hp storage

june 2002

technical white paper

introduction to hp StorageWorks virtual array performance analysis

executive summary

Among the many benefits provided by HP StorageWorks Virtual Array (VA) storage technology is improved efficiency and productivity of storage administration. This is the result of virtualization and automation features in the VA. The VA provides information about its operation in the form of performance metrics saved in the performance log. The performance metrics can be used to observe and analyze the performance and behavior of the VA and build confidence that the benefits claimed for virtualization are being realized. Tools and methods for access and analysis of these metrics are presented in a context of disk array performance analysis as it is applied to the VA.

contents

references

A number of technical white papers are available from HP that provides background information on HP StorageWorks Virtual Array storage technology. It may be helpful for the reader to have some familiarity with these concepts. The papers include but are not limited to the following titles:

"HP StorageWorks Virtual Array Technology"

"Virtualization, Simplification, and Storage"

"Microsoft Exchange and HP's Virtual Storage Technology"

"Microsoft SQL Server and HP's Virtual Storage Technology"

"HP StorageWorks Virtual Arrays and SAN Virtualization"

"High Availability and Data Movement"

In addition to these white papers there is also a detailed explanation of the VA performance metrics and tools in the "Command View SDM Installation and User's Guide."

The VA white papers and product documentation are available on HP's storage web site.

Background information about RAID architecture and performance characteristics of RAID is available in the "The RAIDbook" published by the RAID Advisory Board. See http://www.raid-advisory.com for more information about this reference.

scope

This paper is written for the hands on storage administrator and HP technical consultant who do not have an extensive background in system performance analysis and need to better understand the performance and behavior of virtual arrays in their normal usage environments. Those who are evaluating HP StorageWorks Virtual Array technology for use in new applications may also have interest as a reference to the detailed information available for observing VA operation.

The main purpose is to present tools and methods for using the VA performance metrics. This is done using examples and in a meaningful context. This is not intended to be an exhaustive troubleshooting or analysis guide for every possible performance scenario. The reader is introduced to the tools, methods and context and is expected to use this information as well as their own storage system knowledge and experience in a deductive reasoning process to reach conclusions about specific situations. Some general lines of reasoning about the VA performance metric data is included and more may be added in future versions of this paper.

context

The speed at which computing and information systems perform tasks has been an important field of study throughout the history of computing machinery. Basic questions to consider when approaching any performance analysis task are:

- Who cares about performance?
- Why?
- What are the ultimate goals of the system?
- How do cost, speed and reliability relate to each other and to the work being accomplished?

Questions like this will help identify the criteria to consider in a performance analysis.

At least two categories of performance criteria are important to consider. The first is the real time required to complete a particular specific job. This is defined as the system response time. The term response time often refers to the interactive responsiveness of a system to a human user. More formally, it is the time required to complete any job at any level in the system. Response time is usually most important to individual users of the system because it is a component of the rate their work can be completed or their service provided. In addition, there may be certain large-

scale jobs in which all the resources of the system are not fully utilized but must be completed within in a certain time. System response time can become a limiting factor in the feasibility of such jobs.

A second interesting category of performance criteria is the rate at which the total workload of the system is being serviced. This is often referred to as the system throughput. System throughput is usually less of a concern for individual users but of high importance for system administrators and business managers because they are interested in maximizing the return on investment in computing resources. This leads to the objective of "right sizing" the computing environment so the computing resources are highly utilized with useful work as much of the time as possible while not sacrificing the productivity of individual users due to lengthy response times.

The size and configuration of a computing system usually reflects a compromise between response time and throughput. A system sized to provide the best response time possible to all users would normally be prohibitively expensive and would very likely be under utilized. On the other hand, too small a system will probably result in unacceptable productivity loss for the users because they spend too much time waiting for responses. Finding and maintaining a proper balance of response time and throughput is an important competitive issue for businesses.

This discussion so far has been in reference to the speed of the end user application running on a computer system. A computer system is made up of a number of components, one of which is the storage. Each component has some potential to influence the overall performance achieved by the end user application. The system configuration may not be in balance. That is to say, that some components may have more performance capability than others and the performance being achieved by the application is mostly limited by one particular component. In a balanced system configuration, most components experience similar levels of utilization and no one component is the primary performance limitation.

The balance of the system configuration can be drastically affected by the application. A "compute intensive" application is one that utilizes the computational resources of the system to a much higher degree than other resources. A "data intensive" application is one that utilizes the data transmission and storage resources to a higher degree. Since different kinds of applications may run at different times, the balance may vary dynamically. The overall level of application workload can also be dynamic so that the system has a high demand for performance at times and a low demand at other times.

It is important to consider performance information about storage in the overall context of application workload and system balance. In some cases storage system performance may not be relevant because the overall demand is low or because a different component in the system is the limitation. In other cases, the storage system may be the single most important factor in determining the speed of the application.

The focus of this paper is performance information as it is measured at the external interface ports of virtual array storage devices and as it is measured by the internal VA performance metrics.

definitions

The previous context discussion used some important terms in an informal way. This section will formalize the definitions of several important terms and explain the underlying concepts.

application

The particular task or tasks being accomplished by the computer system. The application is often supported by one or more application software programs that implement many of the functions of the application. However, the application is not the software alone. It is the interactive usage of the application software to accomplish business objectives.

workload/demand

The amount and type of activity in a computer system requested by the application. The workload can be measured at different points in the system and has different characteristics and units of measure based on where it is measured.

The workload of a system can be described at a high level by the class of application and the number of users. For example: a mail server with 400 mailboxes. The workload is specifically the rate at which individual requests for activity (tasks, jobs, transactions) arrive at a particular point in the system. This is to be distinguished from the rate at which those requests are being completed. Demand is synonymous with workload but carries an emphasis that it is the rate of requested activity being measured, not the actual throughput being achieved.

stream/process

A serialized sequence of workload activity created by a background software process or by an interactive user session. A new job in the sequence will not be requested until the previous one is completed. This term is most commonly applied in a storage context to the workload of a storage device when that workload is sequential but use of the term is not limited to that concept.

open loop workload/demand

A class of workload in which the creation of new jobs is not dependent on the completion of previously requested jobs. The rate of job arrival has no relationship to the rate of job completion. Open loop workloads can occur when the potential number of system users is very large such as in a network server application. In this case, it is the unbounded arrival of users that can create the open loop characteristic rather than the workload created by each user. If the open loop workload at a particular point in a system exceeds the throughput for a sustained period of time, the number of job requests waiting at that point in the system will accumulate without bound and some kind of interruption in system operation (such as a timeout) may occur.

closed loop workload/demand

A class of workload in which the creation of new jobs is dependent on the completion of previously requested jobs. This type of workload will occur when there is a limited number of processes each of which is creating a serialized sequence of activity. The processes operate such that a new job in the sequence will not be requested until the previous one is completed. The workload is limited to be the number of processes that are concurrently in operation. A process corresponds to either a background software process or an interactive user session.

System operation in response to a closed loop workload is self-limiting. That is why it is called "closed loop." The demand will be throttled by the lowest point of throughput in the system. All components of the system will have the same measured throughput because the demand is being limited to be the lowest throughput of all the components. Therefore, it is difficult to discover by measurement which component in the system is limiting throughput because all components in the system will have the same measured throughput. It may appear that performance of a closed loop workload is low because the application is not creating a high demand but some other component of the system may actually be the limiting factor.

workload intensity

The number of processes in a closed loop workload.

performance benchmark

A software program and testing procedure that creates a simulated workload. The goal of a performance benchmark is to provide a common basis for performance comparison between different system configurations or products. There is a wide variety of performance benchmarks and they create a wide variety of workloads. Some performance benchmarks attempt to simulate as faithfully as possible the workloads created by particular classes of applications. Others focus on demonstrating the maximum performance characteristics of particular system components such as the maximum data throughput of a storage device. The workload created by a performance benchmark can approximate the workload created by a real application. In some cases, it is a poor approximation. This can lead to false conclusions about expected performance of the real application. The uncertainty can sometimes be addressed by using the real application in a non-production mode as the benchmark. Even then, the nature of the real application may change over time so that the original benchmark results no longer represent actual performance expectations. Performance benchmarks are an important tool in system performance analysis and need to be used carefully.

response time

The real elapsed time from the arrival to the completion of a particular job at a particular point in the system. The response time includes queue waiting time that may occur because the resources needed to complete the job are unavailable at job arrival plus the actual time required by the hardware to perform the task.

service time

The real elapsed time required by the hardware to perform a particular task. The service time is the response time less the queue waiting time. Some applications require storage system response time to be near the service time for best application performance.

throughput

The overall rate at which work is being completed at a particular point in the system. Units of throughput measurement that are typically used in a storage context are MB/sec (mega-bytes per second), IO/sec or IOPs (input/output operations per second) and TPS (transactions per second).

MB/sec

Mega-bytes per second. A unit of measurement that is often applied to a storage device to measure demand or throughput. It measures the rate at which data is transmitted to and from the storage device. The measurement is often qualified by some characteristics of the workload such as reads or writes or a combination of reads and writes of a certain length. MB/sec throughput is usually measured using a large I/O request size (64k bytes or above) because data transfer bandwidth tends to dominate the performance characteristics of a storage device for this type of workload. However, MB/sec throughput can be measured and can have meaning for any type of workload.

IO/sec (IOPs)

Input/output operations per second. A unit of measurement that is often applied to a storage device to measure demand or throughput. It measures the rate at which read and write requests are arriving at or being completed by the storage device. The measurement is often qualified by some characteristics of the workload such as reads or writes or a combination of reads and writes of a certain length. IO/sec throughput is usually measured using a small I/O request size (2k to 16k) because I/O command processing time tends to dominate the performance characteristics of a storage device for this type of workload. However, IO/sec throughput can be measured and can have meaning for any type of workload.

TPS

Transactions per second. A unit of measurement that is usually applied to a database management system to measure demand or throughput. It measures the rate at which database transaction requests are arriving at or being completed by the database management system. This is a relevant measurement for storage because storage is often an important factor in the performance of database management systems.

sequential

A particular type of storage workload in which the addresses and lengths accessed by a sequence of requests specifies in combination a contiguous range of data. This is an important type of workload because many storage devices implement optimizations to accelerate performance of sequential workloads. Data access software such as file systems and databases often include algorithms that attempt to generate sequential workloads to take advantage of the storage device optimizations. It is also representative of the workload created by certain kinds of applications.

multi-stream sequential

This is a workload which comprises multiple, concurrent sequential workloads. Typically, each sequential stream in the workload is created by a separate process. It can be difficult to recognize the individual sequential streams when they are interleaved at a single measurement point. Without careful observation, a multi-stream sequential workload can appear to be a random workload.

random

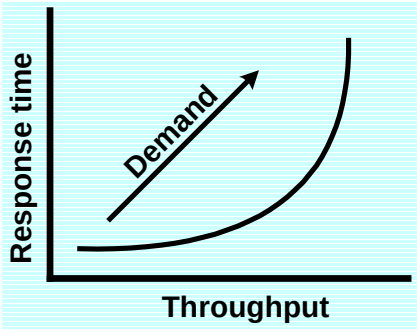
A particular type of storage workload in which the addresses specified by a sequence of requests appears to be randomly distributed over a particular range of addresses. This is an important type of workload because it tends to defeat optimizations implemented in storage devices to accelerate performance in response to a sequential workload. It can be representative of the workload created by multiple, independent users of a system. Even if the workload created by each individual user is sequential, the combination of all the users can appear to be random at a storage device.

utilization

The degree to which a particular component or resource in a system is used. It is measured as an average over a certain period of time and is expressed as a percentage of component utilization over that time period. Zero percent means no use over the measurement period. 100% means fully busy for the entire measurement period. Near 100% utilization indicates a performance constraint in the system. High utilization and high queue depth both are indications that a component or resource is near saturation.

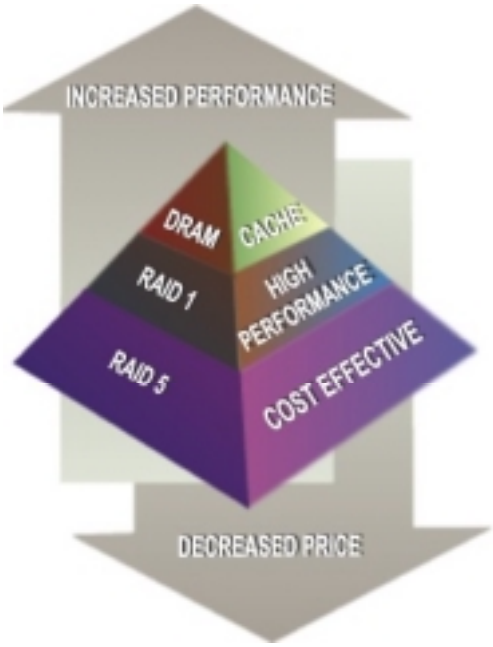
saturation

A condition in which the demand on a particular component or resource in a system is equal to or greater than the throughput of that component or resource. The condition is characterized by asymptotically increasing response time as the demand increases with no further significant increase in throughput. The increase in response time is due to queue waiting time. The saturated component or resource will have near 100% utilization and will have a relatively deep queue.



storage hierarchy

An architectural arrangement of storage methods such that the faster, higher cost methods are closer to the application and the slower, lower cost methods are farther away. The storage methods are arranged in rank according to speed and cost. The goal of a storage hierarchy is to provide storage that appears to an application to have the speed of the more costly methods and the cost of the less costly methods. HP's Virtual Array products include AutoRAID technology which implements a storage hierarchy using DRAM cache memory, RAID 1+0 and RAID 5DP.



working set

A concept that refers to address locality in the workload. The working set comprises a set of data addresses that have been accessed over some measurement period of time. The working set size is usually compared to the storage capacities at various levels in a storage hierarchy. If the working set fits into one of the levels, it will stay resident at that level and the performance of the storage method at that level will be realized over the measurement period of the working set.

thrashing

A condition in which the working set does not fit into a particular level of a storage hierarchy. The result is that the data contents at that level are constantly changing and the performance of that level is not realized.

typical storage system workloads

When analyzing storage system performance it is important to have at least some information about the demand presented to the storage system by the application. The following table provides high level characterization of demand for various classes of application that can be used in the absence of better information. This information is based on experience and knowledge of applications accumulated over time by HP storage engineers.

