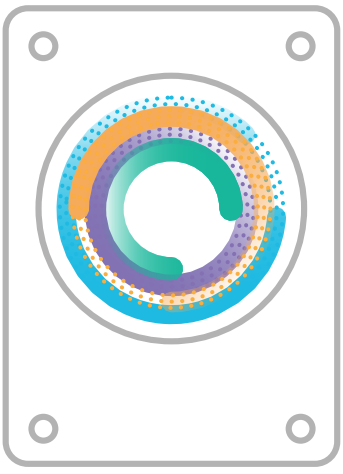
Western Digital.

Zoned Storage Now Encompasses Both HDD & SSD Technologies

WHITE PAPER



Zoned Storage Now Encompasses Both HDD & SSD Technologies

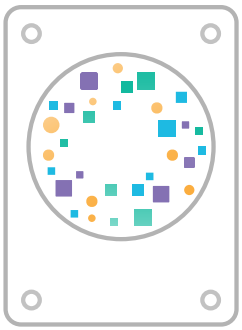
Both HDDs and SSDs have their own on-device controllers that are used to provide low-level manipulation of the drives. The division of compute tasks performed by the host-side CPUs and the drive-side controllers has evolved over time, with the boundaries moving back and forth between them. Computational architectures have evolved, especially in data centers where — as systems started to scale — local optimizations performed by the devices began to impact global optimizations.

The concept of Zoned Storage is a relatively recent development that refers to a class of storage devices — both HDDs and SSDs — that enable the host system and the storage devices to cooperate so as to achieve higher storage capacities.

As its name suggests, Zoned Storage involves organizing or partitioning the drives into zones. In the case of HDDs, Zoned Storage is implemented using a technology known as shingled magnetic recording (SMR). In the case of SSDs, Zoned Storage is implemented using a technology known as Zoned Namespaces (ZNS).

Data Taken As-Is | Serialize Data | Zoned Storage

HDDs



An HDD contains one or more rigid rapidly rotating platters coated with a magnetic material. Electromagnetic read/write heads are positioned above and below each platter. Data is stored on the platter as a series of thin concentric rings called tracks. In turn, the tracks are sub-divided into sectors.

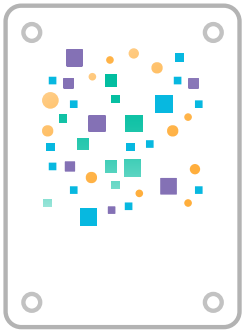
The HDD also contains a hard disk controller (HDC), which may be thought of as a small processor that is used to provide low-level control of the drive. When the main (host) computer first powers up, it communicates with the HDCs on any HDDs to determine their types and capacities. As far as the operating system (OS) on the host computer is concerned, data addresses on an HDD are logical rather than physical in structure. The HDC handles communication between the host and the drive; it employs error correction codes (ECCs) to detect and correct data errors; and it keeps track of any problem areas, such as bad sectors, on the drive, and remaps failing sectors to spare sectors that are provided for this purpose.

Since the advent of the HDD, there has been a constant motivation to reduce size, increase capacity, and decrease cost. This is especially true in the case of hyperscale facilities like data centers.

SSDs

SSDs are storage products based on multiple NAND flash devices. Flash memory stores information in an array of memory cells made from floating-gate transistors. In single-level cell (SLC) devices, each cell stores only one bit of information. Multi-level cell (MLC) devices can store two bits per cell, while triple-level cell (TLC) devices can store three bits per cell. The most recent development in flash technology is quad-level cell (QLC), which can store four bits per cell.

Although they offer high speed and low latency, the very nature of NAND flash means that SSDs based on this technology have inherent restrictions, such as being composed from blocks that have to be erased before new data is written into them and having regions that have to be written sequentially.



Remembering that the original SSDs were created to look like HDDs insofar as the host system was concerned, the SSD's controller handled these restrictions "under the hood" using an internal management system called the flash translation layer (FTL).

Unfortunately, implementing the FTL often uses DRAM. The larger the capacity of the drive, the more DRAM used. For today's larger drives, 1 GB of DRAM or more is not uncommon, which increases the cost of the drive. Furthermore, performing data management internal to the drive introduces other inefficiencies, including write amplification, over-provisioning, and quality of service (QoS) variability.

In order to manage the constraints associated with not overwriting data, the SSD's controller has to perform garbage collection (GC) by moving data around the drive internally. This process results in multiple writes of the same data (hence the term "write amplification"); it requires extra space to be set aside ("over-provisioning"); and — since garbage collection can kick in at any time without the knowledge or control of the host processor — it can delay the drive's response time resulting in QoS variability.

