HP Smart Array Controllers and basic RAID performance factors

Technology brief

Abstract

RAID storage technology has been used in industry-standard servers for almost 20 years. Over that time, significant advances in disk drives, storage interfaces, RAID controller technology, and processing power have continued to change the storage landscape. This technology brief provides an overview of the basic factors driving RAID performance today, including RAID levels themselves as well as controller and drive technologies.

Introduction

Disk arrays are designed to address several basic issues with disk drive-based storage:

- Allowing the creation of large storage volumes using multiple smaller disk drives.
- Increasing the I/O capabilities and maximum throughput of the storage subsystem over that of individual disk drives.
- Increasing the reliability of data storage by using redundancy techniques to ensure that the failure of one or more physical drives does not result in a permanent loss of data.

There are many variables that influence the overall performance of RAID arrays, so it could be instructive to think about the primary factors that influence RAID performance:

- RAID levels. Each RAID level influences overall performance based on the number of low-level read/write operations and the amount of processing overhead needed to perform the associated high-level reads and writes.
- The RAID controller. This includes the processor and memory required to manage and execute the RAID operations as well as the read and write cache used to optimize read/write performance.
- The number of physical drives in the logical drive array. Having more drives in an array allows the Smart Array controller to execute more read and write operations in parallel, increasing overall performance.
- Drive performance, including drive throughput capability (MB/s) and drive performance when performing random reads and writes (I/O's per second or IOPS).
- Storage interface performance, including the protocols (SAS vs. SATA) and the speed of the physical links between the drives and the controller (3 Gb/s or 6 Gb/s).

Each of these variables not only influences RAID performance but can also, depending on the type of storage operations being performed, become the factor that determines the upper limit of the drive array's performance in a particular application environment.

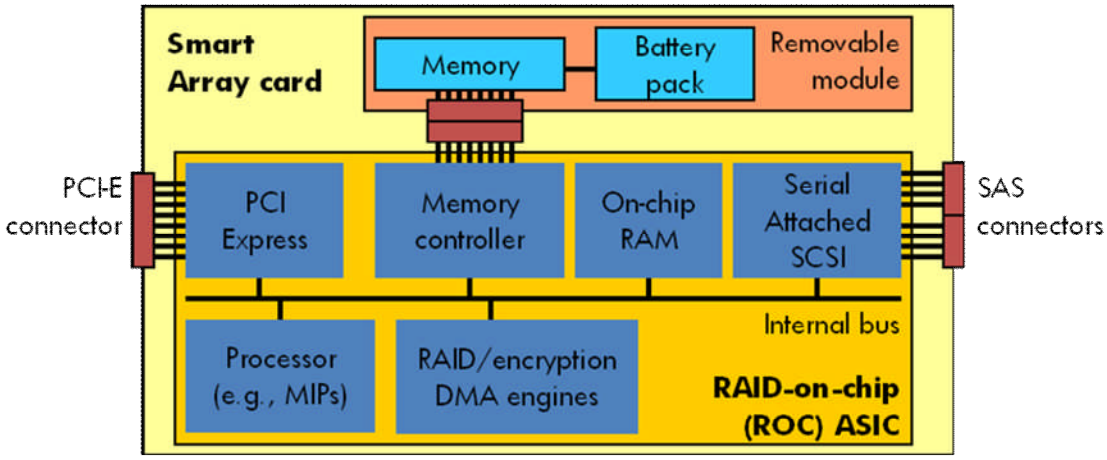
HP Smart Array controllers and performance

The new generation HP Smart Array controllers are designed to improve RAID performance. RAID performance is dependent on many different factors. Two of the more important contributors to performance are the Smart Array processor and the read/write cache that are part of the Smart Array controller.

Smart Array processing engine

The processing engine in the Smart Array controller is responsible for managing the RAID system and for performing the operations required to transform the high-level read or write requests from an application into the complex series of individual instructions required to execute this for the RAID array. The current generation Smart Array P410, P411 and P212 controllers use an embedded RAID-on-Chip (RoC) processor running at 600 MHz (Figure 1). While it is not a direct measure of overall RAID performance, the new processor is capable of supporting up to 60,000 4 KB random IOPS compared to 35,000 for the previous generation engines.

Figure 1. HP Smart Array controller architecture



The processing engine in the Smart Array controller is responsible for processing all operations, but its capabilities are particularly critical to complex RAID operations such as write operations for redundant RAID modes. Both RAID 5 and RAID 6 use mathematical XOR (Exclusive or) operations to calculate the parity data that is written to the drive array in order to provide data recovery capability in the case of a physical drive failure. This makes the processing engine's performance a key contributor to array performance - particularly write performance - for disk arrays that use these RAID levels. Performance improvements associated with the newer Smart Array controllers are also most apparent in arrays with larger drive counts. With smaller drive counts, logical drive array performance tends to be constrained by the aggregate I/O of the drives and not the bandwidth of the Smart Array processing engine.

Smart Array cache

Smart Array controllers use their optional cache modules to improve the overall performance of disk arrays for both read and write operations. The percentages of the cache being used for write caching and read caching can be configured using the array configuration utility (ACU). The current Smart Array controllers support 256 MB, 512 MB and 1 GB cache size options.

