drivers.

MEMORY CLASS STORAGE REPLACES DRAM

Memory Class Storage avoids the pitfalls of current NVM module solutions by offering full DRAM performance. When a system can rely on consistent no-compromise performance throughout the memory subsystem, it no longer needs to mount that memory as a drive.

Checkpointing in the New Hierarchy

The storage class memory architectures, by nature of having slower performance than DRAM, are often mounted into the system as drives and accessed as a fast SSD. This allows for the same software to run as shown in Figure 3. Performance is faster than SSD since accesses are over the DRAM bus which operates faster than an I/O bus, however significant delays are still taken due to the need to trap out of running applications through the operating system file drivers, as shown in Figure 5.

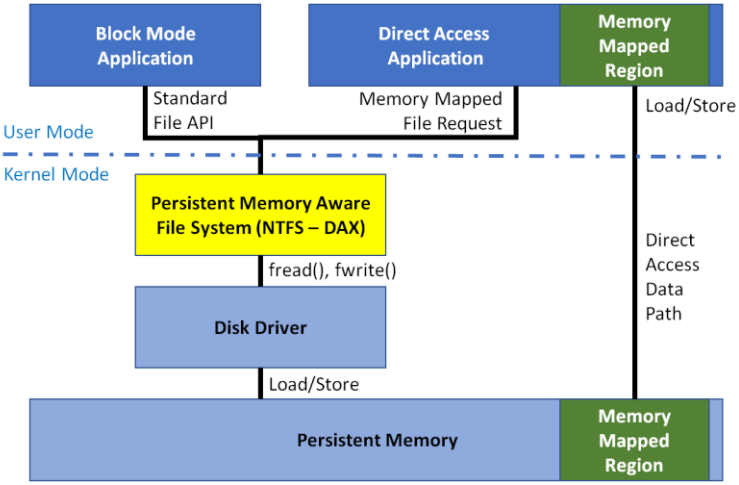

Figure 5: Persistent Memory Aware Operating System Access Paths

The DAX direct access path permits applications to address persistent memory using load and store operations without escaping to the filesystem drivers. This requires that applications be rewritten to use this path, and large scale commitment to these rewrites is less likely when the persistent memory has lower performance than DRAM. Applications have to carefully partition data based on this asymmetrical performance characteristic to avoid major performance penalties.

Industry tools to optimize memory performance on the fly exist as well. MemVerge and Chameleon are two such environments that are capable of monitoring data access patterns. Using these monitors, data can be moved to appropriate memory tiers based on how often the data is used. The net result is that applications literally run faster the longer they run because these performance monitors optimize the data access.

DAX AS AN ENABLER FOR MEMORY CLASS STORAGE

Memory Class Storage overcomes this hesitance to port applications to direct access because MCS devices operate with full DRAM performance. The inherent persistence of MCS simplifies the partitioning of data in the application because all memory has the same high performance.

The introduction of MCS into the memory channel, therefore, is a revolutionary improvement in systems architecture.

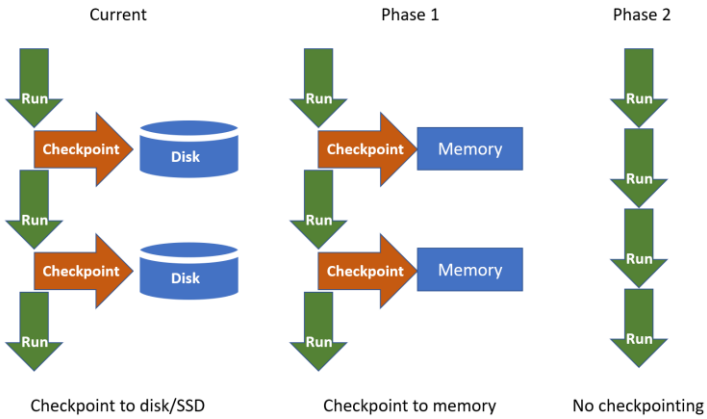

Figure 6: Phased Transition from Checkpointing to Disk to Eliminating Checkpointing

The greatest impact of deploying MCS modules is that, for the first time, high performance systems can be built without any storage at all. Since all data is inherently persistent and impervious to power failure, checkpointing can be completely eliminated. As shown in Figure 6, DAX mode provides a graceful way for applications to migrate from traditional checkpointing to disks to checkpointing to memory, and finally, direct access mechanisms that eliminate the need for checkpointing completely.

