Hewlett Packard Enterprise

NimbleOS: Flash for the Modern Storage System

Contents

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated. Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Publication Date

Tuesday February 20, 2018 09:10:47

Document ID

wsy1513290973522

Support

All documentation and knowledge base articles are available on HPE InfoSight at *https://infosight.hpe.com*.
To register for HPE InfoSight, click the *Create Account* link on the main page.

Email: *support@nimblestorage.com*

For all other general support contact information, go to *https://www.nimblestorage.com/customer-support/*.

Introduction

Is your all-flash array “optimized for flash?” “Flash friendly?” “Purpose-built for flash?” Maybe it was “architected for flash?” These taglines sound impressive, but they do not explain what is so distinct about flash to justify a drastically different approach for storage system design.

In general, for flash arrays to be considered flash optimized, they must strive to achieve the following goals:

- Drastically prolong the lifespan of solid-state drives (SSDs), especially when multilevel cell (MLC), triple-level cell (TLC), or newer technologies are involved
- Provide consistent high performance per SSD without latency spikes
- Enable data reduction during operation, without latency spikes, to assist with cost effectiveness
- Offer extreme data integrity; the proliferation of metadata caused by the use of deduplication necessitates enormous levels of data protection

The HPE Nimble Storage unified family of flash arrays addresses these key flash design goals to provide a compelling tailored solution for flash. All HPE Nimble Storage arrays run the NimbleOS software, which uses the Cache Accelerated Sequential Layout (CASL) architecture. NimbleOS was designed from day one with optimization of flash storage in mind.

This white paper reviews crucial flash technology concepts and describes how HPE Nimble Storage arrays use the NimbleOS flash-optimized architecture to process writes and reads and to mitigate natural SSD limitations.

SSDs and HPE Nimble Storage Arrays

Hewlett Packard Enterprise (HPE) currently offers three types of HPE Nimble Storage arrays:

- All-flash
- Adaptive flash
- Secondary flash

All three array types use a common hardware architecture (same enclosure, same PSU, and so on). Each storage system has 24 drive bays. Each bay can hold a single 3.5” large form factor hard-disk drive (HDD) or a single dual flash carrier (DFC) populated with two 2.5” small form factor SSDs. Each DFC is a full-sized bay enclosure that contains two individually removable SSDs.

Figure 1: HPE Nimble Storage dual flash carrier with individually removable SSDs

All-flash arrays are populated entirely with SSDs. Adaptive flash arrays and secondary flash arrays are populated with HDDs and SSDs. In these hybrid arrays, the SSDs are used to cache data for the HDDs.

All HPE Nimble Storage arrays run NimbleOS. All features of NimbleOS—replication, compression, and more—are provided at no cost to customers. No special licensing is required for NimbleOS or for its features.

Flash Technology Review

SSDs are a very common and hot commodity these days, but a lack of understanding about how SSDs work is still common in the IT industry. This chapter briefly reviews the core flash terminology and the technologies that are referenced throughout this white paper.

SSD Terms

The following terms and concepts are key for informed discussions of NAND flash technology:

write cycles

When cells are read from an SSD, the resulting operations are fast and have no ill effect on the health of the SSD. Each cell holds a charge that determines what value is set (0 or 1) for each bit. Every time a cell is written to (that is, the charge is changed), the cell degrades a little; this is known as *write wear*. Eventually, the cell becomes unable to hold a charge. Write cycles determine how many times at a minimum the system can write to a cell before the cell is in danger of going bad. For example, an SSD with a write cycle rating of 100,000 indicates that each cell can be written to a minimum of 100,000 times before the cells might go bad.

error correcting code (ECC)

The charge level on each cell determines what value (0 or 1) is set for each bit. When the charge of a cell is changed, the charges of the cells near that cell are also affected. As a result, the charge on a cell can fluctuate over time even when that individual cell is not directly changed. The larger the number of bits that are stored by a single cell, the smaller the range of charge that determines what value is set. Smaller charge ranges increase the chance that a charge will fall outside the intended range. ECC is required to correct the charges on each cell to ensure that they stay within the range of the value that is currently set.