To summarize, conventional SSD controllers must support these tasks:

1) **Garbage Collection** – Deleting, moving & recombining data to better organize the drive

2) **Over-provisioning** – Reserving drive space for internal garbage collection

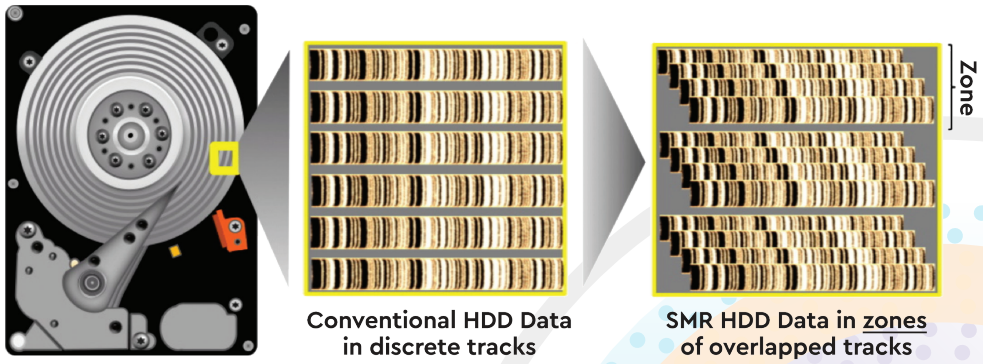
3) **Write amplification** – The same host data is rewritten multiple times as the data is moved around

4) **Large quantity of DRAM** – Used for managing incoming data caching and logical to physical mapping

Introducing Zoned Storage

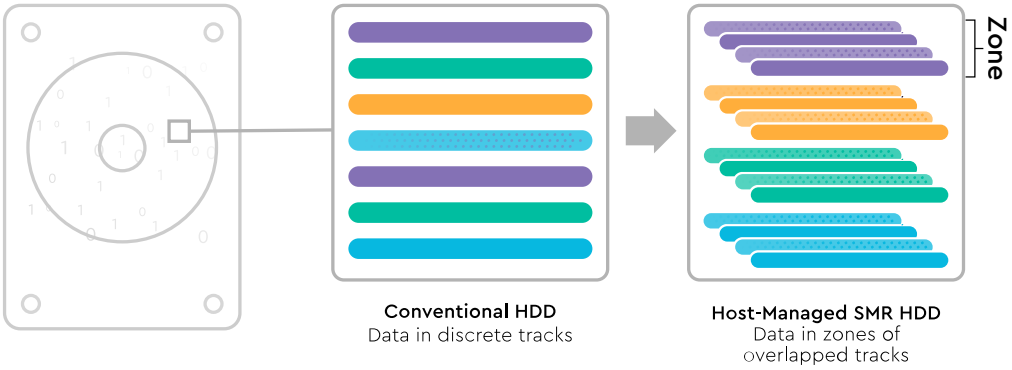
Conventional HDDs record data by writing non-overlapping tracks parallel to each other. However, a key aspect of an HDD is that its write track footprint is significantly wider than that of the read head used to recover the data.

In 2002, a new magnetic recording scheme was proposed whereby the recording tracks were partially overlapped, taking advantage of the disparity between the width of the write head versus the width of the read head to leave behind a more tightly packed set of recorded tracks. At that time an analogy to the way in which rows of shingles overlap each other on a roof, the term "Shingled Writing" was coined to describe this new recording scheme. Over time, this term evolved into Shingled Magnetic Recording (SMR) and — following an effort by the International Disk Drive and Equipment Materials Association (IDEMA) — the SMR term became standardized in the industry circa 2010.



Comparison of conventional and SMR HDDs (Image source: ZonedStorage.io)

The initial concepts of Zoned Storage — whereby the address space is divided into zones of overlapping tracks that can be independently written and erased — were described in a couple of foundational patents filed by HGST in 2003 (HGST was acquired by Western Digital in 2012). Unfortunately, the inherent architectural difficulties associated with SMR initially discouraged its adoption by HDD manufacturers. However, by the late 2000s, it was becoming increasingly clear that the traditional scaling of track width was facing severe headwinds and that — despite its difficulties — SMR offered the best path to continue scaling areal density.



At this point, the HDD industry was presented with a problem similar to that of the early days of SSDs. That is, it was faced with two main scenarios, or models. The first option, called Drive-Managed, is where all of the Zone handling is performed by the drive itself, thereby providing a backward-compatible interface that allows the drive to be plugged into the system without having to modify the OS. The second alternative, called Host-Managed, does require modifications to the OS, but it also delivers predictable performance and control at the host level. There is one further implementation option, called Host-Aware, which provides backward compatibility with regular disks while also providing the same host control interface as host managed models.