Without the need for checkpointing, clearly the need for external storage devices becomes optional.

Nantero NRAM®, The Ideal NVRAM

Nantero NRAM is an NVRAM constructed using carbon nanotubes to implement a resistive switching memory device. It can be designed using a traditional 1T1R structure, but for higher densities it is preferable to use an internal crosspoint structure. Internal circuits translate the I/O physical interface of the crosspoint into the banks, rows, and columns of a standard DRAM interface.

Nantero NRAM is the ideal NVRAM. The initial version of NRAM uses a DDR5-compatible interface, and design is under way on a DDR6-compatible interface. NRAM operates with the full speed of DRAM, with infinite data retention once the memory cells are written, even in situations such as power loss and under extreme temperature conditions. With no wear-out mechanisms, there is never a need for non-deterministic delays such as wear leveling.

With no dynamic cells like a DRAM array, Refresh is never needed. Systems can either see a 15% reduction in power with their existing protocol, or take advantage of eliminating Refresh

recovery intervals to achieve a 15% increase in data transfer performance at the same clock frequency and power compared to DRAM. Combining these advantages, NRAM provides a 34% improvement over DRAM in gigabytes processed for every watt burned.

NRAM also offers a non-destructive activation. Banks never close once opened, and can be accessed at any time. There is no need for Precharge commands which are recognized for DRAM compatibility but ignored; this opens up additional command slots for system use.

NRAM incorporates a write-through guarantee from the inputs of the device to the core non-volatile memory array. Data is permanent within tens of nanoseconds of delivery, freeing the requesting application to continue execution immediately. This contrasts with other storage class memory solutions that only offer data persistence with a separate Flush command, or rely on backup energy in the case of power failure.

Can NRAM Compete on Price?

The success of DRAM and Flash as the dominant memory solutions is their ability to offer a low price per bit of stored data. In order for Memory Class Storage to compete with DRAM, it must offer a competitive price as well. NRAM is well positioned to offer competitive pricing. The main factors contributing to NRAM's competitiveness are process geometry and three dimensional scalability, processing steps, and raw materials.

NRAM achieves the upper limit of DDR5 addressability, 32Gb per chip, on a 24nm logic process. This is achieved using a combination of small memory cell sizes yielding 8Gb per layer, and building up four layers of cells on top of each other. To achieve equivalent device density, a DRAM must have all 32Gb of cells on a single layer, requiring fine geometry processes such as 1Y or 1Z. The cost of NRAM is inherently lower than equivalent DRAM both from the simplicity of manufacturing and the higher density per area due to multiple 3D layers.

The simplicity of processing steps required to layer carbon nanotubes cells in 3D is one of NRAM's great strengths. Drivers and receivers for the memory array are fabricated on standard wafers. On top of the wafer, a layer of carbon nanotube material is deposited using standard spin coat equipment with procedures much like a photoresist layer, and cured. Metal is added on top of the cured carbon nanotube layer, and standard patterning and etching done to separate metal lines and singulate the memory cells. This is repeated as many times as needed to achieve the desired device density.

Carbon is the second most common element on earth, greatly simplifying the supply chain for carbon nanotube based memories and keeping production costs low and predictable. This compares quite favorably with other non-volatile memory technologies such as MRAM which requires a number of exotic compounds. Of the many technologies competing to become

Memory Class Storage, Nantero NRAM offers the lowest cost path to high density high performance non-volatile memory.

NRAM Density Roadmap

Traditionally, DRAM density doubled approximately every 2 years. This trend may be slowing down as evidenced by the lack of support for DRAMs greater than 32Gb in the DDR5 SDRAM specification. This is due to increasing difficulties scaling an inherently 2D architecture of tiny capacitors to fabrication geometries as small as 10nm or less.