write wear

Write operations cause flash cells in SSDs to degrade. Write wear can be expressed as the number of writes that have occurred to each block or cell of an SSD. Another commonly used method for tracking write wear on an SSD is spare block counting.

write amplification

SSDs contain 4 KB blocks that are organized into 16 KB pages. When data is read from an SSD, reads can be completed at the 4 KB granularity level. However, writes to an SSD must be made in 16 KB pages even when writing a 4 KB block of data to the SSD. The read-modify-write approach is used to read a full 16 KB page into memory, change the individual block (or blocks), then rewrite the 16 KB page back to the SSD. The original page is marked for reclamation through a process known as *garbage collection*. Write amplification occurs when significant amounts of read-modify-write operations are triggered, causing data that has not changed to be rewritten to the SSD and increasing write wear on the cells. The visible effects of write amplification for the user are higher latency (operations take more time) and lower endurance (SSDs wear out faster).

wear leveling

Wear leveling is a process that evenly distributes write wear across all flash cells in an SSD. When an SSD is new, this process is fairly straightforward. All writes, regardless of how they have been triggered (new writes or read-modify-writes), consume unused flash cells. Flash cells continue to be consumed until all cells on the SSD have been written to at least once. Wear leveling becomes more involved when the system needs to find free space on the SSD to write data while attempting to evenly distribute writes across available flash cells.

garbage collection

As data is deleted or rewritten, SSD pages are marked to be reclaimed by the SSD so that new data can be written. This process of reclaiming SSD pages is called garbage collection. Originally, garbage collection was triggered only as a result of all cells on the SSD being written to at least once. Proactive garbage

collection jump-starts page reclamation by beginning the reclamation process before all SSD cells have been written to.

TRIM

In many storage operating systems, when a data block is deleted, the metadata structures are updated to reflect the deleted data, but the physical block is not immediately deleted. The storage operating system goes back at a later time and reclaims the deleted block to make it available for new writes to the storage system. The problem with this process for flash drives is that the SSD is likely attempting to retain deleted data in the file system when running the garbage collection process. As a result, many unnecessary read-modify-write operations might occur. The TRIM command allows the storage operating system to notify the SSD that a block has been marked for deletion and that it is safe to delete that block during garbage collection.

spare block counting

All SSDs eventually degrade to the point of being unable to accept new writes. It is critical for storage operating systems to keep track of write wear on SSDs. Each SSD has a number of spare blocks that are reserved to replace blocks that wear out. Spare block counting is the process of keeping track of how many spare blocks remain on the SSD. If the number of spare blocks hits a predetermined threshold, the SSD is considered failed, and it is reconstructed elsewhere. RAID reconstruction usually uses a quick method for reconstruction because the drive can still be read, so there is no need for parity-based reconstruction.

SSD Types

Three types of SSDs are commonly available today:

- Single-level cell (SLC)
- Multilevel cell (MLC, covering both enterprise MLC and MLC)
- Triple-level cell (TLC)

As with any technology, new flash technologies are always in development or just coming to market. Newer advancements such as quad-level cell (QLC), 3D XPoint (a nonvolatile memory technology), and storage-class memory (SCM) are outside the scope of this paper.

Single-Level Cell SSDs

SLC SSDs were the first SSDs to become available at the start of the flash storage system surge. “Single level” refers to the fact that each flash cell holds a single bit value, with two possible values (0 and 1).

Although SLC SSDs are highly reliable (rated at 100,000 write cycles), their capacity is limited and their cost is high. Most storage systems quickly moved away from using SLC in favor of MLC; more specifically, eMLC.

Multilevel Cell SSDs

The primary benefit of MLC is that each flash cell holds two bits, with up to four possible values (00, 01, 10, and 11). Although capacity density is increased, MLC has a high bit error rate that requires additional error correction measures to be put in place.