Two of the main interface standards used to allow a host computer to talk to an HDD are the Small Computer System Interface (SCSI) and the Advanced Technology Attachment (ATA) interface. In order to implement the Host Managed and Host Aware models, the OS has to be augmented with additional capabilities. In the case of the SCSI standard, these capabilities are known as Zoned Block Command (ZBC). The corresponding capabilities in the case of the ATA standard are known as the Zoned Device ATA Command Set (ZAC).

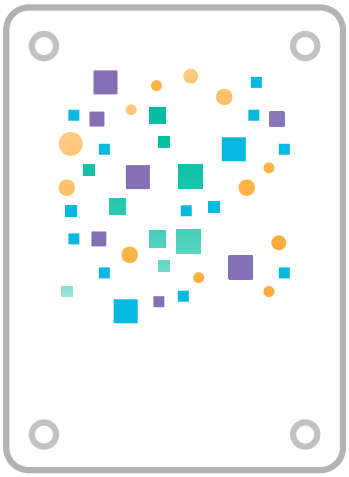
Of particular interest for the purposes of this paper is that Zoned Storage is being extended to cover solid-state drives (SSDs) using a new standard called Zoned Namespaces (ZNS).

The Evolution of SSDs

As was previously noted, when SSDs were first introduced, they were designed to use the same interface standards as HDDs. Furthermore, the fact that SSDs were much faster than HDDs masked any inefficiencies associated with their internal data management functions.

Over time, however, the overhead of using legacy HDD interfaces became apparent and a number of startup companies, most notably Fusion IO, introduced designs employing NAND flash-based PCIe cards that bypassed traditional HDD host bus adapters (HBAs). These designs also put the NAND management functions on the host by means of proprietary device drivers. While this solution enabled much higher performance, it also required complex integration of the proprietary device drivers with the host applications.

The overhead of HBAs and the HDD software stack was addressed by the industry through the creation of the Non-Volatile Memory Express (NVMe™) organization and associated specifications. The NVMe protocol is layered directly on PCIe and removes the need for an HBA. This also introduced an architecture with multiple independent input/output (I/O) queues, thereby removing another big bottleneck associated with the legacy HDD architecture that required all I/O operations to be serialized into a single I/O queue. The associated host software stack was also substantially streamlined, thereby allowing NVMe SSDs to provide substantially improved performance relative to legacy SAS or SATA SSDs.

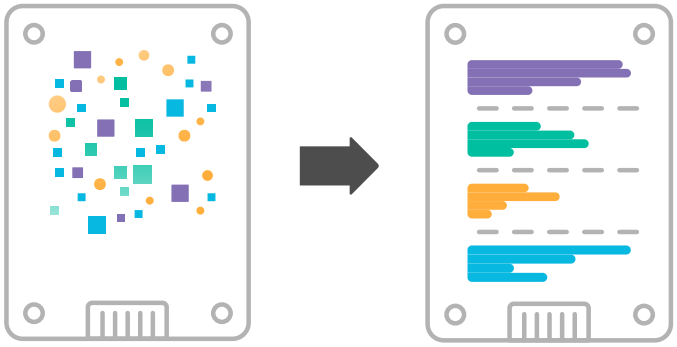


While NVMe SSDs provided a much more efficient interface and architecture, they still carried one important legacy architectural construct from the early HDD era: namely, the linear block address space. That is, these SSDs continued to be addressed using 4kB blocks that could be independently read and written in any order without restrictions. That meant that these SSDs still had to maintain the FTL and internally abstract the true nature of the NAND flash media.

ZNS Standard

The lack of a standard solution made progress somewhat haphazard. In 2018, for example, Microsoft led an effort to try to create a standard around the Open Channel concept. Known as Project Denali, this effort was introduced by the OCP (Open Compute Project), which is an organization that shares designs of data center products and best practices among companies. Around that time, realizing the synergies between the raw NAND flash erase block constraints and that of SMR, Western Digital proposed the concept of Zoned Namespaces (ZNS). With inputs and collaboration from many companies, the NVMe consortium standardized the ZNS specification in June 2020.

ZNS extends the existing SMR HDD zone model to benefit from the mature zone storage stack in Linux and other software ecosystems. All of the data management tasks that are problematic for an SSD controller to handle internally can be addressed if these tasks are instead handled by the host system.