NRAM is inherently a scalable solution. It is capable of competing with DRAM using silicon processes one or two generations older than equivalent DRAMs, for example, 14nm to make a 32Gb NRAM versus 10nm to make a 32Gb DRAM. The older fabs are partly depreciated, making them less expensive to use. However, analysis shows that NRAM can be successfully scaled to the newer processes as well, achieving increases in density with each shrink as more memory cells fit onto each chip. NRAM built in a 10nm process similar to that used to make 32Gb DRAMs would yield NRAM densities of 64 to 128Gb per chip.

Is MCS an Evolution or a Revolution?

Yes.

While the migration of non-volatile memory into the DRAM channel is an evolutionary incremental step in line with decades of improvements and refinements, it is not possible to overstate the impact that NVRAM will have on systems architectures when it has the potential to disruptively change design choices that have been universal since 1971. Essentially every system architecture choice made through these decades since 1971 has been predicated on the volatile nature of the memory subsystem, i.e., understanding that the D in DRAM stands for "Dynamic".

A non-volatile main memory subsystem changes server architecture in dramatic and permanent ways. Systems no longer need to use the I/O channel to perform application checkpointing, which includes avoiding performance delays caused by context switching to I/O drivers. For some systems, it will no longer be required to even have a storage subsystem as shown in Figure 7; if all data fits into main memory, there is no longer a need for external storage. In-memory computing combined with data persistence results in a high performance, high reliability, self-contained processing subsystem.

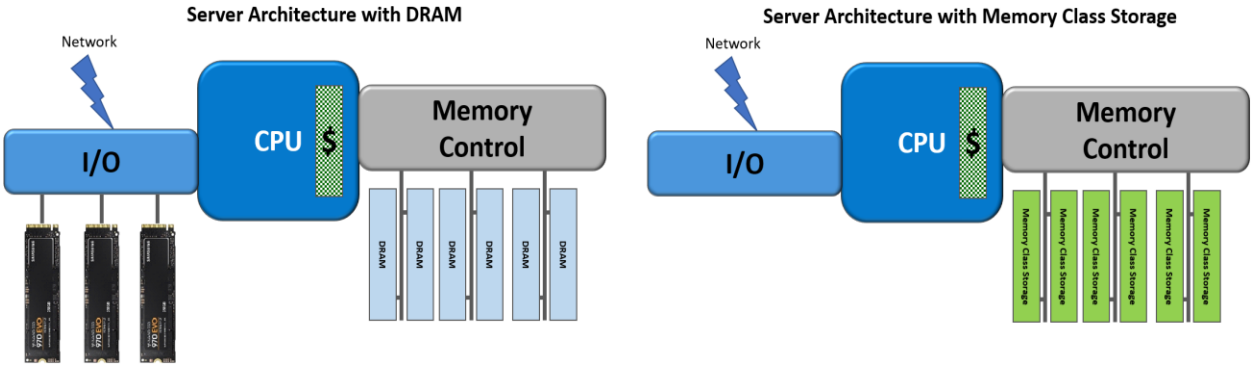

Figure 7: Comparing Servers with External Storage and Servers with MCS

NVRAM may also be combined with enhancements such as fabric buses like Compute Express Link (CXL) to provide persistent expansion of the system memory space where the Memory Class Storage nodes can reply to requests in a fully deterministic way. Putting system expansion in the memory domain instead of the I/O domain avoids the performance degradation caused by operating system overhead.

Artificial intelligence devices, including deep learning or hyperdimensional computing variants, can exploit NRAM technology to allow back propagation of data into the internal cells without the need to provide access paths for checkpointing of intermediate data. This results in significantly higher density and higher performance AI solutions using NRAM as they are able to escape the von Neumann bottlenecks and apply NRAM as high performance embedded storage.

Conclusions

In summary, NRAM technology redefines memory for server architectures. It is a natural replacement for DRAM, offering a compatible interface and much higher memory capacity in the coming generation. It may be integrated into new fabric and artificial intelligence devices to provide magnitudes of performance improvements.

Checkpointing critical data has been a requirement for all DRAM based systems for decades, however the intrinsic persistence of NRAM enables new architectures that eliminate this power and performance vacuum – just leave the data where it belongs!

NRAM offers higher performance, lower power, lower cost, and higher density than DRAM.

The persistent memory revolution is coming, and just in time.

--
bg