Because MLC stores more bits per cell, write cycles are greatly reduced for MLC SSDs. Initially, all MLC SSDs were generically rated for between 10,000 and 30,000 write cycles. This rating was problematic because most storage system suppliers offer five-year warranties for their storage systems. MLC SSDs are unlikely to last for the warranty periods that are required for enterprise storage systems.

For this reason, MLC SSDs have been split into two subtypes: eMLC (enterprise MLC) and MLC. eMLC came to be from the need for better write endurance through the deployed life of the SSD. Flash providers can offer eMLC SSDs that are rated at 20,000 to 30,000 write cycles. MLC is rated at 8,000 to 10,000 write cycles. The small differences between eMLC and MLC that account for the differences in write-cycle ratings (for example, eMLC uses a larger reserve) are outside the scope of this paper.

Triple-Level Cell SSDs

TLC SSDs provide greater capacity density than MLC. Each flash cell holds three bits, with up to eight possible values (000, 001, 010, 100, 011, 101, 011, and 111).

TCL SSDs are rated at 3,000 to 5,000 write cycles. TCL SSDs are generally used only in consumer-based products, such as USB thumb drives. Given that storage system suppliers struggle to offer full-term warranties even for MLC SSDs, the potential use of TCL SSDs in enterprise storage is even more challenging. Any storage system that uses TLC SSDs needs to be extremely efficient with the usage of the flash drives.

Benefits and Limitations of SSDs

One can certainly debate many aspects of system design that are related to flash technology. However, everyone agrees that the most fundamental goal of flash storage is to maximize SSD life while providing the capability for superior performance. The more that writes can be preprocessed in memory before going to the SSD, the better. Minimizing writes can greatly extend the deployed life of an SSD or allow a storage system to use higher capacity SSDs such as TCL SSDs, which have very limited write cycles.

For traditional storage systems that are designed to work only with HDDs, drive degradation caused by data writing is not a concern. In fact, more drive activity is better because it exercises the read and write heads more frequently, making it more likely for mechanical issues to be detected before they become catastrophic.

Storage operating systems that primarily deal with HDDs usually perform several postprocess activities, such as deduplication or block reclamation, that care little for how much churn is caused in the blocks on the physical drives. Key HDD design considerations are related to mechanical operations, which generally resolve around sequential versus random operations and the location of data on the platters. For example, read and write heads can span a track faster near the interior of the platter than near the outside of the platter. However, these important considerations for HDDs are meaningless for SSDs.

Modern storage systems must be specifically designed with SSDs in mind; otherwise, they will be extremely inefficient. It is critical to understand the benefits and limitations of SSDs when compared to HDDs and how these benefits and limitations can affect system design.

SSD Benefits

SSDs have the following notable benefits in relation to HDDs:

- **High altitude operation.** SSDs can operate at much higher altitudes than HDDs. In addition to not relying on mechanical components, SSDs require less air flow for cooling because they generate less heat. Therefore, the thinner air at higher altitudes does not adversely affect the drives.
- **Power and heat.** Because they do not have mechanical components, SSDs consume less power and generate less heat than HDDS. In HDDs, friction from mechanical components generate significant amounts of heat.
- **Read performance.** The read performance of SSDs is in general better than the read performance of any HDD, especially random read performance. SSDs do not have mechanical parts, so no latency is caused by seek time.
- **Reliability.** SSDs are very resilient because of their lack of mechanical parts. They are far less susceptible to shock and vibration issues as compared to HDDs. SSDs are also resilient to magnetism, which HDDs rely on to write data (for this reason, magnets can erase data on HDDs).
- **Size.** SSDs are available in the 2.5”, 1.8”, and 1.0” form factors while HDDs come in the 3.5” and 2.5” form factors. With per-drive SSD capacity increasing on a regular basis, the availability of smaller form factors for SSDs means more capacity in a smaller footprint.
- **RAID reconstruction.** The size and speed of SSDs allow them to be reconstructed exponentially faster than HDDs.
- **System design.** The power efficiency provided by SSDs translates into better options for internal system design and operational cost savings. The need for less airflow or less empty internal space for cooling means more space for components or a smaller system footprint.