Read cache

On a Smart Array controller, read cache is used in a predictive capacity to pre-fetch data. The controller's operating program identifies the pattern of the read commands and reads ahead on the drives, placing this data into the cache where it can be more quickly accessed if the upcoming read commands call for it. Read cache is really only effective in increasing the performance on sequential read workloads. The Smart Array controller is sophisticated enough to differentiate between sequential and random workloads, using read cache pre-fetch only when sequential workloads are detected. Additionally, read cache does not greatly improve array read performance since the raw performance of a drive array on reads is already relatively fast. These are the primary reasons why the default configuration on Smart Array controllers assigns only 25% of the cache for read cache.

Write Cache

Smart Array controllers use the write cache as an output buffer that allows the host applications to post write commands to the controller and continue without waiting for the write operation to complete to the disk. The application sees the write as completed in a matter of microseconds rather than milliseconds, and the array controller will complete the actual write to disk later as it works through the list of write commands that have been posted to the cache. This technique is often referred to as posted writes or write-back caching.

In high workload environments, the write cache will typically fill up and remain full most of the time. The controller then uses this opportunity to analyze the pending write commands in the cache and determine more efficient ways to execute them. The controller can combine small writes to adjacent logical blocks into a single larger write that can be executed more quickly. This technique is called write coalescing. The controller can also rearrange the execution order of the writes in the cache in such a way that the overall disk latency is reduced. This technique is often referred to as command reordering. With larger amounts of write cache memory, the Smart Array controller can store and analyze a larger number of pending write commands, increasing the opportunities for write coalescing and command reordering while delivering better overall performance.

Write caching stores pending writes in the cache for later completion. Since the applications see the writes as already completed, these cached writes must be kept by the Smart Array until they are completed to disk. If not, data corruption will occur. Smart Array controllers address this issue by using batteries or flash memory to maintain write cache integrity even in the event of a server crash or power failure. On those Smart Array controllers where battery-backed or flash-backed cache is an option, the default is for none of the cache to be used as write cache if the option is not installed. This can be overridden, but doing so opens a window for possible data loss.

Cache width

The new generation Smart Array controllers support 256 MB, 512 MB, and 1 GB cache modules. In addition to providing significantly more cache for read and write operations, the 512 MB and 1 GB modules also use a 72-bit wide (64 bits data + 8 bits parity) cache instead of the 40-bit wide (32 bits data + 8 bits parity) cache used in the 256 MB modules. This doubles the bandwidth for moving cache data to and from the storage system, contributing further to overall increases in array performance.

Battery backed and flash backed write cache

At any point in time, the write cache in the Smart Array controller contains data that the OS and applications consider to have been written to disk but which is, in fact, still in memory on the controller. To avoid possible data corruption issues in the event of a power loss, all Smart Array controllers maintain the write cache. Battery backed writes cache uses an attached battery to maintain the contents of cache memory if power is lost. The batteries are capable of maintaining cache data for up to 72 hours. The new flash backed cache modules use onboard power from a capacitor to write the cached data to non-volatile flash memory where it can remain almost indefinitely.

It is important to note that although the Smart Array cache module can be used without battery back-up, the Smart Array controller will not use any of the cache memory for write caching if the battery back-up is not present. This will significantly impact write performance, particularly in RAID 5 and RAID 6 modes and their derivatives.

Zero Memory RAID

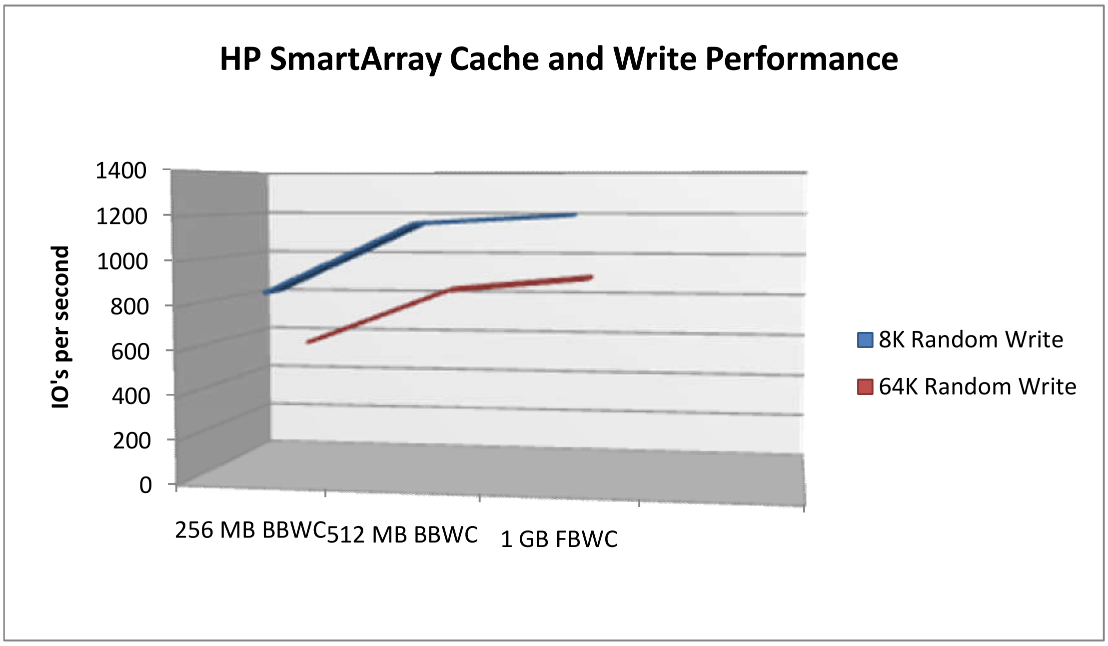
Although it is referred to as read/write cache, the Smart Array controller actually uses from 32 to 64 MB of the memory in the cache module to support the execution of advanced RAID functions, including the XOR operations required to calculate parity for RAID 5 and RAID 6 logical drives. Several of the Smart Array controllers ship without cache modules as part of their standard configuration. Not having this memory available affects more than just performance. It also limits the functionality that the controller can support.