Comparison of conventional and ZNS SSDs (Image source: ZonedStorage.io)

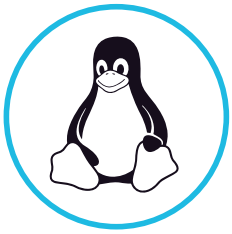
SSDs that support ZNS do not require large quantities of DRAM to implement the FTL. The act of implementing Zone Storage on the host allows the system software and hardware to work together more efficiently by eliminating the multiple (duplicated) levels of indirection required for logical to physical mapping between the host and the SSD controller and the file system to the device. ZNS also reduces the need for over-provisioning and removes the issues associated with write amplification and QoS variability. In short ZNS addresses the issues of scale, QoS and TCO which hinder conventional SSDs. The architecture of a ZNS SSD yields the following characteristics:

1) **Garbage Collection** – Minimal required for a ZNS SSD
2) **Over-provisioning** – Virtually none needed
3) **Write amplification** – Because of serial writes this is 1 or very close to 1
4) **Small amount of DRAM** – Minimal required to manage incoming data caching

Additionally, although TLC and QLC flash technologies offer higher capacities, they cannot sustain the same number of write cycles offered by SLC flash. ZNS opens the door to mixed technology SSDs that comprise both SLC flash (providing high endurance) and TLC/QLC flash (providing high capacity) to be effectively utilized.

ZNS Architecture and Support in the Linux Kernel

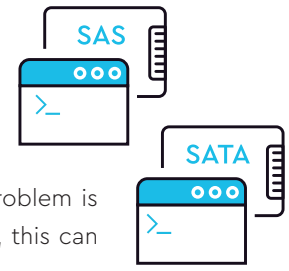
Drives that support Zoned Storage are known as zoned block devices (ZBDs). Support for ZBDs in the form of SMR HDDs was introduced in Linux with kernel version 4.10, which was released in 2017, and which provided functional ZBC/ZAC command support. This support has continued to improve, with device-mapper support and filesystem support available in more recent kernel releases.



One of Western Digital's research milestones was to contribute base ZNS support upstream into the Linux kernel before 2020. To this end, Western Digital submitted multiple changes to the kernel following public release of the ZNS specification in mid-June 2020. In mid-August 2020, these changes were accepted and published in the first release candidate (RC1) for version 5.9 of the kernel. This means that Western Digital's proposed changes to support ZNS in the kernel have been accepted by the Linux community, with ZNS support now supported in the released 5.9.0 kernel.

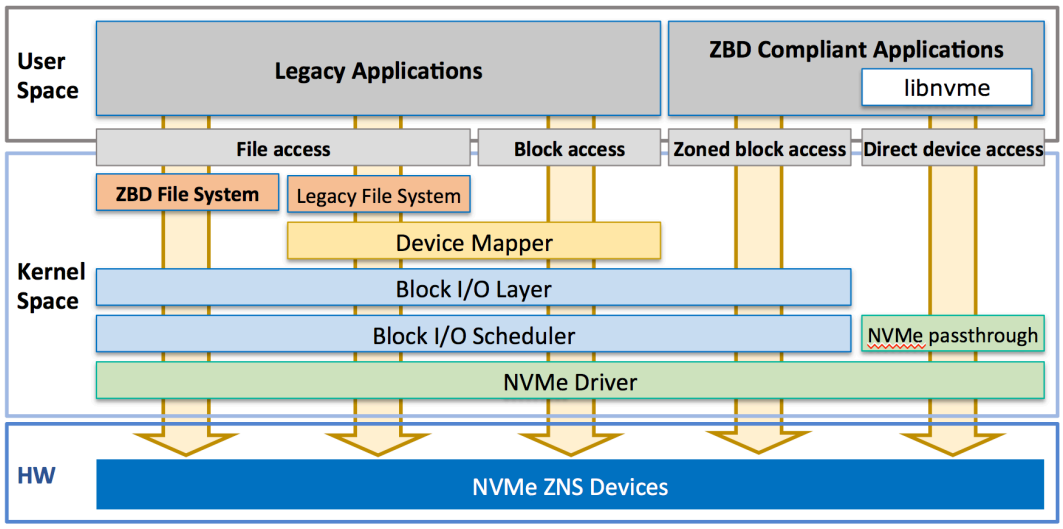
Within the Linux kernel, the block I/O layer and the device driver layers together support most of the basic input and output operations performed on ZBDs, including generic reading and writing, management of queues, and conversion of generic block I/O requests into specific commands for whatever protocol is being used at the device level (SCSI, SATA, NVMe). The most obvious change required for ZNS is to rewrite portions of these layers to support the new commands as defined in the NVMe ZNS specification.