SSD Limitations

SSDs have the following key limitations in relation to HDDs:

- **Endurance.** SSD cells degrade when data is written to them. With every SSD, a finite number of writes can occur before the SSD no longer accepts writes. Although HDDs also eventually fail, there is no specific indicator of when they will fail. Short of mechanical failure, there is not a practical limit to the number of times that an HDD can write to a platter.
- **Write performance.** The write performance of SSDs is not much better than the write performance of HDDs; in some cases, SSD write performance is on par with HDD write performance or is potentially worse. As for memory technology, SSD reads are comparable to RAM reads, but writes are far faster with RAM than with SSDs.
- **Cost.** The cost of SSDs has consistently become lower over time, but they are still more expensive than HDDs. Per-drive SSD capacity is on track to exceed that of the largest available HDDs, but the cost per gigabyte remains higher with SSDs. Therefore, it still makes financial sense to use HDDs (or even tape) to store certain types of data. For example, SSDs are far too expensive to store archive data.

How HPE Nimble Storage Arrays Mitigate SSD Limitations

HPE Nimble Storage arrays are a compelling portfolio of modern storage systems with extreme flash efficiency. The arrays are designed to maximize the benefits of SSD technology while mitigating its limitations.

Mitigating Endurance Limitations

HPE Nimble Storage arrays minimize the number of writes and block changes that occur on SSDs by processing changes in memory before sending them to the SSD. For example, the following strategies illustrate ways in which the arrays reduce or avoid read-modify-write operations in the SSD:

- The arrays use inline deduplication and compression because postprocess deduplication or compression results in unnecessary read-modify-write operations.
- To avoid changes that affect a single block within a page and that lead to read-modify-write operations, writes to SSDs are performed in block sizes (for instance, 512 KB) that align with page sizes.
- The TRIM command allows the storage operating system to tell the SSD which of the blocks that have been marked for deletion in the file system have yet to be deleted on the SSD media. Consequently, during garbage collection, the SSD does not attempt to retain blocks that have already been deleted, thus reducing unnecessary read-modify-write operations.

Mitigating Write Performance Limitations

HPE Nimble Storage arrays use nonvolatile random access memory (NVRAM). This approach allows writes to be acknowledged at memory speeds when they are made persistent in NVRAM rather than waiting for storage media (SSD or HDD) to complete the write operation.

Mitigating Cost Limitations

SSD cost is driven by many interwoven factors. Given that most storage array providers in the market rely on a small number of flash manufacturers, it is safe to assume that SSD technology is fundamentally the same across storage array vendors. In this scenario, providing maximum SSD capacity at the lowest cost comes down to two key factors:

- **A strong supply chain.** HPE has a tremendously competitive supply chain. The sheer size and scope of the buying power that HPE possesses enables the company to acquire SSDs at a lower cost than competitors and forces them to offer deeper discounts to match HPE prices.
- **An efficient flash technology architecture.** Most storage arrays use eMLC SSDs. HPE Nimble Storage arrays use eMLC SSDs and TLC SSDs because NimbleOS is extremely efficient with SSD endurance. In addition, contrary to what many analysts in the industry claim, not all datasets need all-flash arrays; hybrid arrays are very much applicable for many datasets. Between all-flash arrays, adaptive flash arrays, and secondary flash arrays, the HPE Nimble Storage family of storage technologies can handle any workload. When a workload can be easily addressed by a hybrid array, that means lower cost for the customer.

Processing I/O Operations in NimbleOS

Although I/O processing is similar across HPE Nimble Storage array models, there are some subtle differences in the way NimbleOS processes I/O for reads and writes in all-flash arrays versus hybrid arrays (adaptive flash and secondary flash). NimbleOS is the common storage operating system that is used by all HPE Nimble Storage arrays.