Weighting Scale: 1 = lowest and 5 = highest level of demand

	DSS	CRM	CAD	FM	ERP	DIM	Email	SD	CM	Desktop
Read Demand	4	3	1	3	3	1	5	3	5	5
Write Demand	2	1	2	3	2	5	4	4	1	4
Concurrent Clients	4	4	3	1	4	1	5	2	5	5

Note: Read and Write Demand are rated on a scale from 1 to 5 and are not dependent or related to each other.

DSS= Decision Support Systems

CRM= Customer Relationship Management

CAD= Computer Aided Design

FM= Forecast and Modeling

ERP= Enterprise Relationship Management

DIM= Document Image Management

Email= Electronic Mail

SD= Software Development

CM= Content Management

Desktop= Word Proc., Spreadsheets, etc.

disk array performance basics

It is important to consider basic information about RAID architecture and RAID performance characteristics as they relate to HP StorageWorks Virtual Array technology when analyzing virtual array performance. A full description of RAID architecture is available in "The RAIDbook" reference cited above. Following is a brief treatment of the subject.

why disk arrays?

The capacity, availability and performance requirements of applications historically have exceeded the capabilities of individual disk drives. Coupled with commoditization of small form factor disk drives this has led to the wide spread use of disk arrays. The main values being added by disk arrays over the direct use of multiple disks by an application are manageability and modularity of function. For example, suppose an application has a requirement to store a single data file that is larger than any available disk drive. The application software could be modified so that it can split up the data file across multiple disk drives in an application specific way. With that approach, every application that had this requirement would need to be modified. Disk arrays provide a layer of virtualization above disks that presents composite capacity, availability and performance characteristics. With disk arrays, applications, system administrators and users need not be concerned with the details of combining multiple disk drives to form "virtual" disks.

A disk array in the broad sense is any combination of multiple disk drives whose access and management is presented to higher levels in the system through some form of virtualization software. The virtualization software can take many forms including: file systems, volume managers, databases, storage device drivers, firmware on storage bus adapters and firmware in dedicated disk array controllers. There is great advantage to locating the disk array software on a dedicated disk array controller as in the implementation of HP's Virtual Array storage products because it can be tightly coupled to the array hardware and specifically optimized for control of a storage array.

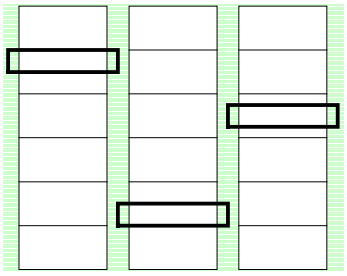
disk striping

One of the simplest ways to form a disk array is to use a striping algorithm to spread data across multiple disks. This provides higher capacity since the capacity of all the disks are added together. It has the potential to provide higher performance because all the disks can possibly be used in parallel to service the workload. The performance benefit is realized only if the combination of the workload and the array management algorithms actually result in parallel disk operation. A number of methods are available to generate parallel disk operation.

shallow striping

With shallow striping, the size of the block of data placed on each disk in a stripe (the stripe block) is small relative to the read/write request size prevalent in the workload. A single read/write request from the workload corresponds to a sequential set of stripe blocks in the striping sequence, each from a different disk in the stripe. This has the effect of multiplying the data transfer rate by the number of disks on average used to service a request. This mode of operation is usually associated with the RAID level 3 and with applications that have a high sequential MB/sec demand although it is not a formal part of the RAID level 3 definition. This kind of striping is not implemented by HP StorageWorks Virtual Array products but is described here for completeness.

deep striping



With deep striping, the size of the block of data placed on each disk in a stripe (the stripe block) is large relative to the read/write request size prevalent in the workload. Most read/write requests are contained within a single stripe block on a single disk in the stripe. This does not effectively generate parallelism in disk operation unless there is concurrency in the workload as it appears at the disks. The concurrency can be inherent in the workload of the application due to multiple processes in the application or can be generated by data caching and management algorithms in the disk array. With the proper level of concurrency in the disk workload, deep striping has the effect of multiplying the I/O rate by the number of disks in the array. This mode of operation is usually associated with the RAID levels 0, 0/1, 4 and 5 and with applications that have a high random IO/sec demand although it is not a formal part of the definitions of those RAID levels. HP StorageWorks Virtual Array products implement deep striping as RAID 1+0 and RAID 5DP.

A closed loop, single process application does not in itself generate a workload that contains concurrency. Some applications are of this nature by design because that is the true nature of the workflow process implied by the application. Others are this way because storage performance implications have not been considered to any large degree in the application software design. Perhaps of less importance but worth noting is that some performance benchmarks also behave this way by default or by configuration. This type of application will not realize the performance benefit of deep striping unless some other means is used to create concurrency.

caching

Read and write caching is one means often implemented in disk arrays to enhance performance. It can enhance performance in a number of ways. The HP StorageWorks Virtual Array products implement read and write caching.

write back caching

One way that write caching can enhance performance is to create additional concurrency in the disk write workload not present in the application workload through write back caching. Concurrency is created because the application does not wait for a write operation to be completed by a disk. Rather, the write is reported as complete to the application immediately after the write data has been received and stored in the write cache of the storage device but before it is actually written to disk. This releases the application to continue and create another write request much faster than if the application had waited for completion of the disk write. In this way, an application can generate writes much faster than they can be completed by the disks and so they can be serviced in parallel by multiple disks. If the write cache becomes full, the application is forced to wait for the completion of disk writes as if a write cache were not available. However, in this case the write throughput has been increased because of parallel disk operation and other write cache optimization methods. The write cache acts like a buffer between the application and the disks to absorb temporary bursts of high write demand.

write merging

Another way that write caching can improve performance is by merging sequential writes into a single, larger write request to disk. While it would seem that this would reduce write performance because it reduces concurrency, there are actually two performance benefits. One performance benefit occurs if the merged writes would have all addressed the same stripe block on the same disk anyway. In this case, concurrency is not reduced by merging the writes and only one instance of overhead occurs instead of one for each unmerged write. Another performance benefit of sequential write merging is to reduce the number of parity updates that are needed to write the data. Disk arrays (except for RAID level 0) store both application data and redundant data (see "redundancy" section below). If writes to multiple stripe blocks in a stripe can be processed in one

operation, the parity for the stripe need only be updated once instead of once for every stripe block written. If a long series of sequential writes can be merged such that the merged write covers an entire stripe of a deeply striped array, the write data throughput (MB/sec) of the array can approach that of an array with shallow striping.

over-writes

Write caching can improve performance if the workload has a write working set that fits in the write cache. A cache over-write is a write that overlaps another write already held pending in the write cache. The overlapped portion of the pending write is replaced by the new write in the cache. This eliminates the need to post the overlapped portion of the pending write to disk. This can improve performance by reducing utilization of the disks.

write caching and data integrity

The risk to data integrity posed by write caching is usually addressed in disk array implementations with batteries. This has no direct effect on performance characteristics but it does enable the many performance benefits of write caching like write back caching.

read pre-fetch

One way that read caching can enhance performance is read pre-fetch. In read pre-fetch the read cache detects a sequential pattern of read access and anticipates the application workload by pre-fetching the next set of sequential data into the read cache before it has been requested by the application. Then when the pre-fetched data is requested, it is available immediately from the read cache rather than reading from disk. The pre-fetch can occur concurrently with reads that have already been requested creating concurrency in the disk read workload and resulting in parallel disk operation. Another name used to refer to read pre-fetch is "read ahead."

read hits in a working set

The read cache can also enhance performance if the workload has a read working set that fits in the read cache. In this case, there is a high probability that some requests will be satisfied from data that is already resident in the read cache from a previous request. This provides faster response time to the read request since it doesn't have to wait for the disk and reduces disk utilization. However, this type of hit in the read cache is far less likely than a read pre-fetch hit because the read working set is often much larger than the read cache.

interaction with upstream caching

The cache in a disk array may not be the only cache in the data access path of a system configuration. For example, file systems and databases often implement their own data cache. This is referred to as an upstream cache. Upstream caches can heavily influence the application workload as it appears at the disk array. They often implement many of the same caching algorithms implemented in the disk array cache. In many cases, the behavior of multiple caches is complimentary. It is also possible that the behavior of multiple caches is contradictory. A prime example of contradictory cache behavior is a CPU resident read cache similar in size or larger than the read cache in a disk array. The CPU read cache would catch most the working set read hits so the disk array read cache would have very few working set read hits. On the other hand, read pre-fetch in a disk array read cache can act like an extension of read pre-fetch in a CPU read cache. It is important to consider how data caching outside the disk array will impact array performance.

redundancy

Disk arrays distribute application data across disks in a way that is transparent to the application. By design intent, the application does not have visibility to the details of the layout of the application data objects on the disks. As a result, the loss of any one disk in the array could render useless the whole application data set stored since arbitrary portions of the application data would no longer be available. The availability of the application data set is dependent on the combined availability of all the disks in the array. Without redundancy, the combined availability is that of a single disk divided by the number of disks. Availability decreases as more disks are added to the array while capacity and performance grow. This is contrary to application needs that typically require capacity, performance and availability to all scale upward. Together, this results in a need for redundancy so that availability does not depend on the failure of a single disk.

performance impact of redundancy

Maintaining data redundancy in a disk array has an impact on the performance of writes but not on the performance of reads. Additional disk writes are needed so the data can be stored in a redundant form on

multiple disks instead of just on a single disk. Then if a single disk fails, the data can be recovered from redundant data stored on the surviving disks. The additional disk writes needed to maintain redundancy in some cases can proceed in parallel on multiple disks so there is not a significant impact on the response time of the write operation. Even so, the need to write to multiple disks does increase the utilization of the set of disks over the non-redundant case so there is a performance impact. In other cases, the process required to maintain data redundancy results in some serialization of disk operations. In these cases, the performance impact includes both response time and disk utilization effects.

cost of redundancy

In addition to the performance impact of redundancy, there is also a cost to yield the same virtual capacity as a non-redundant array. Additional disks are needed to store the redundant data.

RAID 1+0

In the HP StorageWorks Virtual Array implementation of RAID 1+0, the data is both striped and mirrored on the disks. Each write request issued from the write cache to the disks actually results in write operations to both the primary disks and the mirror disks. The primary and mirror write operations proceed in parallel so the performance impact of redundancy is restricted to increased disk utilization. This results in an approximate factor of two reduction in the IO/sec performance that could have otherwise been obtained from the disks without redundancy since twice the number of disk operations are required (N/2 where N = # of disks). For example, an array of 30 disks, each of which is capable of 100 IO/sec at a given response time would provide about 1500 IO/sec of write performance (30 x 100 / 2) in RAID 1+0 mode. Since the read and write IO/sec throughput of disks are approximately the same, a simplifying rule is that the IO/sec write throughput of RAID 1+0 is expected to be about 1/2 the IO/sec read throughput. Two full copies of all data are maintained in RAID 1+0 making the cost approximately a factor of two higher than a non-redundant array (2 x N where N = # of disks = required virtual capacity / capacity of a single disk).

RAID 5DP

In the HP StorageWorks Virtual Array implementation of RAID 5DP, the data is striped and two blocks of redundant data (parity blocks) are maintained with each stripe. A write request issued from the write cache that addresses a portion of a single stripe block and is not merged with any other write request is handled by the traditional RAID 5 read/modify/write process. The data from the data disk and each of the two parity disks is read into a buffer, the old data is subtracted from the two parity blocks, the new data is added to the parity blocks then the data and parity is written back to the disks. The read/modify/write sequence is serial on each of the disks so there is a response time impact to maintain redundancy. However, with write back caching in the array write cache, the response time impact is not visible to the application unless the array write cache is full because demand is approaching the saturation level. The read/modify/write sequence proceeds in parallel on each of the three disks so there is also a disk utilization impact to maintain redundancy.

The read/modify/write process in RAID 5DP results in an approximate factor of six reduction in the IO/sec performance that could have otherwise been obtained from the disks without redundancy since six times the number of disk operations are required (N/6 where N = # of disks). For example, an array of 30 disks, each of which is capable of 100 IO/sec at a given response time would provide about 500 IO/sec of write performance (30 x 100 / 6) in RAID 5 DP mode for a workload that results in mostly read/modify/write operations. Since the read and write IO/sec throughput of disks are approximately the same, a simplifying rule is that the IO/sec write throughput of RAID 5DP for this type or workload is expected to be about 1/6 the IO/sec read throughput. Two additional disk's worth of capacity are required by RAID 5DP to store the redundant data. The additional cost for redundancy in RAID 5DP is the incremental cost of the two extra disks (N + 2 where N = # of disks = required virtual capacity / capacity of a single disk).

other RAID 5 write processes

Read/modify/write is one write process that can be used in a RAID 5 implementation. There are additional write processes implemented in HP StorageWorks Virtual Array RAID 5DP and usually in traditional RAID 5 implementations that can be more efficient than read/modify/write. These processes involve the merging of multiple write requests into a single, larger operation that covers a larger portion of a stripe or even a whole stripe. The parity is updated once for the entire operation rather than once for each stripe block. In a traditional RAID 5, the ability to do this kind of merging is dependent on a sequential workload. The HP StorageWorks Virtual Array has additional flexibility to do partial and full stripe writes because AutoRAID technology allows blocks that are not sequential in the application data space to be merged into physically contiguous blocks in the RAID striping layout. This process is called "log structured writes."

Two VA performance metrics are available in the OPAQUE category to monitor the type of write activity that is occurring in RAID 5DP. "RAID 5 DP Writes in Place" count the writes performed with the read/modify/write process. "New RAID 5 DP Writes" count writes that require the allocation of new space including log structured writes.