For the new generation of Smart Array controllers, the limited modes of operation available without the use of a cache module are known as Zero Memory RAID. Zero Memory RAID provides an entry-level RAID functionality with RAID 0 and RAID 1 only, and supports only a limited number of physical drives in an array.

Overall effect of cache on Smart Array performance

Using cache provides significant storage performance improvements for Smart Array controllers, especially for write-intensive operations. While read cache may provide modest performance gains for read operations, write cache is crucial to improving the write performance of drive arrays. This is because advanced RAID levels may require up to six individual read and write operations to physical drives in order to complete a single array-level "write" to a logical drive. Figure 2 shows the relative performance of a 10-drive RAID 5 array using the P411 Smart Array controller with varying levels of cache.

Figure 2. Effect of Smart Array cache on array write performance



Configuration: P410 and P411 Smart Array controllers; 8-Drive RAID 5 logical drive; 256 KB strip size; queue depth 64; ProLiant DL380 G6

Smart Array device driver

In addition to the Smart Array controller and processing engine, the Smart Array device driver for the Windows Server operating system improves storage performance when possible. The driver analyzes the pending drive I/O queue at the operating system and logical drive level, above the Smart Array controller level. Under the appropriate conditions, the driver coalesces these pending requests, reducing the total number of I/O commands sent to the Smart Array controller in order to increase performance.

The Smart Array device driver's coalescing capability improves performance for small request-sized sequential transaction streams in environments that create large pending I/O queues at the operating system level. I/O coalescing is not performed when the I/O queue is small, since doing so might actually lower overall performance of the Smart Array storage system.

SAS links, disk drives and array performance

The new Smart Array controllers connect to the disk drives in a drive array using up to eight primary SAS-2 physical links. Each of these physical links is capable of supporting a maximum bandwidth of up to 6 Gb/s, or 600 MB/s, depending on the type of drives attached. SAS-2 links will only operate at 6 Gb/s if 6 Gb/s SAS drives are attached to them. Smart Array controllers support SATA drives operating at a maximum channel bandwidth of 3 Gb/s (300 MB/s).

SAS bandwidth is never really an overall performance limiter in application environments that rely heavily on random read and write operations. The fastest current disk drives are capable of delivering about 470 random IOPS using 4 KB reads and writes. This translates to a throughput of 1.8 MB/s, or less than 1 percent of the bandwidth of SAS-2 physical link. Even in a larger RAID configuration using a SAS expander to place 6 drives behind a single SAS channel, the aggregate throughput would be less than 15 MB/s, far less than the SAS bandwidth. Different disk drives influence the random read and write performance of logical drive arrays by virtue of the number of random IOPS that they can sustain, as shown in Table 1.

Table 1. Maximum sustained throughput and random IOPS capabilities for HP disk drives

Drive RPM	Form Factor & Interface	Max. Throughput (64 KB seq. read at queue depth>4)	Typical IOPS (4 KB random read at queue depth of 16)
15,000	LFF 6 Gb/s SAS	200 MB/s	335
15,000	SFF 6 Gb/s SAS	155 MB/s	375
10,000	SFF 6 Gb/s SAS	150 MB/s	270
7,200	LFF 3 Gb/s SATA	130 MB/s	140
7,200	LFF 3 Gb/s SATA	95 MB/s	128

With sequential operations, particularly sequential reads, SAS channel bandwidth can become a factor in overall Array performance. As Table 1 shows, no single disk drive can sustain a throughput that is capable of saturating a 3 Gb/s SAS channel.

Larger drive arrays can have multiple drives sharing the bandwidth of a SAS channel. When more than two disk drives share a single 3 Gb/s SAS channel, the performance for sequential operations will start to be limited by the bandwidth of the SAS channel. With 6 Gb/s drives attached to 6 Gb/s SAS-2 channels on the newer Smart Array controllers, sequential performance should continue to scale until more than three drives are sharing each channel.

Disk Striping and Performance

Most RAID levels are designed to provide increased read performance by distributing, or striping, data across the set of physical drives that have been configured as a single logical drive. With striping, each X number of bytes of data of the logical disk is placed on a different physical disk in the array on a rotating basis. In industry terms, each set of X bytes is called a strip. A stripe is one complete row of data strips across all of the drives in an array. HP configuration tools have used the term stripe size to refer to what most of the industry refers to as the strip size, although this is being changed in 2010.

The strip size for an array is configurable, and can be set from 16 KB up to 512 KB. In general, using a larger strip (HP stripe) size delivers higher performance for a RAID array. The Array Configuration Utility (ACU) determines the largest strip size that can be set for a given logical array based on the RAID level of the array and the number of physical drives that it contains.