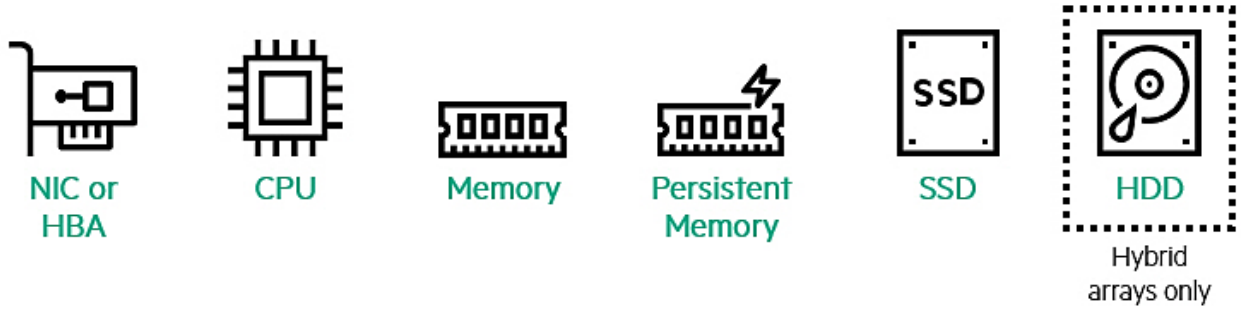
Several storage array components are used for processing I/O operations. The following physical components are often involved:

- Network interface card (NIC) or host bus adapter (HBA)
- CPU
- Memory (RAM)
- Persistent memory (NVRAM)
- SSD or HDD

Beyond the physical components, other technologies are also part of the process:

- The file system (CASL, in the case of HPE Nimble Storage arrays)
- The RAID level
- Deduplication
- Compression

Figure 2: Physical components of the storage array that are used to process I/O



Traditional storage systems are architected to rely both on compute resources—CPU and memory—and on the storage subsystem—RAID, file system, and SSDs or HDDs—to achieve system performance. NimbleOS is designed to decouple system performance from the storage subsystem and push it into the compute domain (CPU and memory).

There are two main reasons for this design:

- In the context of a storage system, CPU and memory are less expensive and their technology is evolving at a far faster rate than disk technology.
- More critically, to gain more system performance, it is easier to nondisruptively upgrade storage controllers by adding CPU and memory than it is to migrate data to a faster storage tier or to add storage media to achieve more speed, regardless of whether the resulting extra space is needed or not.

Processing Writes

Memory is still, by far, the fastest technology available for processing write operations. NimbleOS is designed to process writes at memory speeds through the use of an ultra-low latency, byte-addressable DDR-4

NVDIMM-N, a type of SCM that HPE Nimble Storage arrays use as NVRAM. This SCM technology makes writes in system memory persistent.

After the write is made persistent in memory and is mirrored to the standby controller, NimbleOS can acknowledge the operation back to the host. Later, after the write is further processed, NimbleOS coalesces multiple operations into one large object and destages the write to SSD or HDD, depending on the type of storage array being used. This approach decouples the system's reliance on the storage subsystem to achieve high write performance and avoids a large percentage of read-modify-write operations.

The complete write process for NimbleOS can be summarized as follows:

1. A write I/O request arrives from the network to the storage array through a NIC or an HBA.
2. The write is processed into the main system memory.
3. The write is committed to NVRAM on the active controller and is mirrored to the NVRAM of the standby controller.
4. The write is acknowledged back to the host.

   At this point, as far as the host is concerned, the write is complete. All processing is performed with variable blocks, so block sizes are not broken down into a fixed size by the file system.