AutoRAID addresses the performance impact of read/modify/write

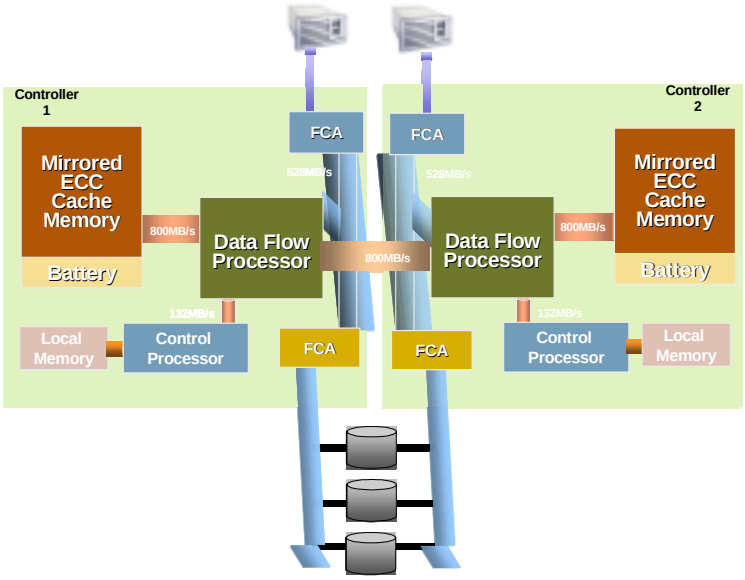
Write merging is one way to avoid the performance impact of RAID 5 read/modify/write. However, some important classes of workload cannot be merged. An example would be a small, random write workload that is typical of an online transaction-processing (OLTP) database. Historically the performance impact of read/modify/write has been one of the major drawbacks to RAID technology. AutoRAID technology in the HP StorageWorks Virtual Array addresses this issue by implementing a storage hierarchy for writes using RAID 1+0 and RAID 5DP. RAID 1+0 has higher performance for writes (approximate N/2 compared to N/6) while RAID 5DP has lower cost (approximate N + 2 compared to 2 x N). AutoRAID is designed to maintain the write working set in the higher performance RAID 1+0 and the rest of the application data set in the lower cost RAID 5DP. The application receives the approximate write performance of RAID 1+0 and the approximate cost of RAID 5 DP as long as the write working set fits in the RAID 1+0 storage area. AutoRAID reserves 10% of the usable capacity for RAID 1+0. This is based on studies that have shown the write working set is typically less that 10% of an application data set. If multiple applications on the same or different servers use a single array, the 10% rule applies to the fraction of the array usable capacity dedicated to each application data set so that the total write working set for all applications is still 10% of the array usable capacity.

AutoRAID policy observation

AutoRAID implements policies whose purpose is to maintain the write working set in RAID 1+0 at all times. Performance metrics are available to observe the operation of AutoRAID policies. The metrics can be used to determine how well AutoRAID is achieving this goal.

resources and topology

The number, capacity and arrangement of resources within a disk array comprise the basic hardware architecture and determine the underlying performance potential. The architecture coupled with the data management policies and how this is used within a particular application environment determine the performance that will be realized.



Important resources in disk array architecture are: disks, external data transmission ports, internal data transmission busses, cache/buffer memory and processor system(s) used to run the array control software/firmware. Most of the important resources except for the disks themselves are often contained in the design of an array controller module. Array implementations can allow for use of two or more controller modules to provide redundancy and additional performance. Knowledge of the architecture and performance measurements in relation to individual resources can provide important insight in array performance analysis. The diagram gives details of the HP StorageWorks Virtual Array architecture.

hp StorageWorks virtual array performance analysis tools

Command View SDM provides two facilities for accessing the virtual array performance metrics. Many (but not all) of the metrics can be accessed and displayed in graphical form from the performance page of the Command View SDM GUI. The full set of performance metrics can be accessed and displayed in character form by the Command View SDM "armperf" command line. The "Command View SDM Installation and User's Guide" is a detailed reference for usage of the performance GUI and armperf command line. In addition, the user's guide and the on line GUI help provide detailed descriptions of the performance metrics.

Note: The usage of Command View SDM referenced in this paper requires Command View SDM version 1.03 or later. Some aspects of Command View SDM were modified in version 1.03 that greatly enhances usability of the armperf output with the methods presented here. The detailed descriptions of the performance metrics did not appear in the user's guide prior to the 1.03 version.

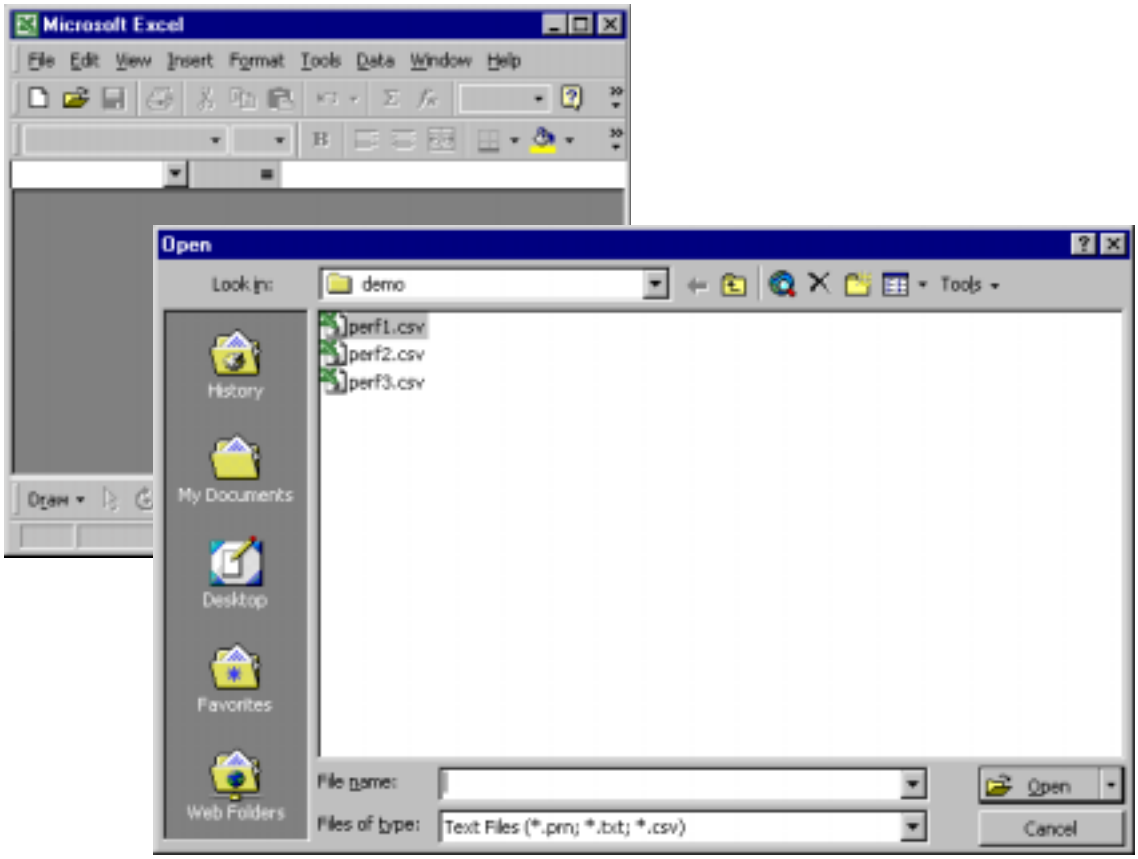
In addition to the performance GUI, another useful method to view the performance metric data in graphical form is to import the output from the armperf command line into an external data analysis and charting tool. This paper will give examples of graphical performance data presentation using both the performance GUI and Microsoft MS Excel. Microsoft MS Excel examples are given using MS Excel 2000.

importing armperf output into Microsoft MS Excel

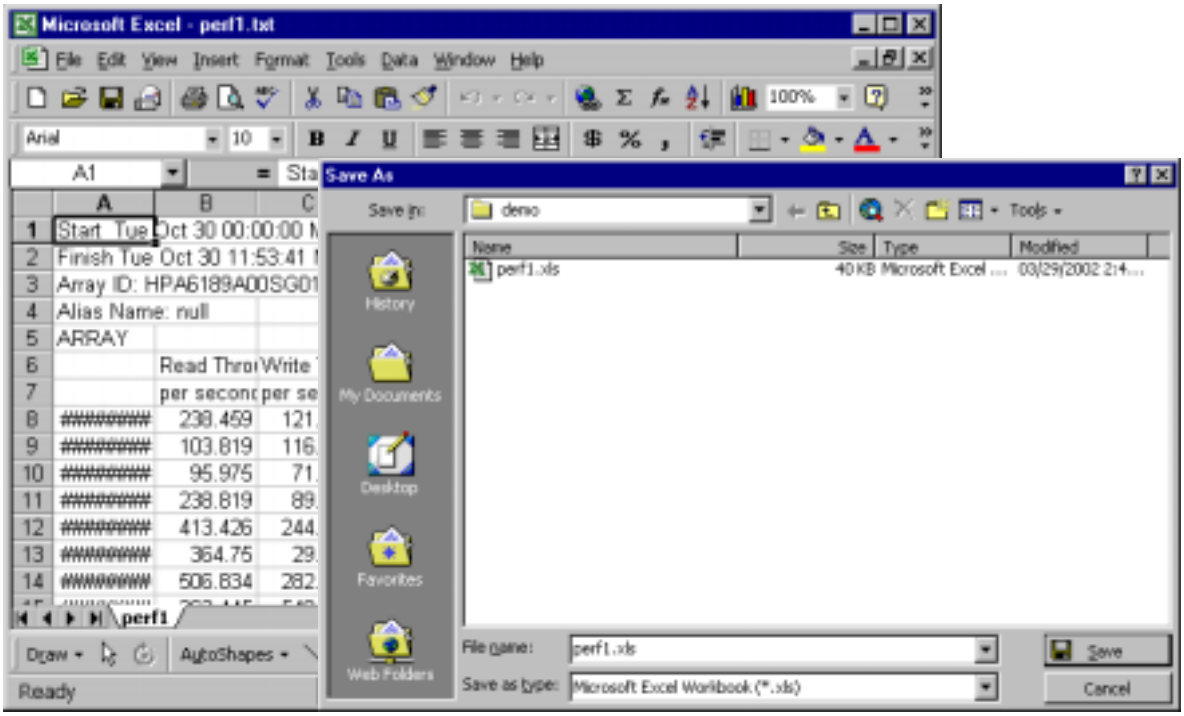
The armperf command line supports the "-x COMMA" option that produces output in comma-separated format. This is the most convenient format for import into MS Excel. To import into MS Excel, use armperf to produce a comma separated file with the desired performance data. If it is unknown which data is needed, a good place to start an analysis is with all the ARRAY category performance data for the full time range that is available:

armperf -c ARRAY -x COMMA <array-id> > perf1.csv

MS Excel recognizes the csv file type extension as a file in comma-separated format. The comma-separated file can be opened directly by MS Excel. In the open window, specify the "Text Files (*.prn; *.txt; *.csv)" file type to see all text file types.

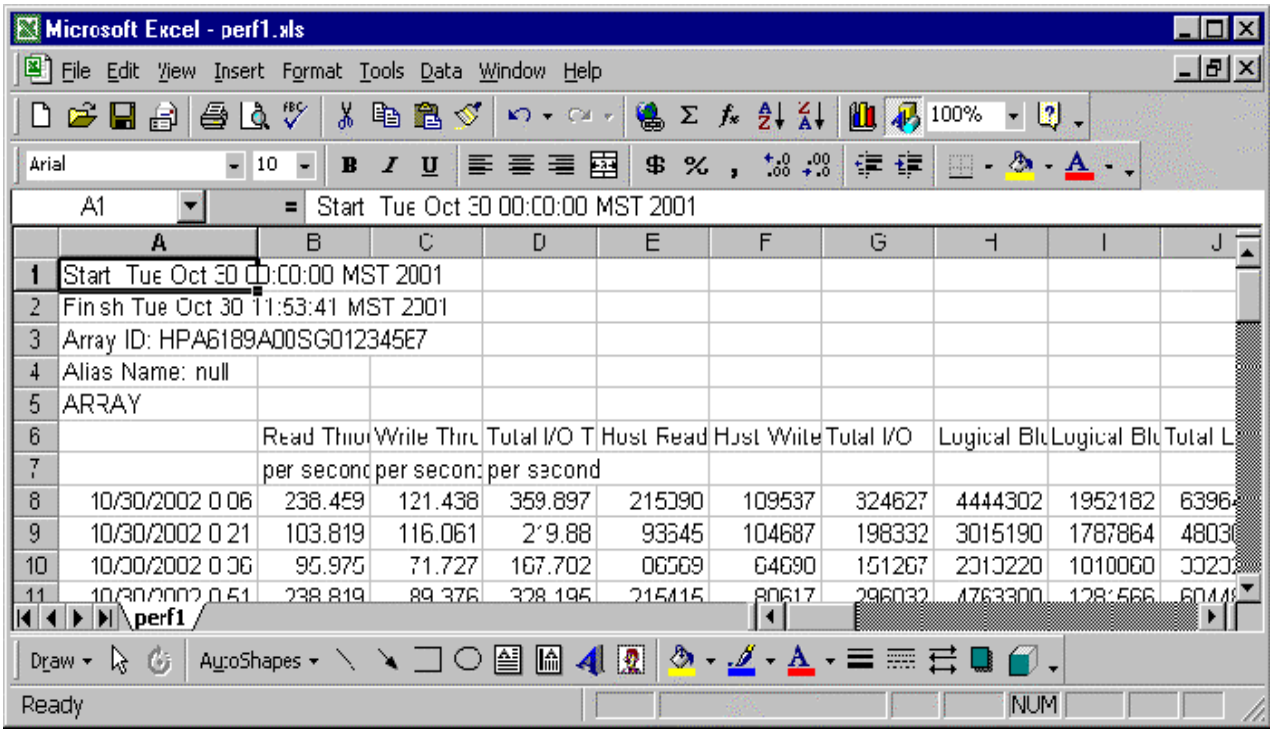


After the import is complete the data will be displayed in an MS Excel worksheet. Save the file in MS Excel workbook format.