RAID levels, drive count and read performance

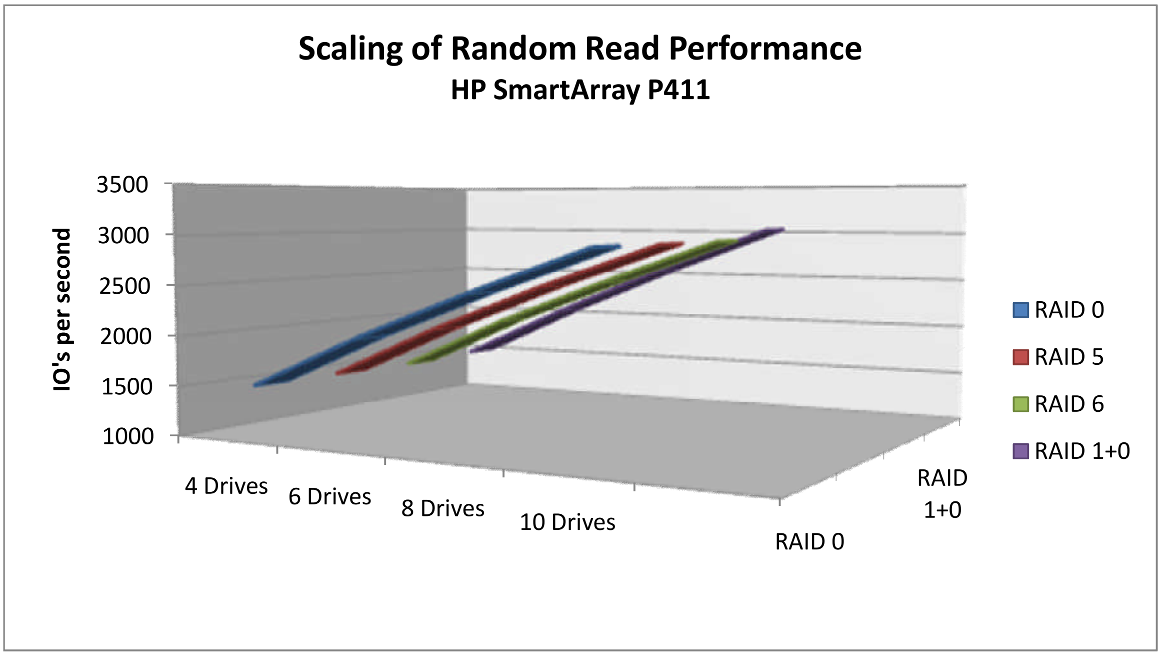
One of the goals for using drive arrays is to increase the read performance of storage subsystems over that of single physical disk drives. In general, this is accomplished by using multiple disk drives and distributing the data across them using striping. As a result, the read operations required to access data can be distributed across the multiple drives and executed in parallel by the Smart Array controller. In general, read performance for Smart Array drive arrays is typically determined more by the performance characteristics of the drives themselves and is not bound by the speed of the Smart Array processor or the cache size.

Random read performance

Drive array read performance, particularly random read performance, is greatly influenced by the use of data striping and by the number of drives present in an array. Data striping distributes data evenly across all the drives in an array, allowing the Smart Array controller to achieve increased performance since read requests can be executed in parallel across all of the disks.

RAID 0, RAID 5, and RAID 6 use data striping, resulting in similar read performance. Random read performance is typically measured in the number of small (4 KB to 8 KB) random read operations that can be performed per second (often referred to simply as IOPS). For these RAID levels, random read performance scales almost directly with drive count, as shown in Figure 3. With all other factors being equal, a 12-drive array can deliver approximately four times the number of random IOPS-per-second as an array with only 3 drives.

Figure 3. Scaling of 8 KB random read IOPS. RAID 0, RAID 5, RAID 6 and RAID 10 (1+0)



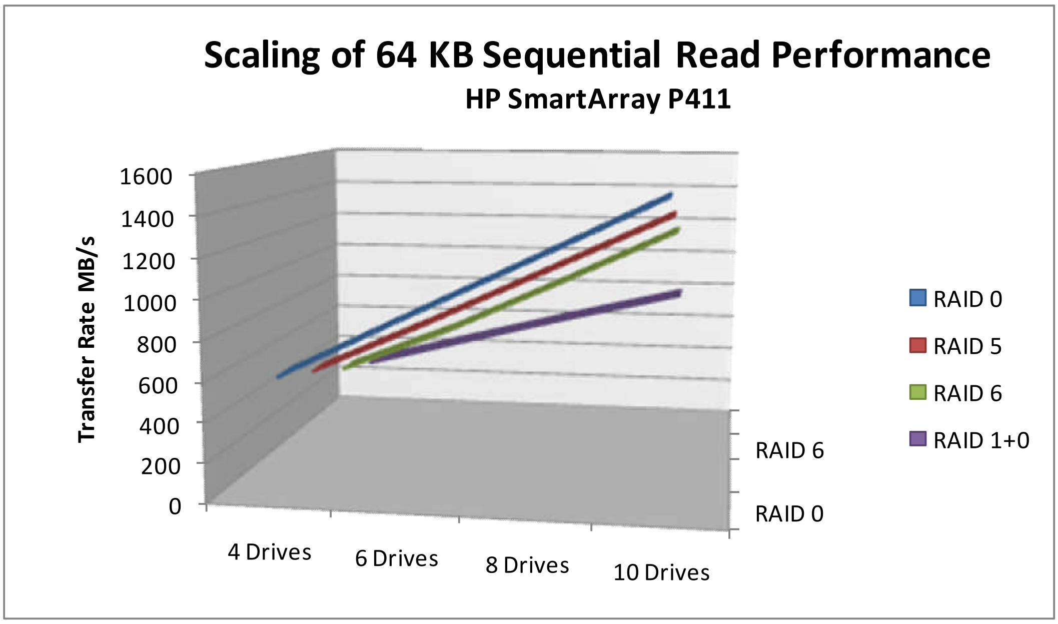
Configuration: Smart Array P411 with 512 MB cache. 256 KB strip size. Queue depth of 64. DL 380 G6