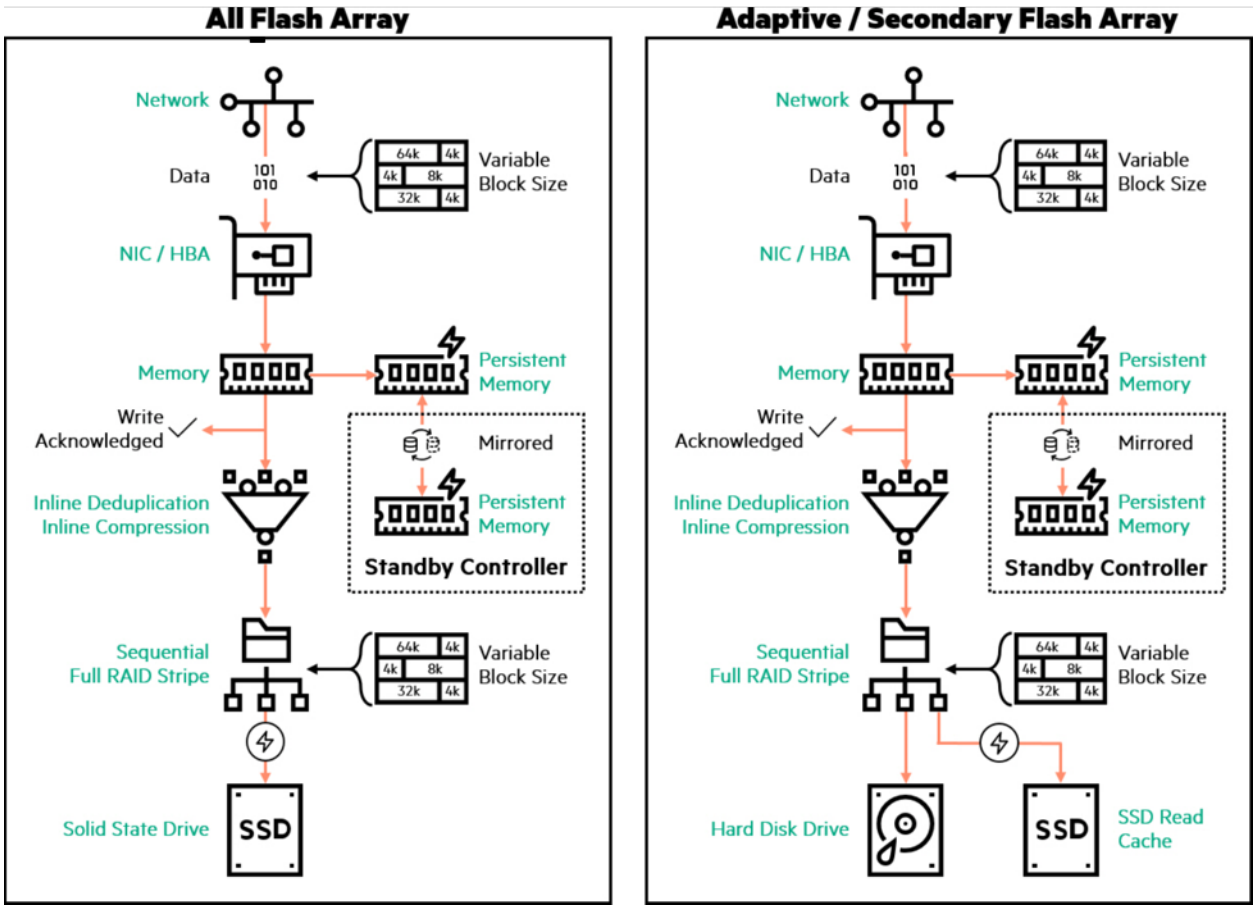
5. Postprocessing starts with NimbleOS determining if the write data is targeted at a volume in which deduplication is enabled. If so, the data is deduplicated by using variable block inline deduplication.
6. If compression is enabled in the performance policy (which is almost always the case because compression is enabled by default), the data is compressed by using variable block inline compression.

   Compression in NimbleOS does not have a performance impact, and it is an extremely efficient process. Even if compression achieves little or no savings, having it enabled does not affect performance in any negative way.