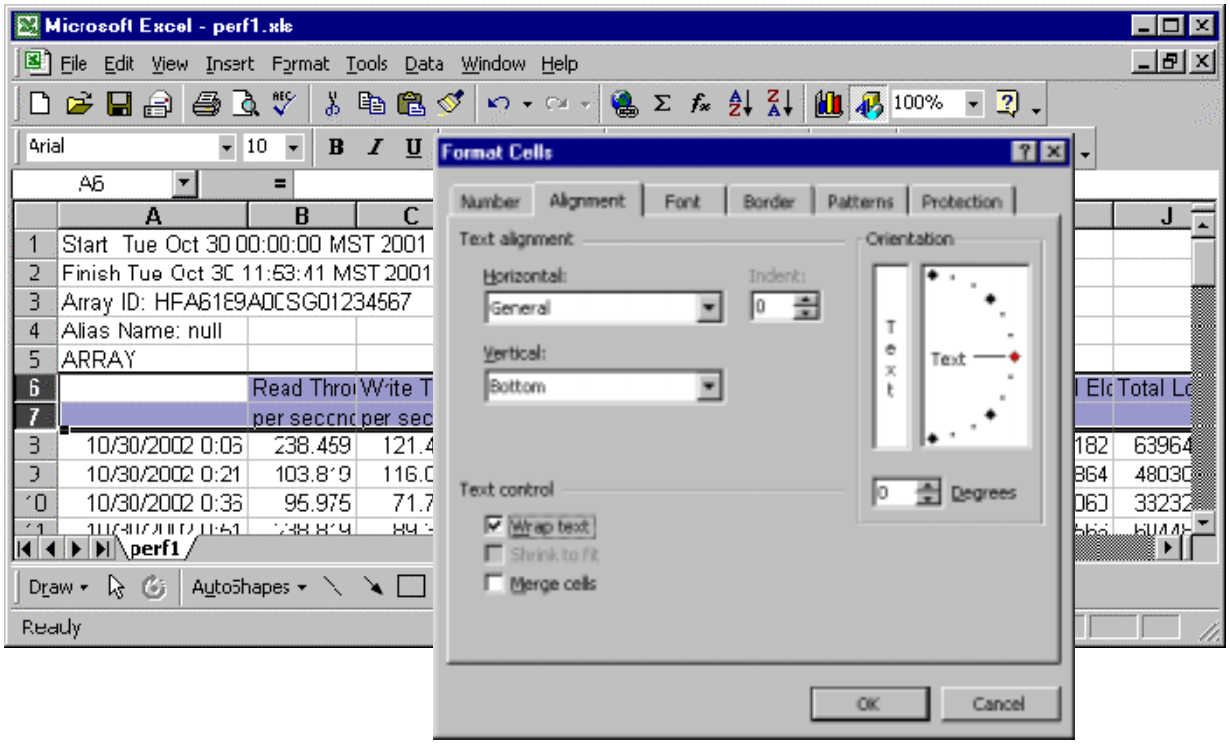


formatting tips for the data worksheet

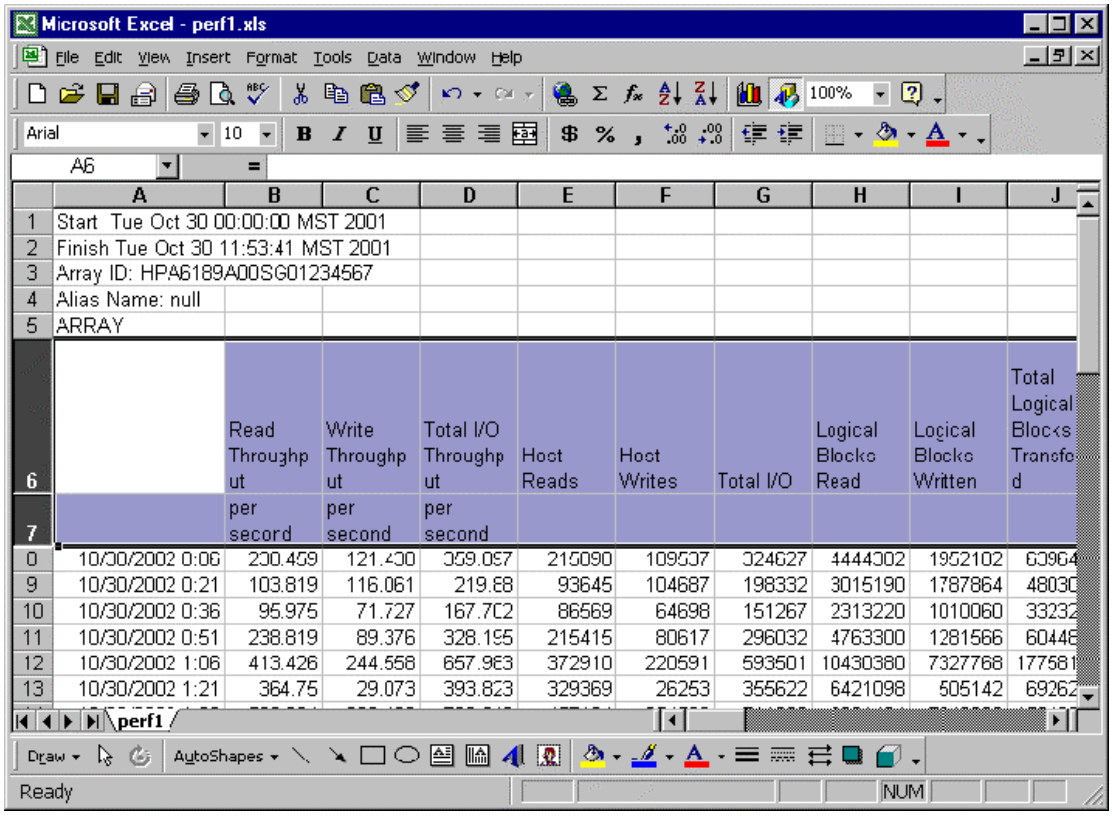
Once the performance data is imported into an MS Excel worksheet, some formatting operations will make the data more readable. Increase the width of the first column so that the time stamps are visible.

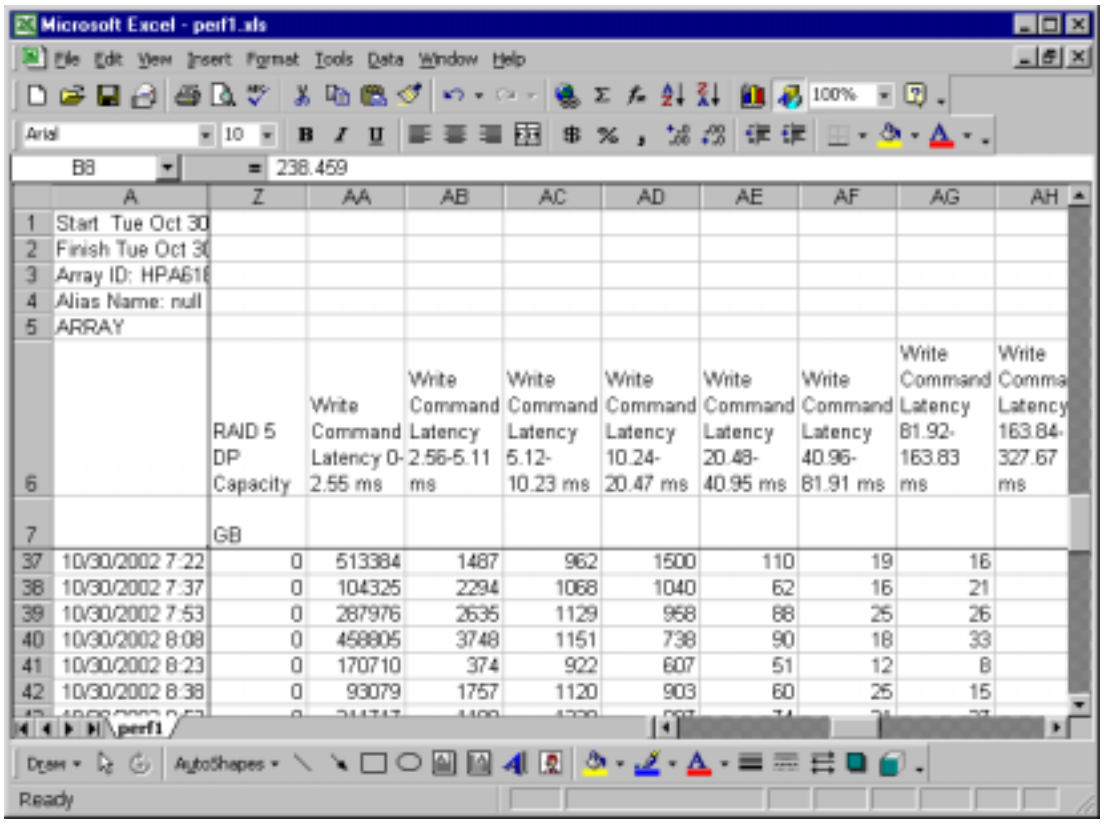


Select the two rows that contain metric names and unit names (rows 6 and 7 in the example) and format the cells so that the text wraps. This makes it easier to read the names.



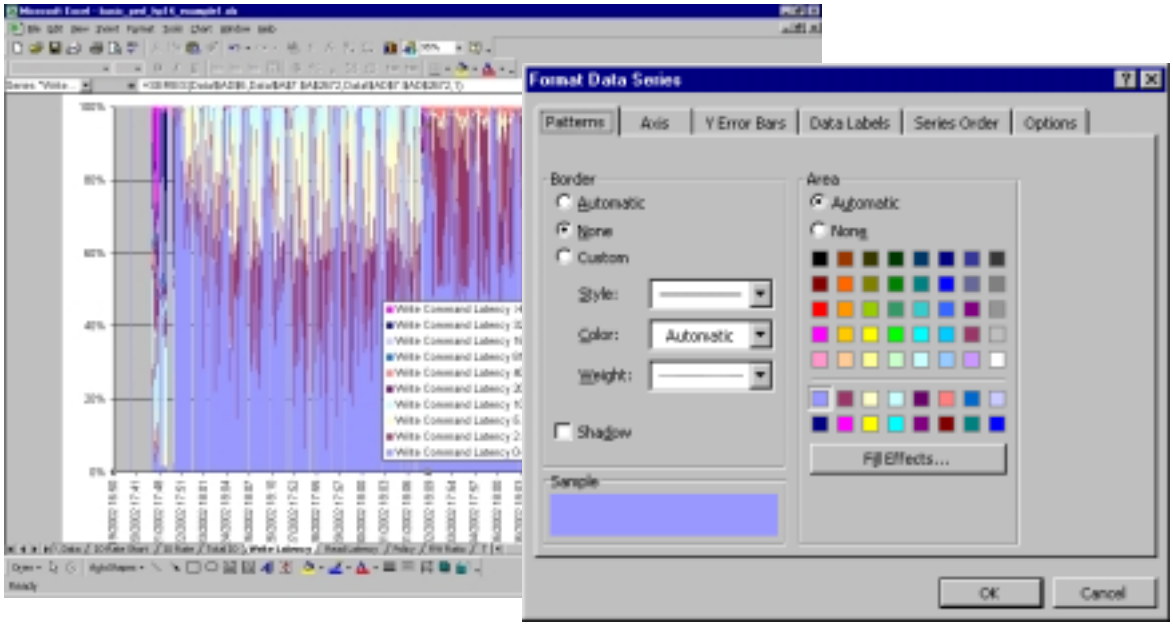
Select the cell just below the unit names row and just to the left of the time stamps (cell B8 in the example). Select Window->Freeze Panes. This locks the rows above and the columns to the left of this cell so that the names and time stamps remain visible even when the scroll bars are used to browse the data.