RAID 1+0 also uses striping, and its performance scales linearly with the drive count. Because it is mirrored as well as striped, RAID 1+0 requires two physical disks to achieve the same net increase in data storage capacity as a single additional disk does for RAIDs 0, 5, or 6.

Sequential read performance

With Drive Arrays, sequential read performance also increases as the number of drives in the array is increased. The upper limit on sequential performance of any one drive is determined by the maximum throughput capability of the drive (Table 1). With Smart Array controllers, the sequential read performance of an array also tends to scale directly with the number of drives in the array (Figure 4). RAID 1+0 performance scales more slowly since the striped data is distributed across fewer physical drives as a result of mirroring. With larger drive arrays, the ultimate limiters of sequential read performance are either the aggregate bandwidth of the SAS links themselves or the PCIe bandwidth, whichever is smaller.

Figure 4. Sequential read performance for RAID 0, RAID 5, RAID 6 and RAID 10 (1+0) as function of number of drives.



Configuration: Smart Array with 512 MB cache. 256 KB strip size. Queue depth of 64

RAID levels, drive count and write performance

Write operations are much more complex than reads for most drive array configurations. This complexity also has a significant effect on overall write performance. With drive arrays, all RAID levels other than RAID 0 provide some level of data redundancy and recovery. This redundancy is essential to the Smart Array controller's ability to rebuild a logical drive and recover the data when one or more physical drives in the array fails. This capability comes at a price, which is an increase in the number low-level reads, writes, and calculations that the Smart Array controller must execute when performing a high-level "write" to a logical drive in any of the redundant RAID levels.

Write performance for RAID 0

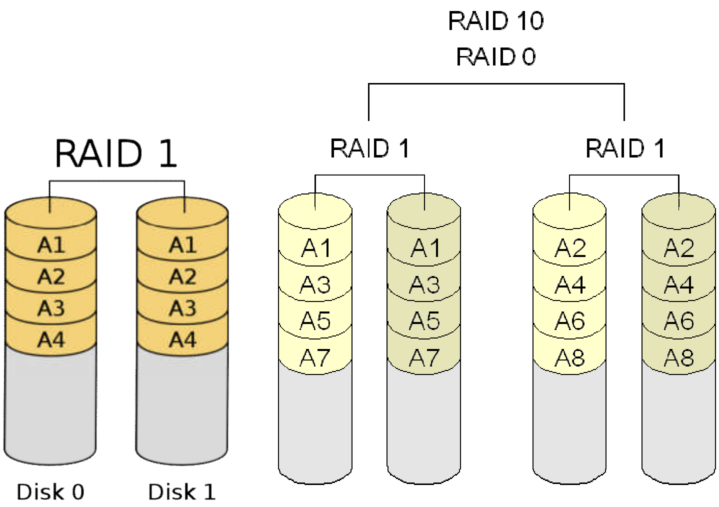
RAID 0 is the only RAID level that does not support any data redundancy. As a result, no extra low-level commands are required to execute a "write" to a logical drive. Because striping distributes the data across the physical drives, the low-level reads and writes can be executed partially in parallel. For RAID 0, both sequential and random write performance should scale as the number of physical drives increases. RAID 0 provides a useful basis for comparison when evaluating higher RAID level performance.

Write operations for RAID 1 and RAID 10 (1+0)

RAID 1 is the simplest example of the additional write overhead associated with redundant RAID levels. In RAID 1, data is simply mirrored across a set of drives (Figure 5). This means that for every "write" of a block of data to a logical drive, the Smart Array controller must execute 2 low-level writes, one to each of the mirrored drives. In a simple non-cached example, this would mean that in the worst-case scenario, write performance could be one-half that of writing to a non-arrayed physical drive. With RAID 1 there is no striping. This reduces the array controller's ability to execute reads and writes in parallel across multiple physical drives, which results in lower performance than RAID 0.

With RAID 10 (RAID 1+0), data is still mirrored; however, it is also striped across the mirrored drive sets to evenly distribute the data across the drives and provide better read and write performance. RAID 10 requires executing two low-level disk writes for each high-level write to the logical drive.

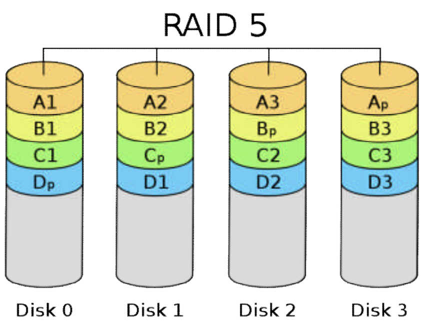
Figure 5. RAID 1 and RAID 1+0 drive arrays



Write operations for RAID 5 and RAID 6 levels

RAID 5 provides data protection by creating a "parity strip" that is mathematically calculated based on the values stored in the corresponding data strips that comprise an entire data stripe across the array of physical drives. RAID 5 requires the equivalent of one physical drive for storing the parity information for a logical drive array. As shown in Figure 6, the position of the parity strip is actually rotated with each stripe in order to balance overall performance. With RAID 5, an array of N number of drives can store N - 1 drives' worth of data. Any single drive can fail and the data it contained can be mathematically reconstructed from the other drives.