7. Data is coalesced and organized into always-sequential full RAID stripe writes (variable block). On HPE Nimble Storage systems, random writes never happen to media. The stripe size depends on the type of storage array:
   - All-flash arrays use a 10 MB stripe.
   - Current generation hybrid arrays (CSxxxx and SFxxx) use 18 MB stripes. One exception is a half-populated CS1000 array, which uses an 8 MB stripe.
   - Older generation CSxxx adaptive flash arrays use a 4.5 MB stripe.
8. The data is destaged (that is, written) to SSDs or HDDs:
   - In all-flash arrays, the data is written to SSDs.
   - In hybrid arrays, the data is written to HDDs, but it might also be written to the SSD cache if the data matches the predefined criteria for caching. The data is written to HDDs regardless of whether it is written to the SSD cache.

     Data is always written to SSDs in 512 KB chunks and to HDDs in 1 MB chunks. Chunk size aligns with the RAID stripe size for each platform type. Most fundamentally, data is written to SSDs in an even increment of the page size; for instance, a 512 KB chunk is exactly 32 SSD pages. Writing in even increments of the page size helps prolong the lifespan of SSDs because SSDs do not like being written to in sizes different from even multiples of their erase page size.

Figure 3: Overview of the write process for HPE Nimble Storage arrays



All data changes are processed inline (in memory) before they are destaged to SSDs. Therefore, when data is written to SSDs, that data is in a highly efficient format that minimizes changes to SSD blocks. If deduplication or compression were fully or even partially performed after the data is destaged to SSDs, the result would be excessive write operations that accelerate write wear on the SSDs.

Processing Reads

Although SSDs are very fast when processing reads, it is still preferable to respond to read requests from memory if at all possible. NimbleOS is designed to always check what is the fastest available media in the storage array before responding to read requests.

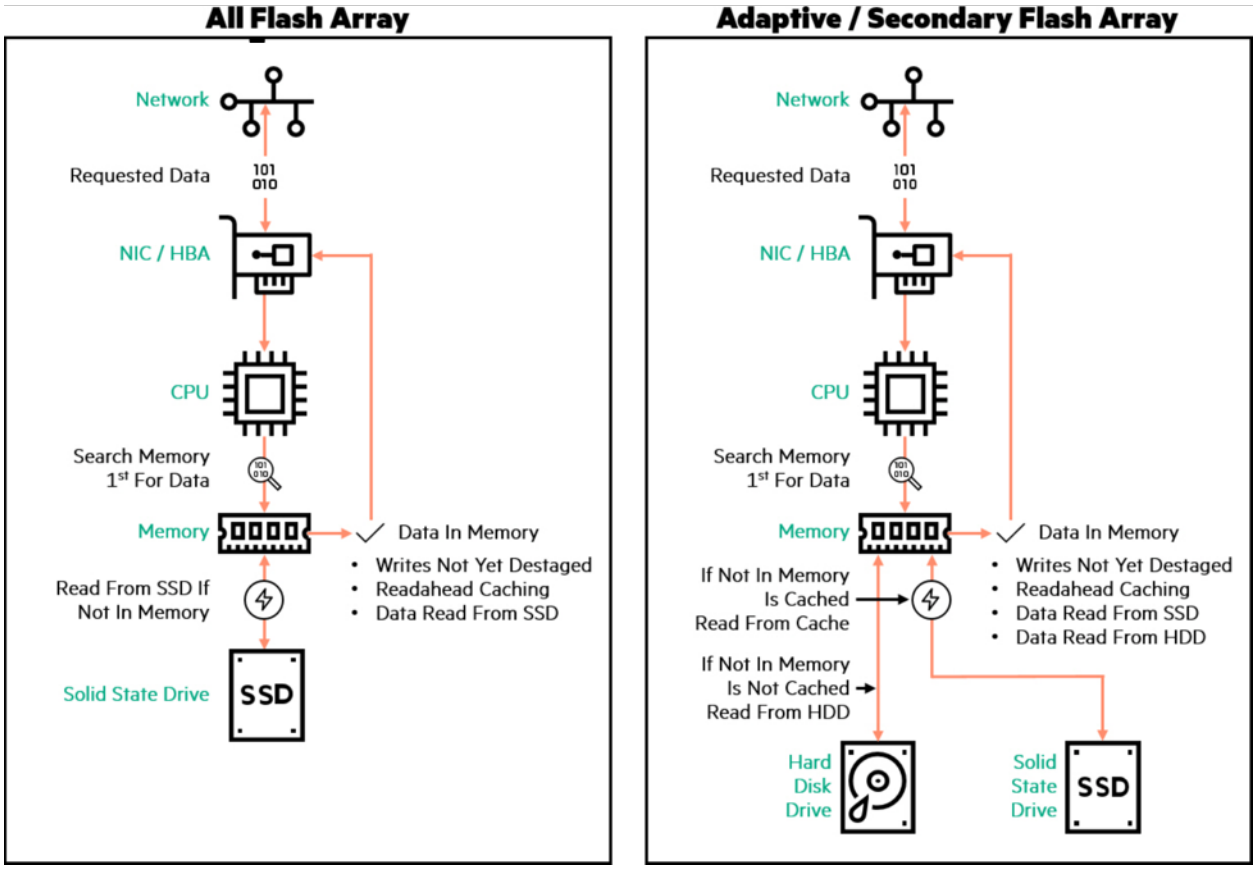
The complete read process for NimbleOS can be summarized as follows:

1. A read request arrives from the network to the storage array through a NIC or an HBA.
2. NimbleOS determines what is the fastest way to send the requested data to the host:
   - If the data is in memory, the requested data is sent to the host and the read operation is complete.
   - If the data is not in memory, NimbleOS checks the next fastest storage media available in the system:
     - In all-flash arrays, the data is read from the SSDs into memory and the requested data is sent to the host.
     - In hybrid arrays, NimbleOS checks whether the data is available in the SSD read cache:

- If the data is in the SSD read cache, the data is read into memory and the requested data is sent to the host.
- If the data is not in the SSD read cache, the data is read from the HDDs into memory and the requested data is sent to the host.

3 After completing the read request, if the data was read from HDDs (applicable only to hybrid arrays), NimbleOS determines whether the data needs to be copied into the SSD read cache to reduce the latency of subsequent read operations.

Figure 4: Overview of the read process for HPE Nimble Storage arrays



When data needs to be read from SSDs or HDDs, NimbleOS always reads that data into memory before sending it to the host that requested the data. This approach enables NimbleOS to maximize read performance for both all-flash arrays and hybrid arrays because data can be cached in memory for short periods of time.

Data can be present in system memory without first having been read from storage media (SSDs or HDDs) in two common scenarios:

- When writes have not yet been destaged to SSDs in all-flash arrays or to HDDs in hybrid arrays; in this case, the data is already in memory waiting to be destaged and it is, therefore, read from memory
- When specific blocks have been cached into memory by readahead algorithms

Readahead

Readahead algorithms identify blocks that are likely to be read in the very near future. These likely-to-be-read blocks are generally read into memory during operations that read requested data into memory.

For example, suppose that there are 10 blocks in the file system, numbered 0 through 9. Blocks 0 and 1 have already been requested and read. Right after the read request for blocks 0 and 1, blocks 2 and 3 are also requested. Readahead can see this pattern emerging. When the system goes to read blocks 2 and 3, it can also read blocks 4 through 9 into memory, anticipating that they will be requested next. If the blocks are indeed requested shortly thereafter, they will be already in memory. If, on the other hand, the blocks are never requested, they will be evicted from memory when space is needed for higher priority data.

Summary

HPE Nimble Storage arrays are designed to achieve an excellent balance between flash efficiency and performance. NimbleOS intelligently takes advantage of key SSD benefits while mitigating SSD limitations. Beyond all-flash systems, which can easily handle primary workloads, HPE Nimble Storage arrays extend into secondary workloads by maximizing hybrid system performance.

In addition to superior performance and efficiency, HPE Nimble Storage arrays deliver a robust set of storage functionality in an easy-to-consume package, from setup to ongoing operations. To learn more about HPE Nimble Storage arrays, visit the *HPE Nimble Storage* product pages.

Version History

Version	Release Date	Description
1.0	February 2018	Initial release