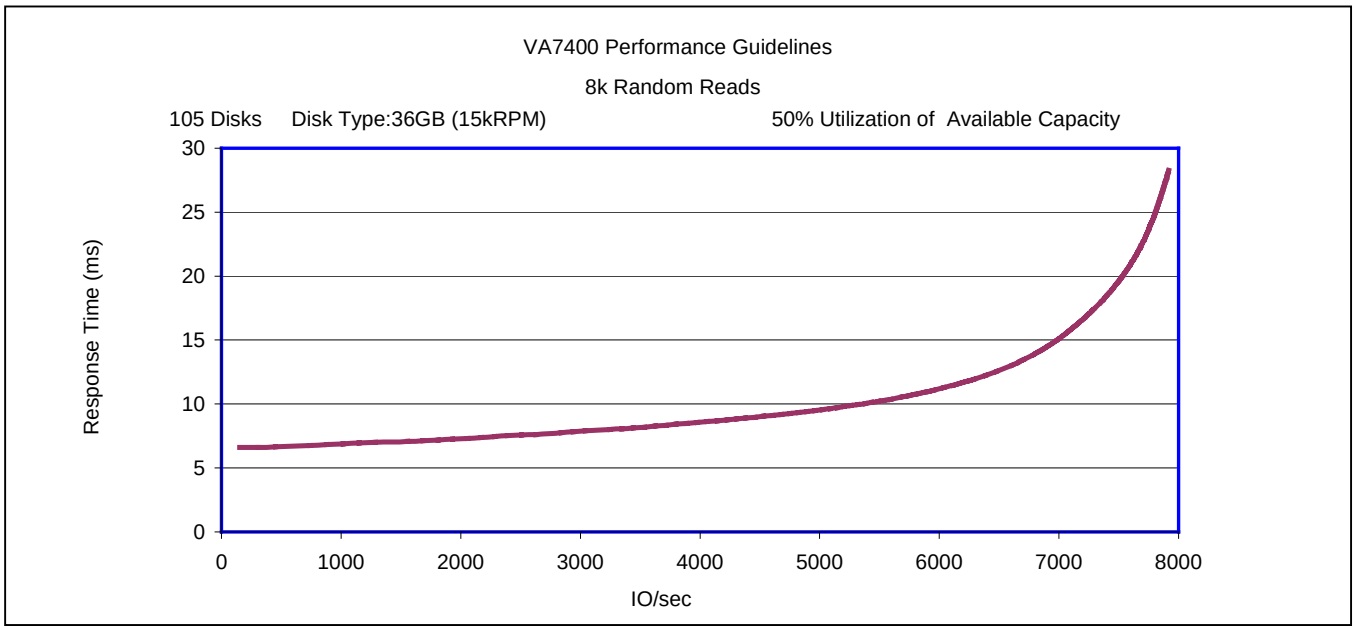
formatting tips for stacked area charts

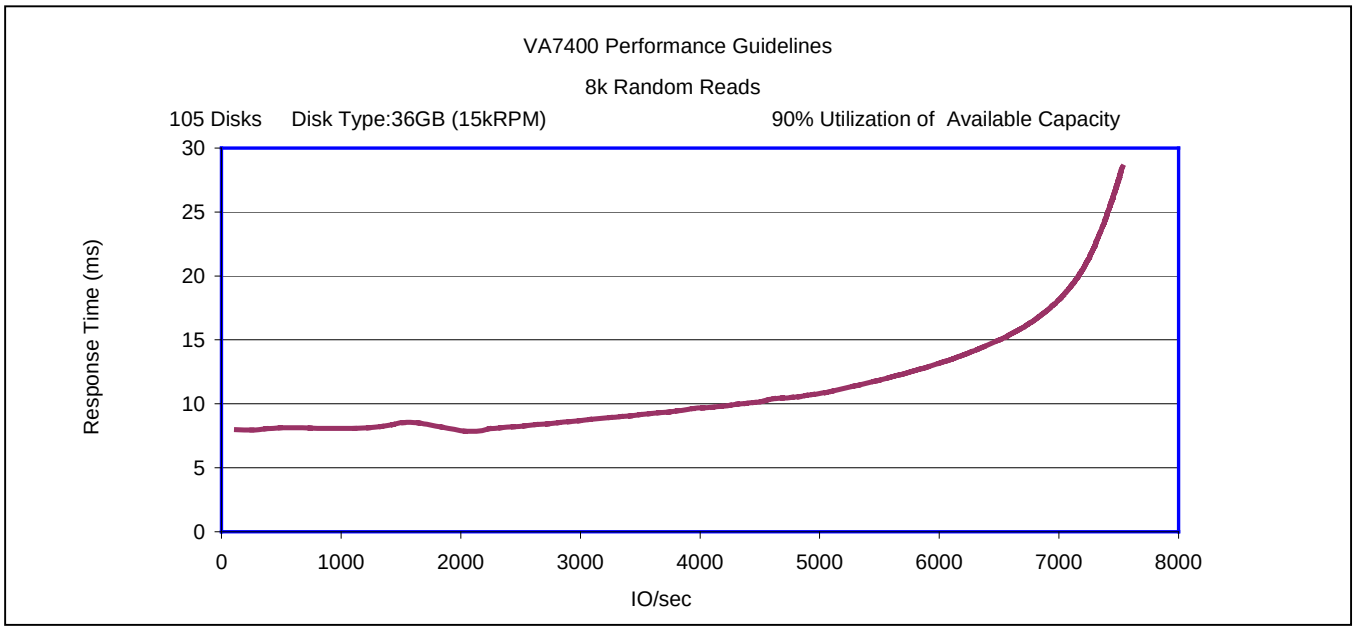
The MS Excel stacked area chart is a useful way to view sets of the performance metric data. In some cases, there is fine detail in the pattern of the data displayed by the chart that can't be seen after the chart has been created with default options. The detail can be revealed by removing the borders from all the data series in the chart. To do this, select each data series then select Format->Selected Data Series. Click "none" as the border option then click "ok". If it is difficult to select the data series with the mouse, use the up and down arrow keys.

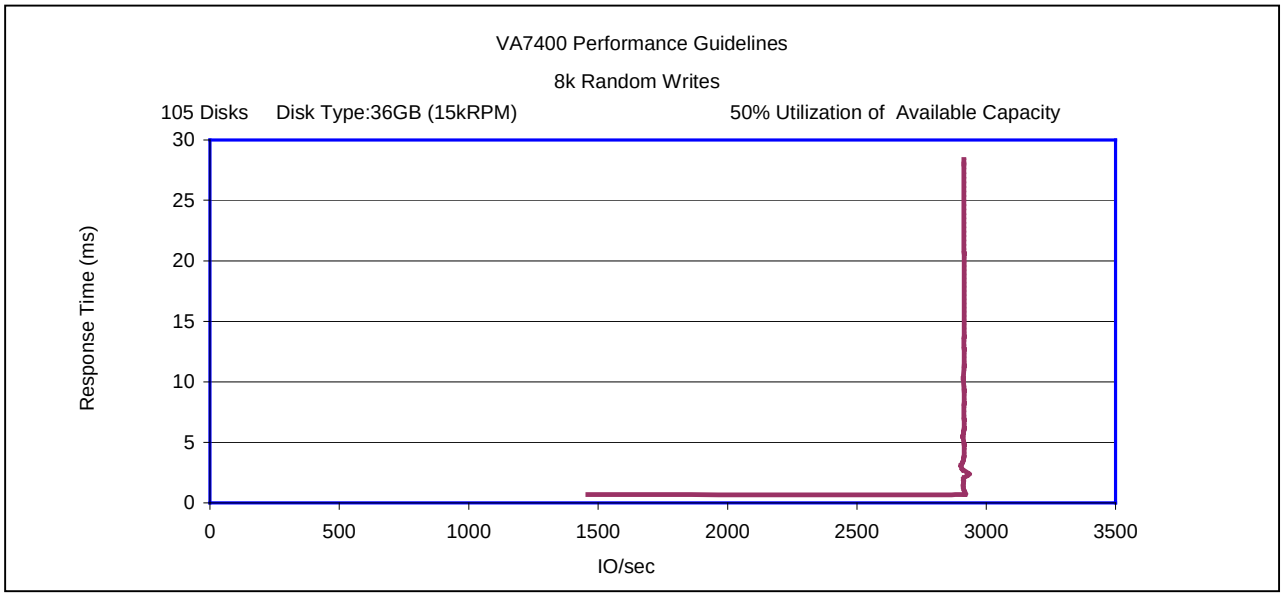


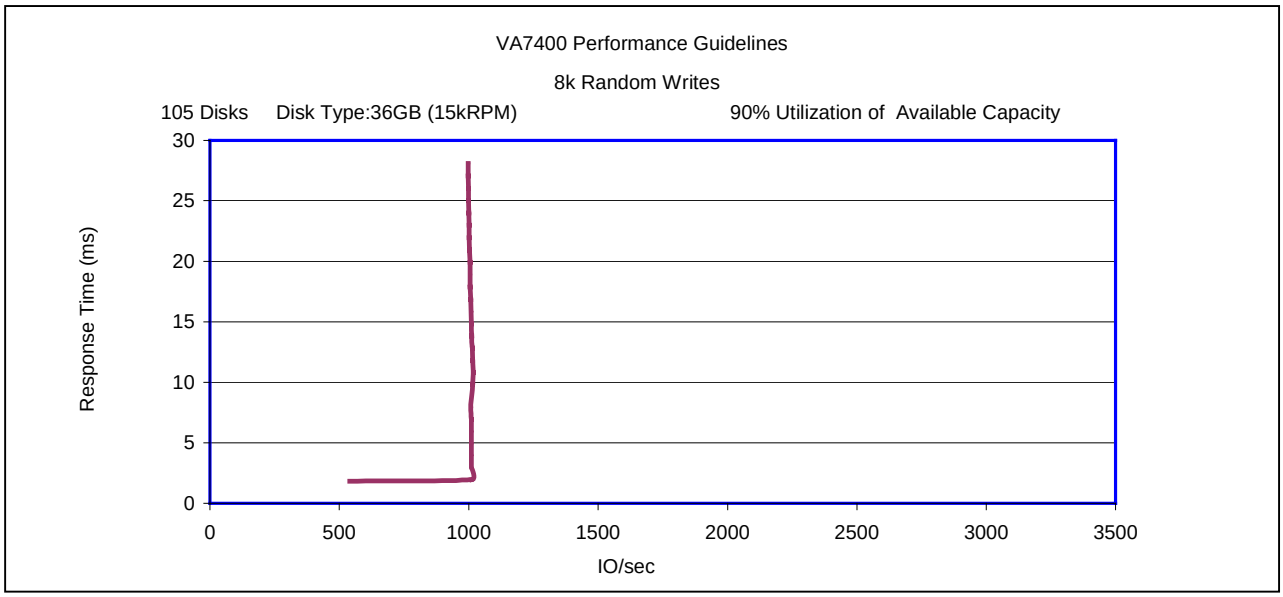
measured saturation curves

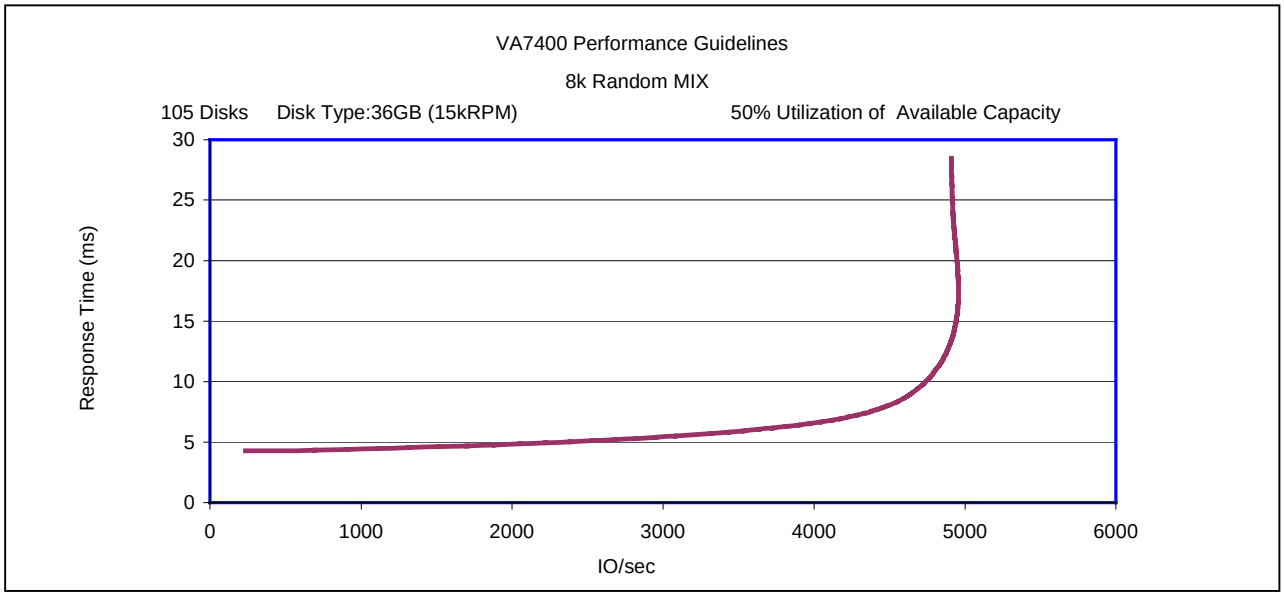
HP has measured performance of the virtual array products with various configurations and workloads. These measurements are useful as a reference when using the performance metrics to analyze virtual array performance. These measurements are the maximum performance that can be expected when ideal workload conditions are present. The measurements were made at both 50% and 90% usable capacity limits. The 50% measurements are representative of the performance capability of RAID 1+0. This is the maximum performance that can be expected when the virtual array is configured for RAID 1+0 mode or for AutoRAID mode and the write working set fits into the RAID 1+0 area. The 90% measurements are representative of RAID 5DP and is the maximum performance that can be expected when the virtual array is configured for AutoRAID mode and the write working set doesn't fit into the RAID 1+0 area. The 50% measurements are the expected norm when the virtual array is in AutoRAID mode because, as explained above, studies have shown that the write working set is typically less than 10% of the total application data set and AutoRAID reserves 10% of the usable capacity for RAID 1+0.