Figure 6. Configuration of a RAID 5 drive array



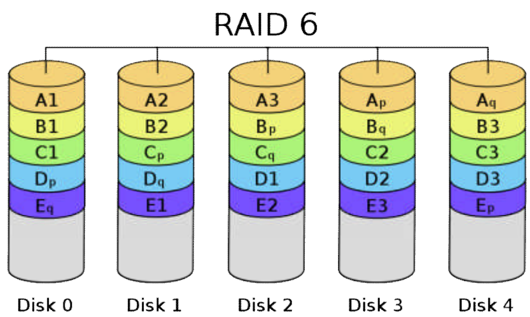
With RAID 5, each high-level write operation to the logical drive takes several lower level operations to accomplish. As Table 2 shows, each RAID 5 write takes four low level disk operations and a parity calculation. In the worst case, RAID 5 random write performance could be only one-quarter that of a single RAID 0 drive.

Table 2. Breakdown of a RAID 5 high-level write operation

Low-level operation	Purpose
Read data drive	Retrieve current data
Read parity drive	Retrieve current parity info.
Compute new parity	Based on current data and parity plus new data
Write data drive	Write new data values to the data drive
Write parity drive	Write new parity values to parity drive

RAID 6, also known as Advanced Data Guarding (ADG), calculates two independent forms of parity check data, creating two parity strips as part of each data stripe across the physical drives in the array (Figure 7). With RAID 6, an array of N drives can store N – 2 drives of data. Any two drives can fail and the data in the array can still be mathematically reconstructed.

Figure 7. Configuration of a RAID 6 drive array



With RAID 6, the write penalty is even greater than with RAID 5 because each high-level write operation to the logical drive potentially requires executing six low-level disk read/write operations and 2 separate parity calculations. In the worst case, random write performance for a RAID 6 logical drive would be one-sixth that of an equivalent RAID 0 logical drive.

Write cache, the Smart Array processor, and RAID write performance.

Write caching and the advanced algorithms used by the Smart Array processor to manage the write process are essential to delivering acceptable write performance for drive arrays when using any of the redundant RAID levels. The significant write performance penalty incurred without write caching is one of the reasons that the Zero Memory versions of the Smart Array controllers only support RAID 0 and RAID 1.

Write cache allows the Smart Array controller to store pending write commands issued by the server's operating system. The Smart Array processor then analyzes the pending queue of write commands and determines if there are more efficient ways to execute them to improve performance. It does this by employing the write coalescing and command reordering techniques discussed in the section on Smart Array write cache.

The Smart Array controller also takes advantage of a technique known as full stripe writes. If the controller determines that a full stripe of data is changing—possibly as a result of write coalescing—then on RAID 5 and RAID 6 operations it no longer needs to perform the additional read operations to retrieve the current data and parity information. All of the information required is already in the controller cache. It simply calculates the new parity values and then writes out the new stripe, including the parity strip(s).

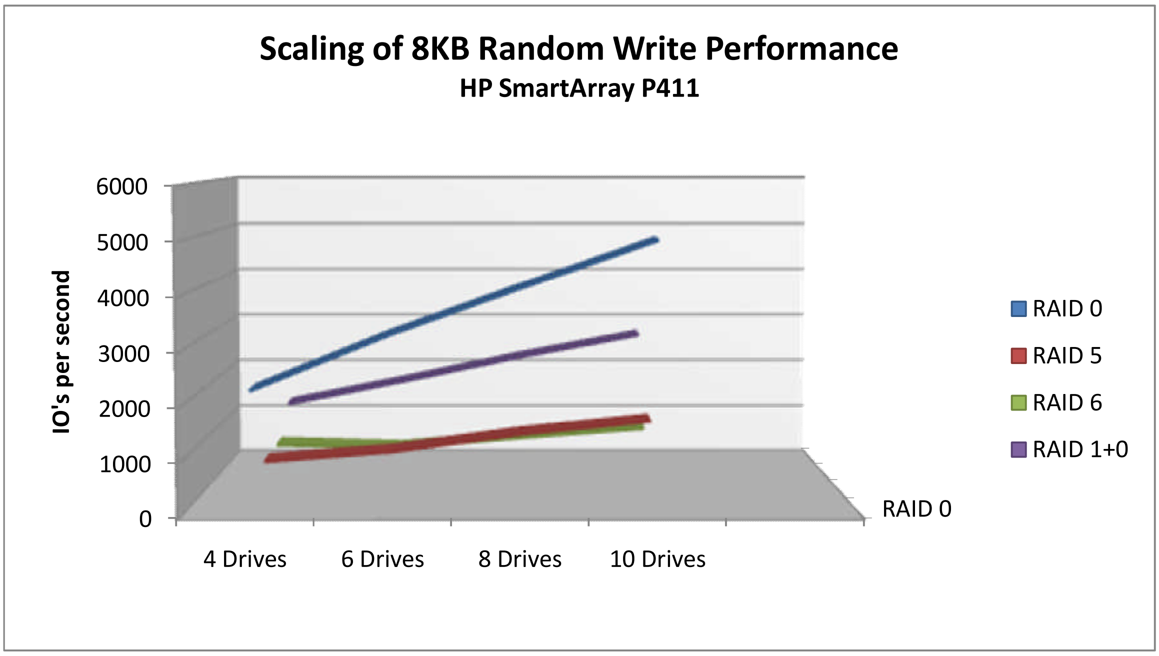
Using a larger strip size for an array decreases the number of full stripe writes that the controller will accumulate, and therefore may negatively affect write performance to a certain degree. This is because larger strips will naturally result in larger stripes and thus lower the probability that write coalescing will accumulate a full stripe of data in the controller cache. Larger strip sizes do tend to improve read performance.