Beyond these straightforward changes, there are a set of problems that are more fundamental and that require substantial changes to the basic "plumbing" of the Linux storage stack in the kernel. These mostly have to do with command ordering. One fundamental problem is that, with zoned storage, the Linux kernel cannot be allowed to randomly reorder write commands. Recall that, with zoned storage, it is necessary to write sequentially within each zone. This means that if two consecutive write commands to the same zone get reordered then things will break, violating the sequential write rule and potentially overwriting previously written data within the zone.



While this command reordering problem was solved previously for SMR, ZNS introduces new issues that need to be addressed. Specifically, one problem for ZNS has to do with the multi-queue architecture used for NVMe. Prior to NVMe, each storage device generally had only a single queue in the block layer, whereas NVMe was designed to have multiple queues (for example, one queue per CPU core) in order to increase performance. The problem is that if multiple queues end up writing to the same open zone within a single ZNS device, this can result in a command ordering problem.

One way to address this problem would be to provide a synchronization mechanism between queues (e.g., a lock), however this would have significant performance impact. To get around this problem, a "zone append" command was introduced to allow writing to a zone without specifying the starting logical block address (LBA). With zone append, instead of writing to a particular location in a zone, the write command starts from the write pointer's current location (provided there is enough space left in the zone). Following the completion of the write operation, the device returns the LBA location where the data was actually written. This mechanism requires bidirectional commands and communication, which can return the LBA location back to the host after completion of the write operation, along with appropriate support in the kernel.



Zoned block device support features (Image source: Western Digital)

NVMe's ZNS Devices will be compatible with the Linux kernel ZBD interface. The kernel and user space tools for ZBDs will be updated to enable ZNS support now that the specification is ratified and published. Further details can be found on the **www.zonedstorage.io** website.

Conclusion

The concept of Zoned Storage is a relatively recent development that refers to a class of storage devices that enable the host system and the storage devices to cooperate so as to achieve higher storage capacities, improved QoS and lower TCO.

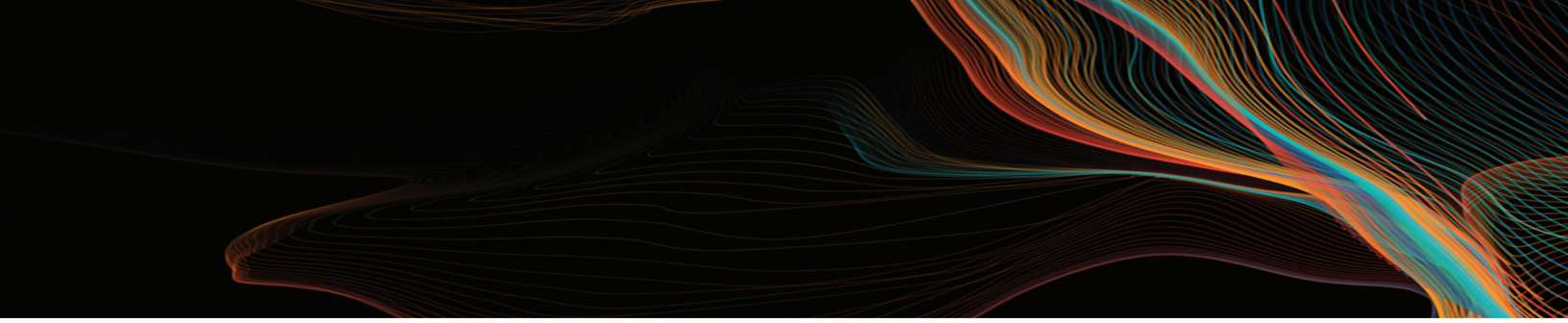
The first Zoned Storage devices were HDDs that employed shingled magnetic recording (SMR) technology, in which each new track partially overlaps the previous track. The ANSI (American National Standards Institute) approved the specifications for ZBC and ZAC interfaces to SMR HDDs. The NVMe organization has similarly ratified the Zoned Storage concept to encompass SSDs by standardizing the Zoned Namespaces (ZNS) specification. ZNS extends the existing SMR HDD zone model to benefit from the mature zone storage stack in Linux and other software ecosystems.

As one of the largest HDD/SSD manufacturers in the world, Western Digital is committed to leading, participating in, and supporting industry standards. As part of this, Western Digital is helping to drive, promote, and support the ZNS specification.

To learn more about Zoned Storage visit www.westerndigital.com/zoned-storage

For more information on Western Digital's line of storage products, please visit our website at:

https://www.westerndigital.com



Western Digital.

5601 Great Oaks Parkway
San Jose, CA 95119, USA
www.WesternDigital.com

Western Digital Corporation | 5601 Great Oaks Parkway | San Jose | CA 95119 | USA

ZONEDSTORAGE-WP_110320