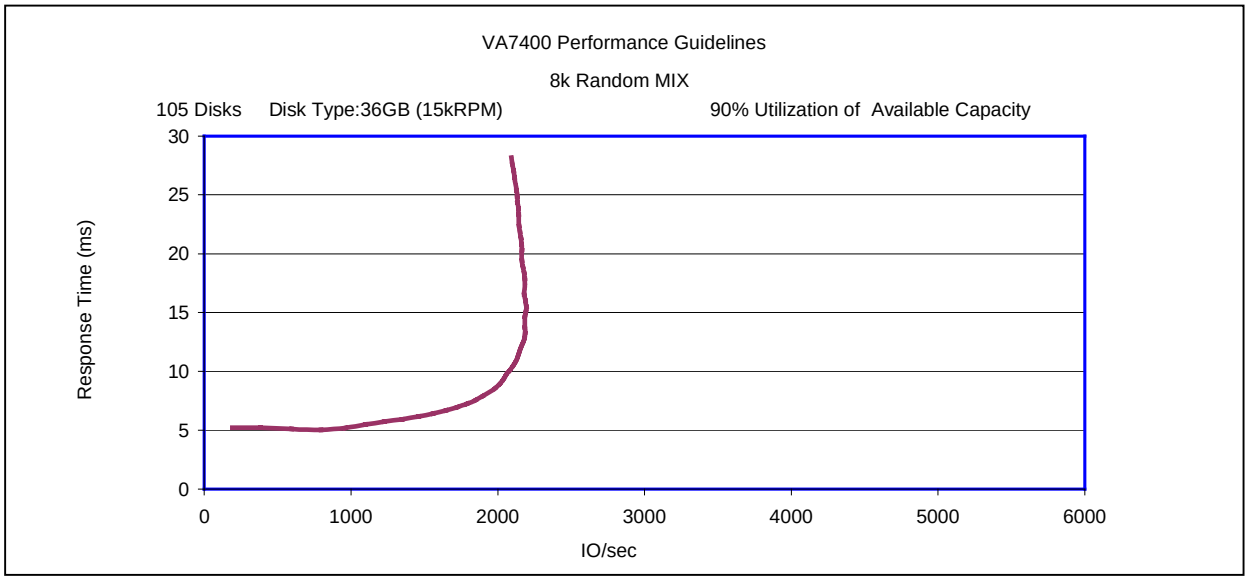


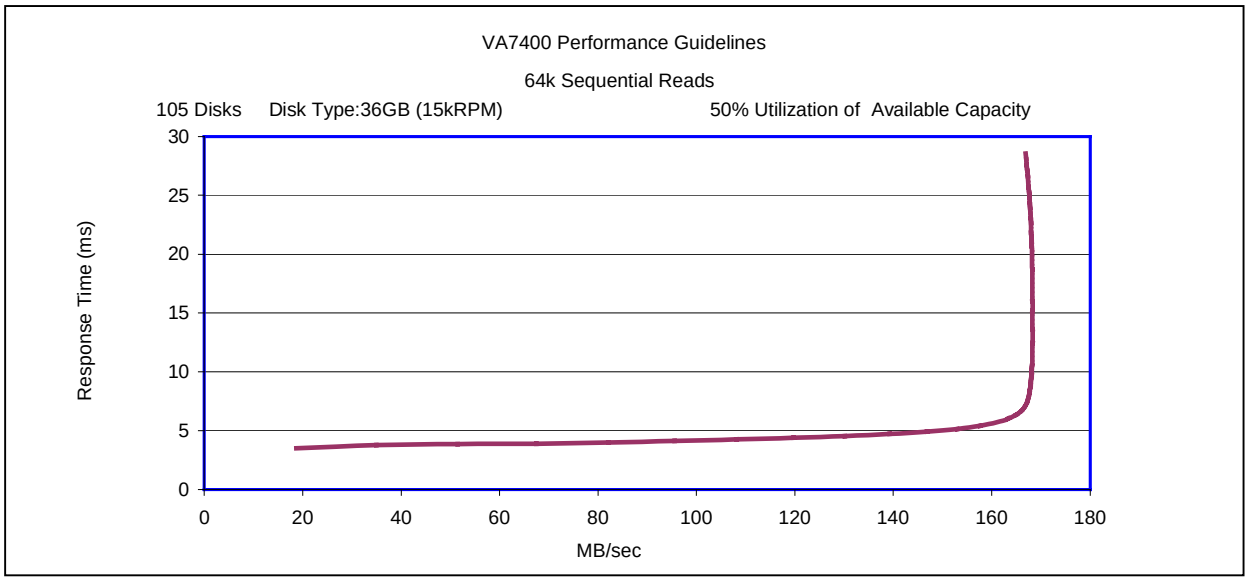


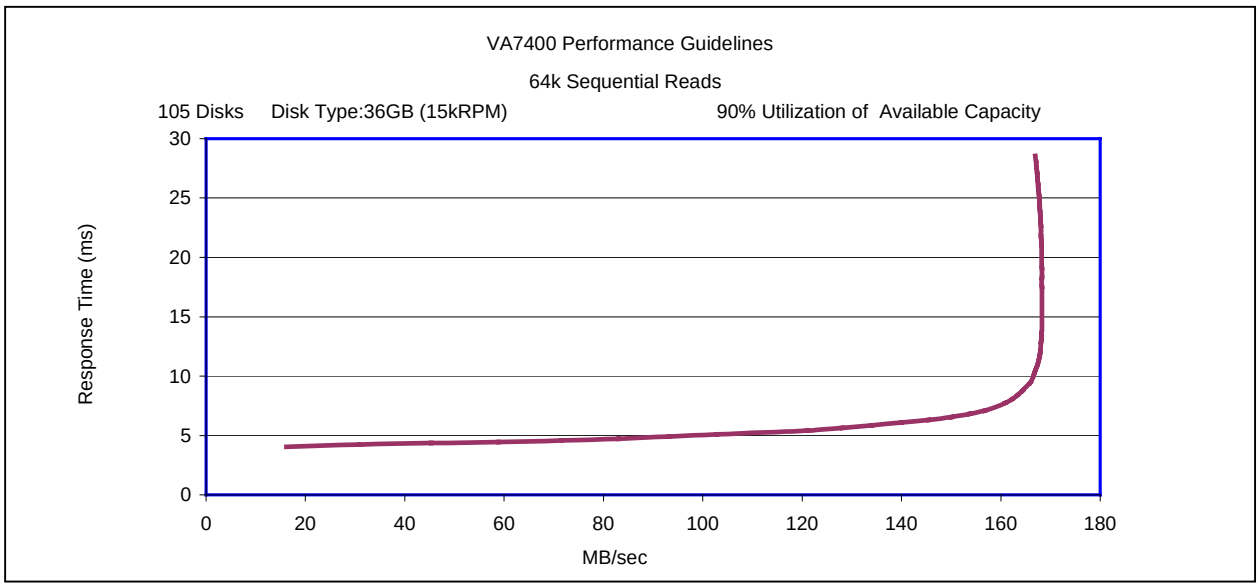


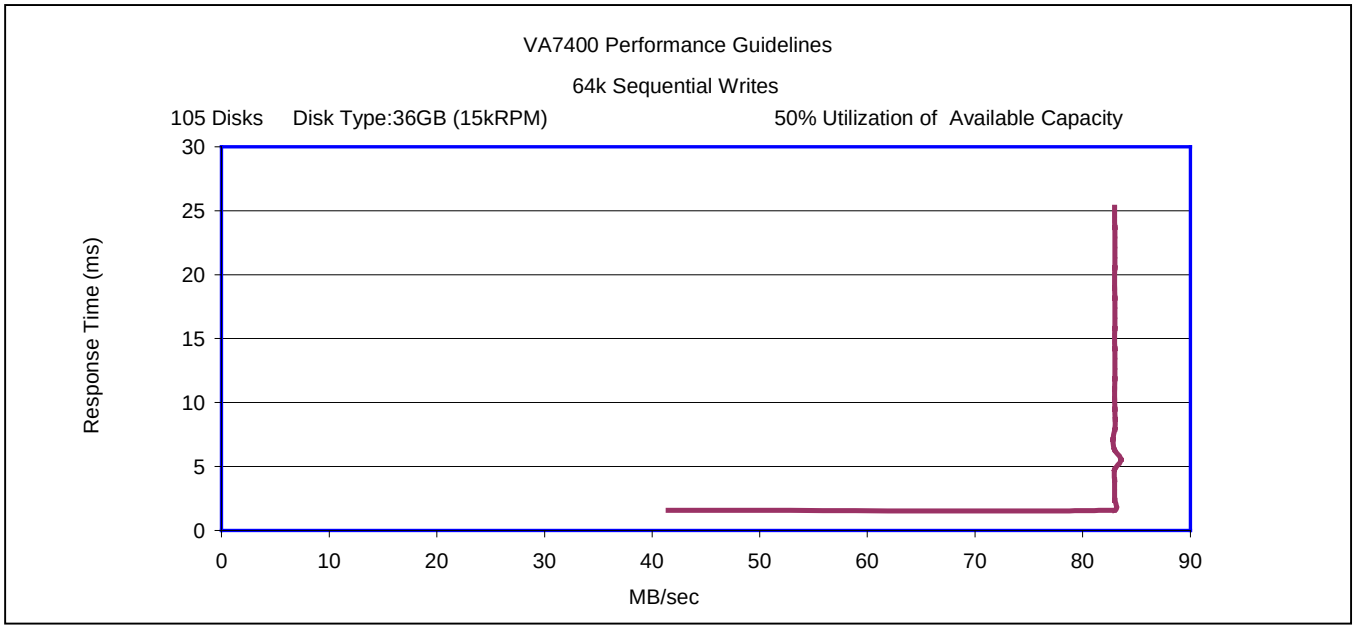


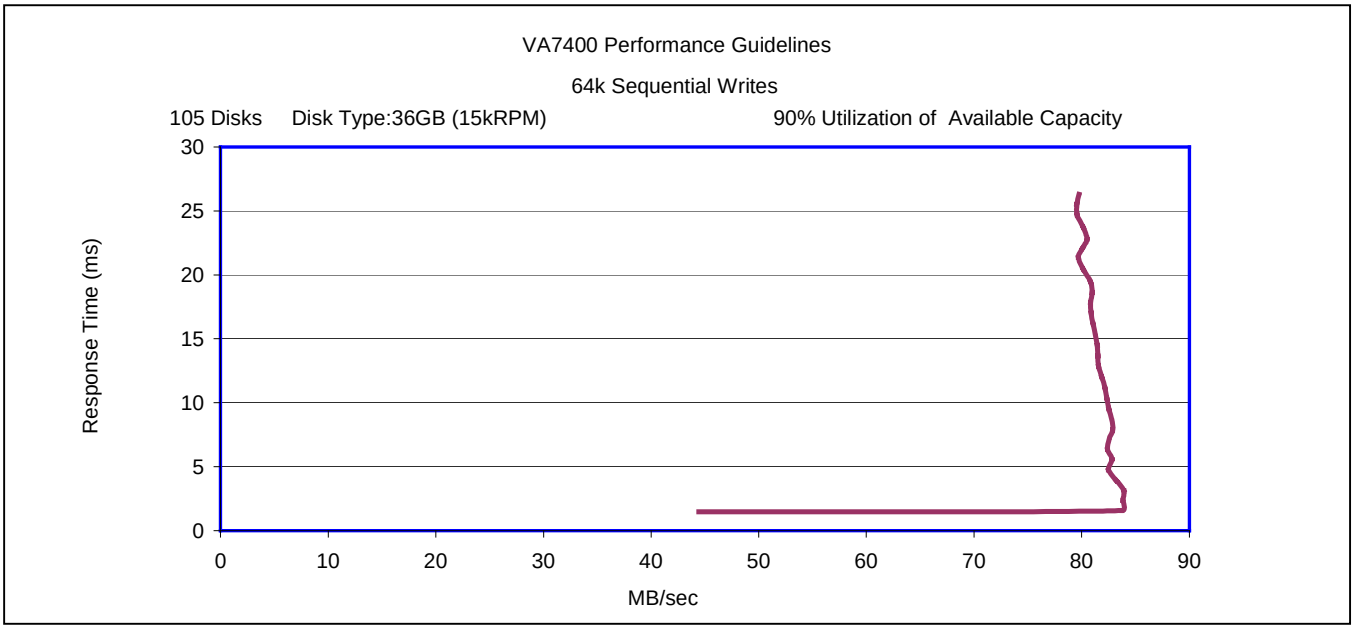












views and interpretations of the performance metrics

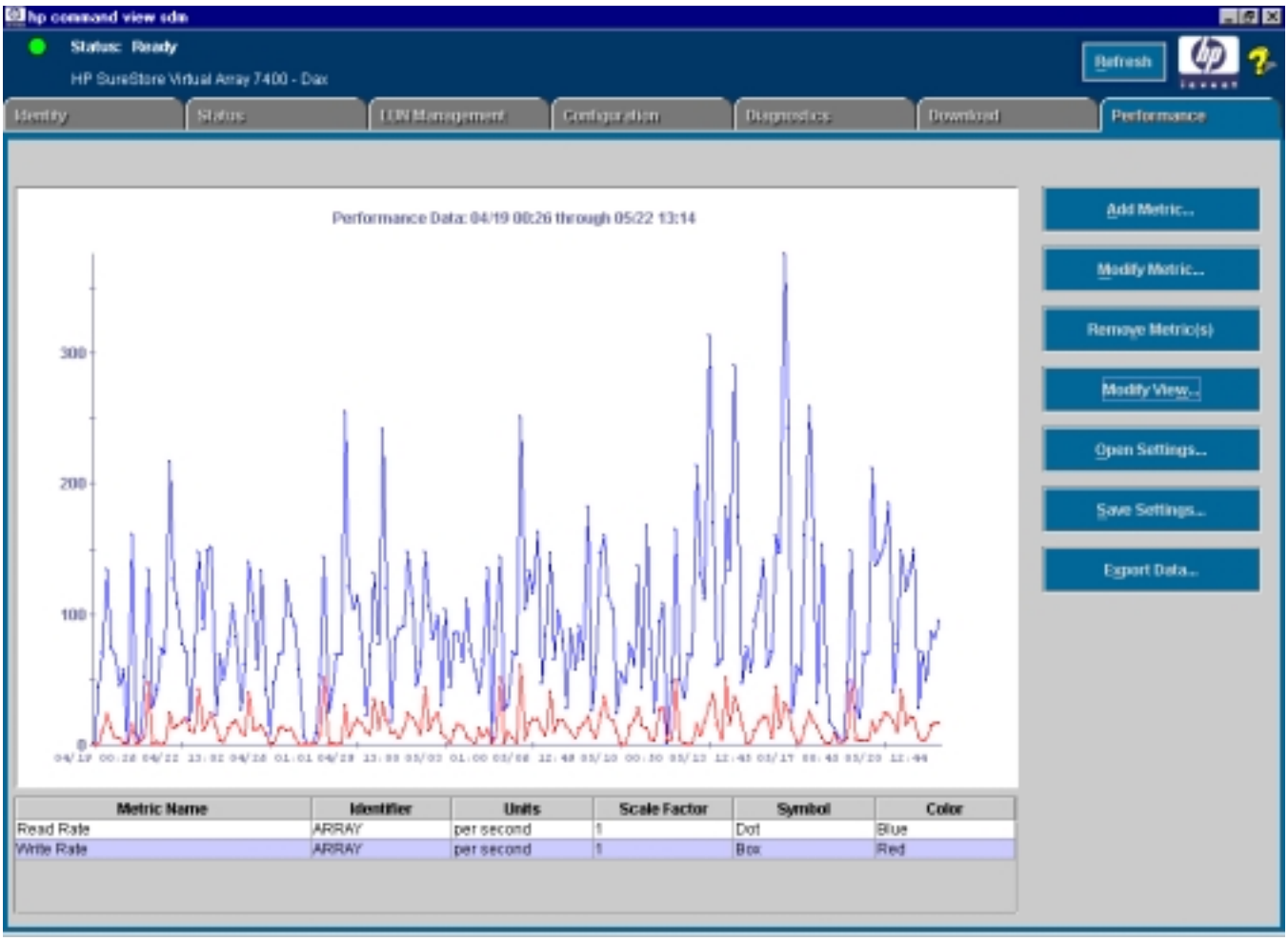
Various views of the performance metric data can be used to analyze different aspects of virtual array performance. The views are presented in Command View SDM GUI format and MS Excel chart format with some interpretation given for each view. Many of the performance metrics are available as summations for the whole array in the ARRAY category and as individual unit metrics in the LUN or OPAQUE categories. The views presented here are of the ARRAY category metrics unless otherwise noted to illustrate the performance analysis techniques, as they would be applied to analyze the performance of a whole array. Similar views could be used at the individual LUN or OPAQUE unit levels for a more detailed analysis.