Random write performance

Figure 8 compares the random write performance of RAID 0, RAID 5, RAID 6 and RAID 1+0 arrays (configured as one logical drive) as the number of physical drives is increased. As predicted, the write performance of RAID 5 and RAID 6 arrays is significantly lower than that of RAID 0 because of the overhead involved with each high level write operation. Performance does scale as the number of drives increases, although not at quite the rate for RAID 6 as for RAID 0.

For the same number drives, RAID 1+0 random write performance is about one half that of RAID 0 and about twice that of RAID 5 or RAID 6. This is consistent with the fact that RAID 1+0 requires two low level disk writes for each high level array write, but does not require any extra reads or parity calculations on the part of the Smart Array controller.

Figure 8. Scaling of 8 KB random write performance for RAID 0, RAID 5, RAID 6, and RAID 1+0.



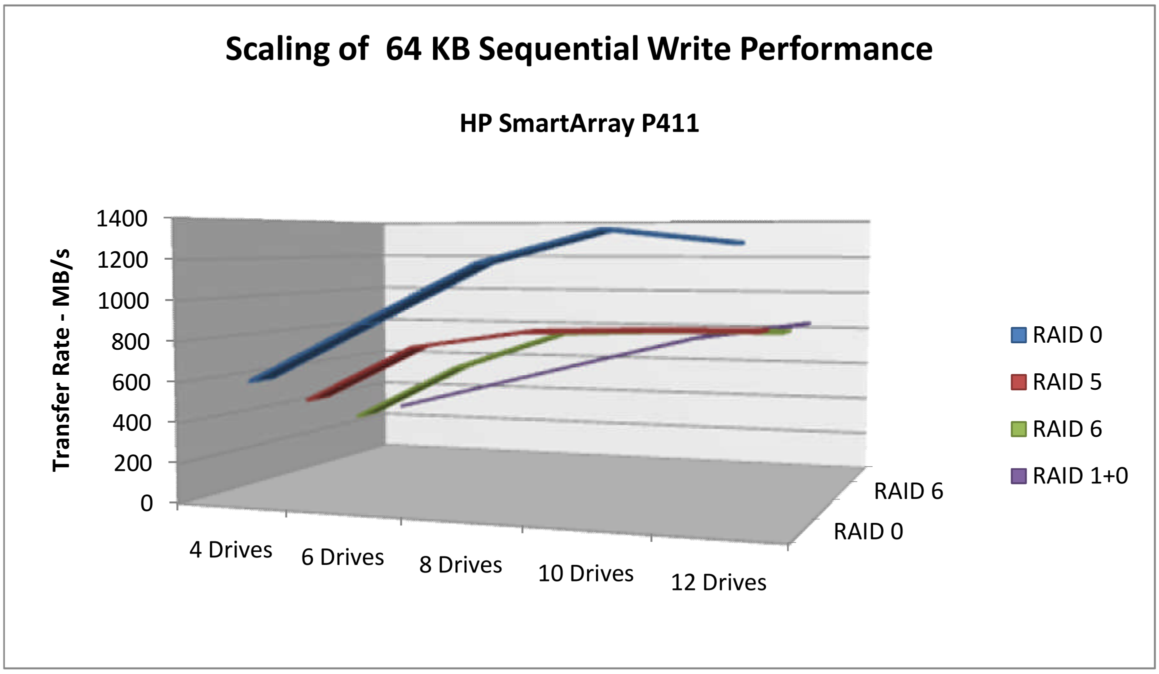
Configuration: P411 controller. 512 MB cache. 256 KB strip size. Queue depth of 64

It is important to note that while the relative random write performance is much more significantly impacted by RAID levels than random read performance, the write cache does help increase random write performance overall. This is best exemplified by RAID 0, which has no write penalty. A ten drive RAID 0 logical disk performs 5015 random writes per second while achieving only 2936 random reads per second. This difference is primarily attributable to the benefits of the write cache.

Sequential write performance

Figure 9 compares the write performance of the different RAID levels when executing 64 KB sequential writes. Compared to random writes, there are two noticeable differences in the performance curves. With sequential writes, the difference in performance between RAID 0 and RAID 5 or RAID 6 is not nearly as great as it was for random writes. This can be attributed to the write cache, and more particularly to write coalescing. Sequential writes allow the Smart Array controller to coalesce them into full stripe writes. For RAID 5 and RAID 6, this eliminates the additional read operations normally required and therefore increases their performance relative to RAID 0. Secondly, sequential write performance does not tend to scale as the number of physical drives in the logical array increases past a certain point. With RAID 5 and RAID 6, this plateau occurs when the controller processing engine reaches the limits of its ability to perform the required XOR computations. For RAID 0, performance plateaus when the maximum throughput that the drives can maintain is reached. In the test shown in Figure 9, increases in total throughput tend to diminish once the drive count exceeds eight.

Figure 9. Scaling of sequential write performance for RAID 0, RAID 5, RAID 6 and RAID 1+0



Configuration: ProLiant DL360 G6. Smart Array P411 controller. 512 MB Cache. 15K 6Gb SAS. 256 KB stripe size. Queue depth of 64)

Additional RAID performance characteristics

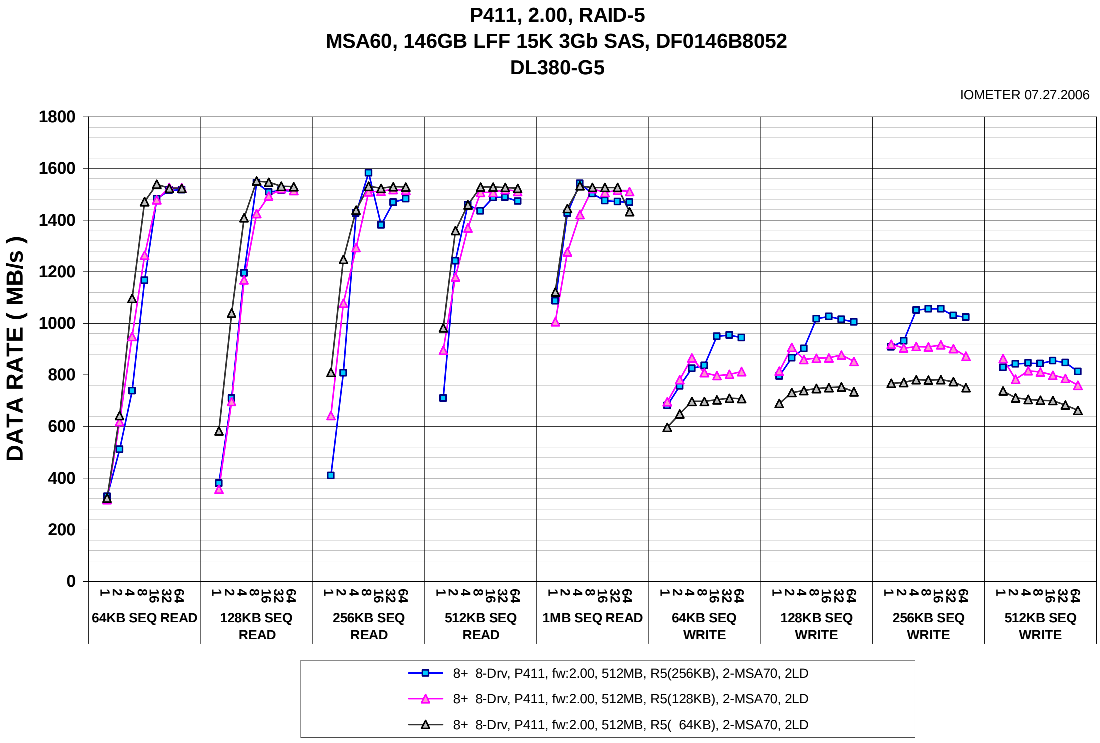
Many different terms and metrics are used in characterizing the performance of Smart Array RAID logical drives. Queue depth, throughput and latency are often referred to in RAID benchmarking tests and need to be understood in relationship to each other.

Queue depth

Array performance benchmarks are often run at varying queue depths. It is important to understand that in normal use, queue depth is not a configurable parameter. RAID benchmarking tests can artificially control the queue depth in order to simulate the effects of controller queue depths growing or shrinking under an application load as shown in Figure 10, a typical RAID benchmark suite.

In actual operating environments, the queue depth, at any given moment, represents the number of pending disk commands that the Smart Array controller has accepted from the operating system but has not yet completed to disk. The controller can analyze the commands in the queue to find more efficient ways to execute them and increase overall throughput for the Smart Array controller.

Figure 10. Typical array benchmark suite run at varying queue depths and array sizes



Throughput versus latency

The Smart Array controller uses various techniques to increase data throughput as queue depth increases. However, increasing queue depths are an indication that the Smart Array controller is falling behind in processing the disk commands from the operating system and applications. As queue depths increase, latency—the time the OS or application sees it take to complete a disk request—tends to increase. This situation can sometimes be influenced by the Smart Array controller, itself. The very tools that the controller uses to maximize data throughput—command coalescing and reordering—can increase the overall variability of latency. Applications requiring lower and/or consistent latencies need environments where queue depths remain low. In general, large queue depths against the Smart Array controller can indicate a potential controller and disk IO bottleneck, which can possibly be addressed by adding more drives to the arrayed logical disk.

For more information

For additional information, refer to the resources listed below.

Resource description	Web address
HP Smart Array Controller technology – Technology Brief	http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00687518/c00687518.pdf
Performance factors for HP ProLiant Serial Attached Storage (SAS) – Technology Brief	http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01460725/c01460725.pdf
RAID 6 with HP Advanced Data Guarding technology – Technology Brief	http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00386950/c00386950.pdf

Call to action

Send comments about this paper to TechCom@HP.com

TC100501TB, June 2010