Data presented in the Command View SDM GUI may not always correspond directly to the same data presented in an MS Excel chart. If a long time range is selected in the GUI, each data point will be an average of some number of consecutive data samples. In MS Excel, there will be a one to one correspondence of the data points in the chart to the data samples reported by armperf. No averaging will occur. Data presented by the GUI can be scaled or un-scaled. If a scaled view is selected in the GUI, only those parameters whose maximum value exceed the absolute value 100 will be scaled to fit in the range zero to 100. Other parameters on the chart will be un-scaled. This can

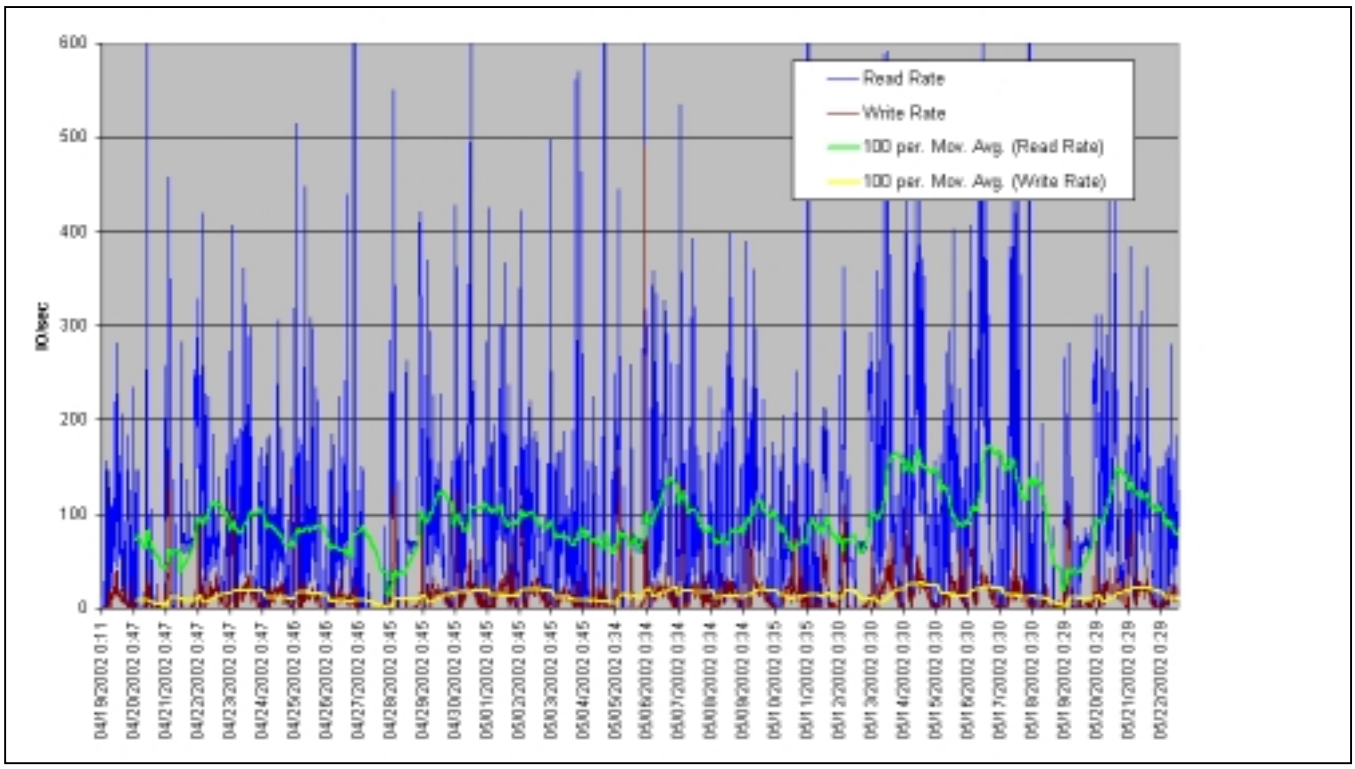
change the appearance of the parameters relative to each other. An un-scaled view is recommended when using the GUI. A scaled view should be used with caution.

total throughput

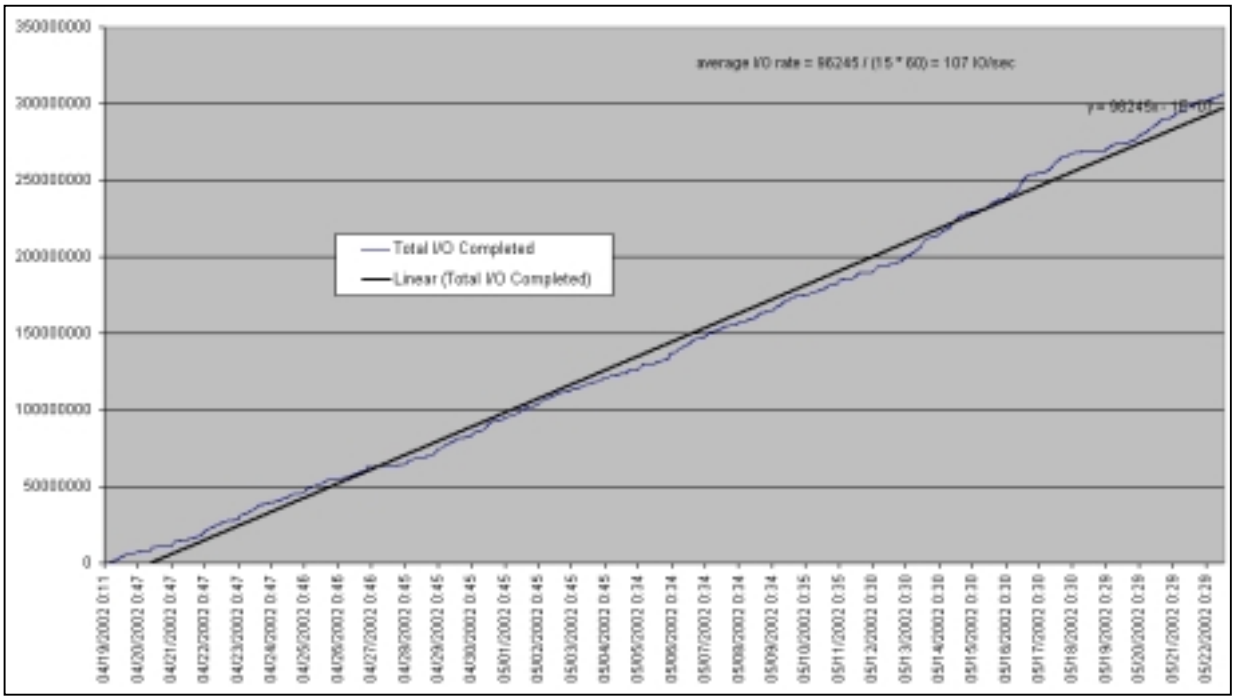
The performance metrics provide measurement of total throughput both in terms of I/O throughput (I/O's completed, IO/sec) and data throughput (blocks transferred, blocks/sec, MB transferred, MB/sec). It is useful to consider these metrics at the start of an analysis to get an overall view of the average performance level being achieved and the variation in performance over time. In many cases the average performance will be much less than the maximum indicated by the measured saturation curves because closed loop workload behavior is causing demand to be throttled.



Throughput data can be difficult to view because the throughput during application use can vary widely over short periods. In MS Excel, the moving average trend line feature of line charts can be used to get a more stable view of the data.

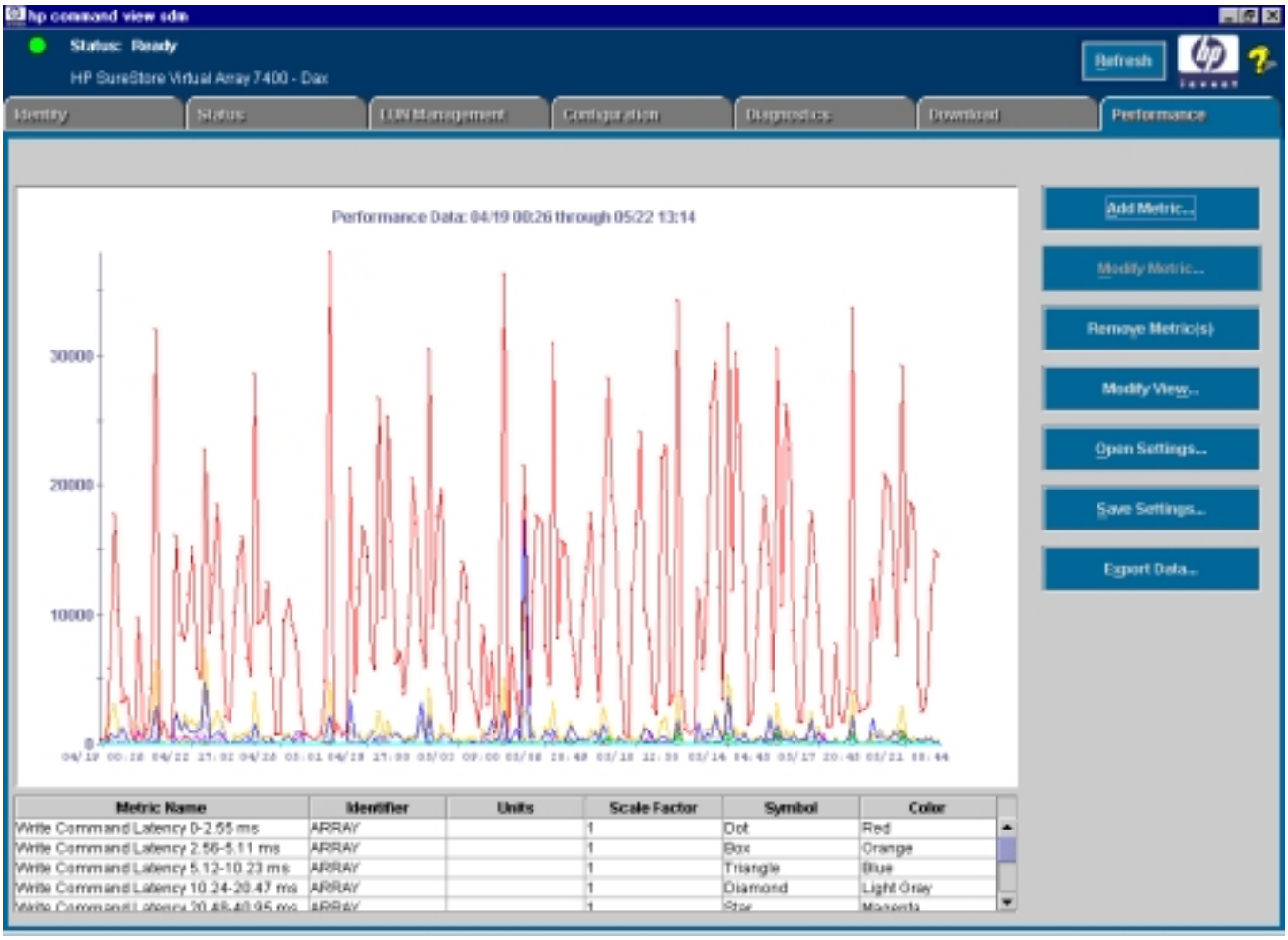


Another useful way to view throughput data which is only available in MS Excel is to view throughput as a running summation of total I/O's completed or total blocks or MB's transferred. The summation is computed using a repeated formula in a new column inserted into the data worksheet. In this type of view, the slope of the line represents the throughput. Variations in slope show how throughput is changing over time. MS Excel has a linear regression trend line feature that can be used to show an average throughput (slope) for the whole period of the chart. A slope/intercept equation can be displayed with the trend line and the slope from the equation can be converted into throughput in conventional units. By default, armperf reports data for all sample periods. A sample period is 15 minutes by default so the time value 15 minutes is used to convert the slope value into a per second throughput number.

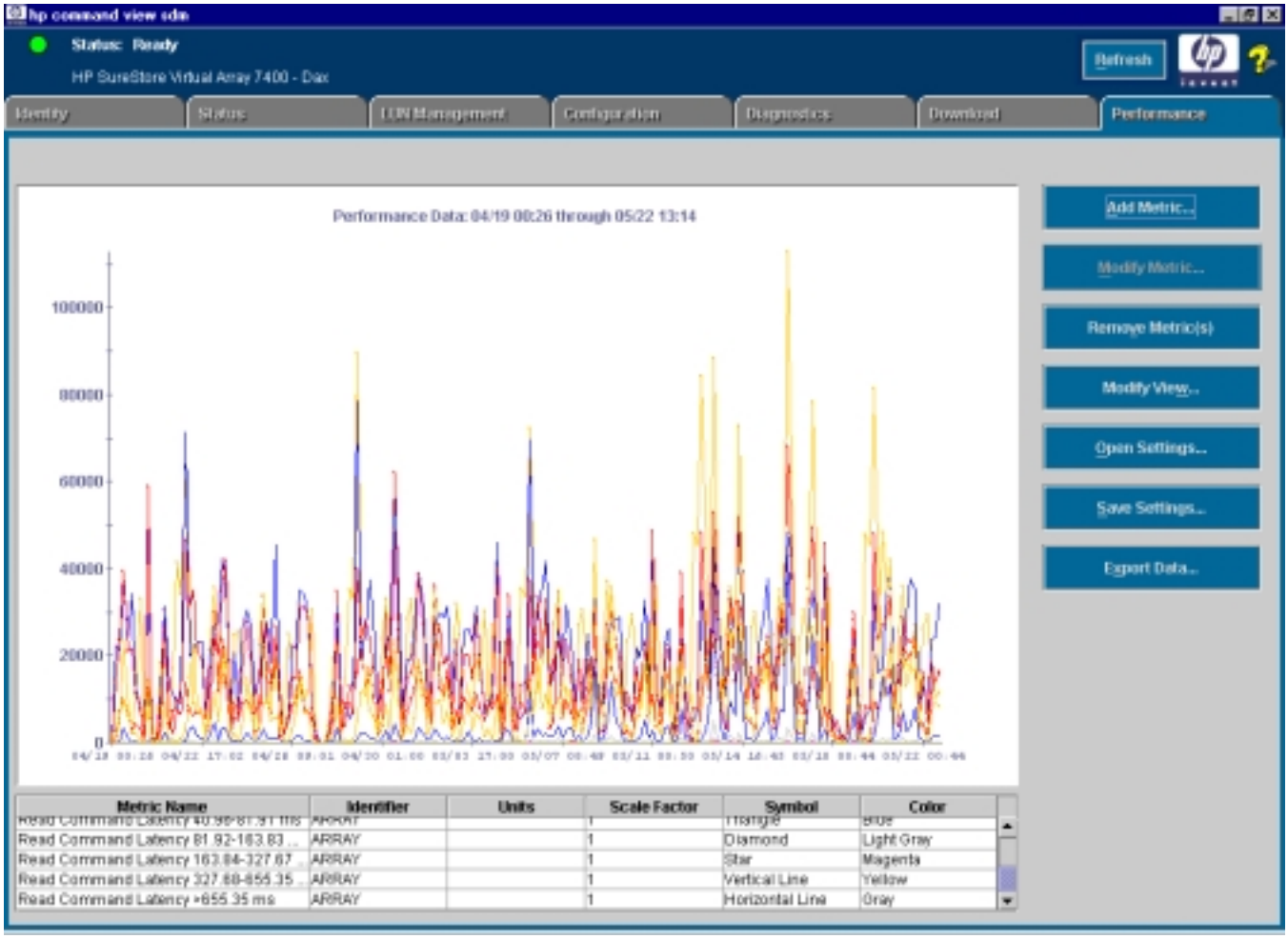


latency histograms

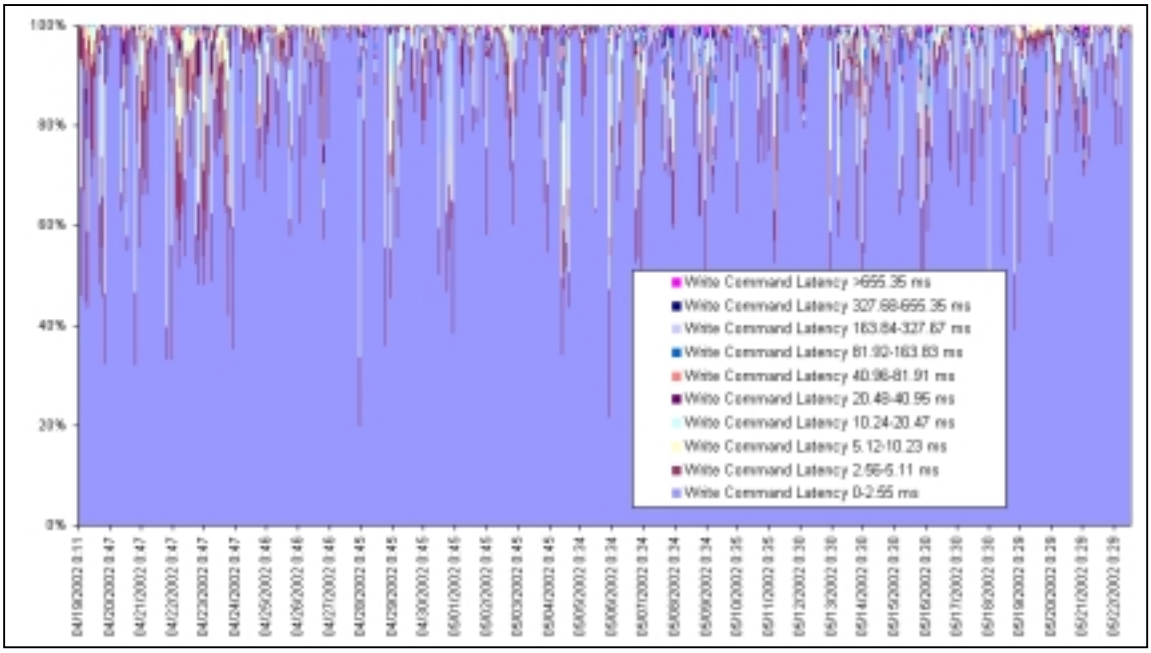
The latency histogram performance metrics are one of the most useful sets of metrics for analyzing VA performance. They count the number of read or write commands whose completion times fall within pre-defined time ranges.



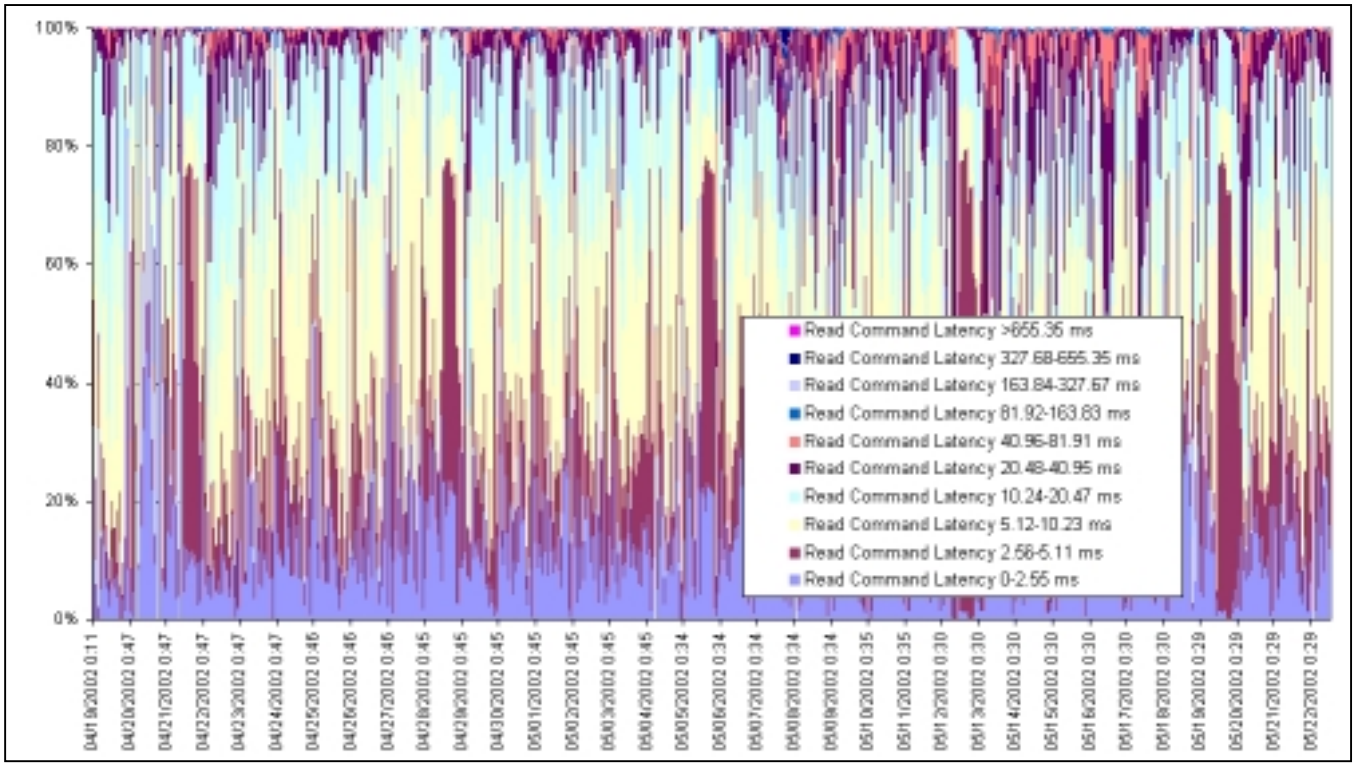
There are separate latency histograms for read commands and write commands.



It can be difficult to view an entire latency histogram in the GUI because there is one line for each time range for a total of 10 possibly overlapping lines. The MS Excel stacked area chart (either absolute or percentage) provides another way to view this data.



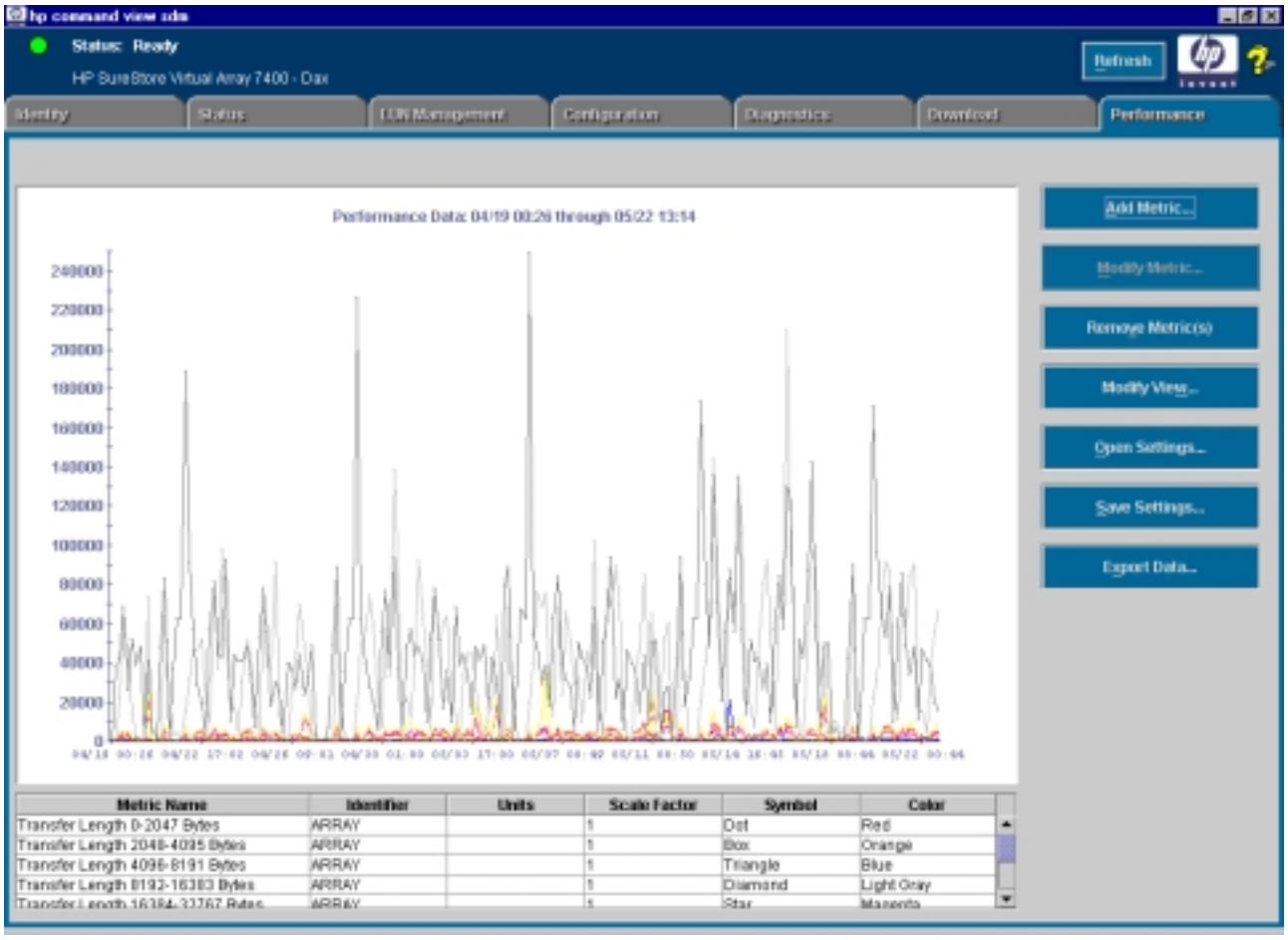
The command latency times indicated by the latency metrics along with the throughput metrics can be compared to the measured saturation curves to determine whether or not an array is operating near or above saturation at particular times. If so, the array itself may be a performance limitation for the application during those times. If there is an indication of high latency (20 milli-seconds or higher) but the throughput is not near saturation levels as indicated in the measured saturation curves, there may be some other aspect of the array or system operation or configuration resulting in the high latencies. In this case, further investigation may reveal the root cause and possible remedies.

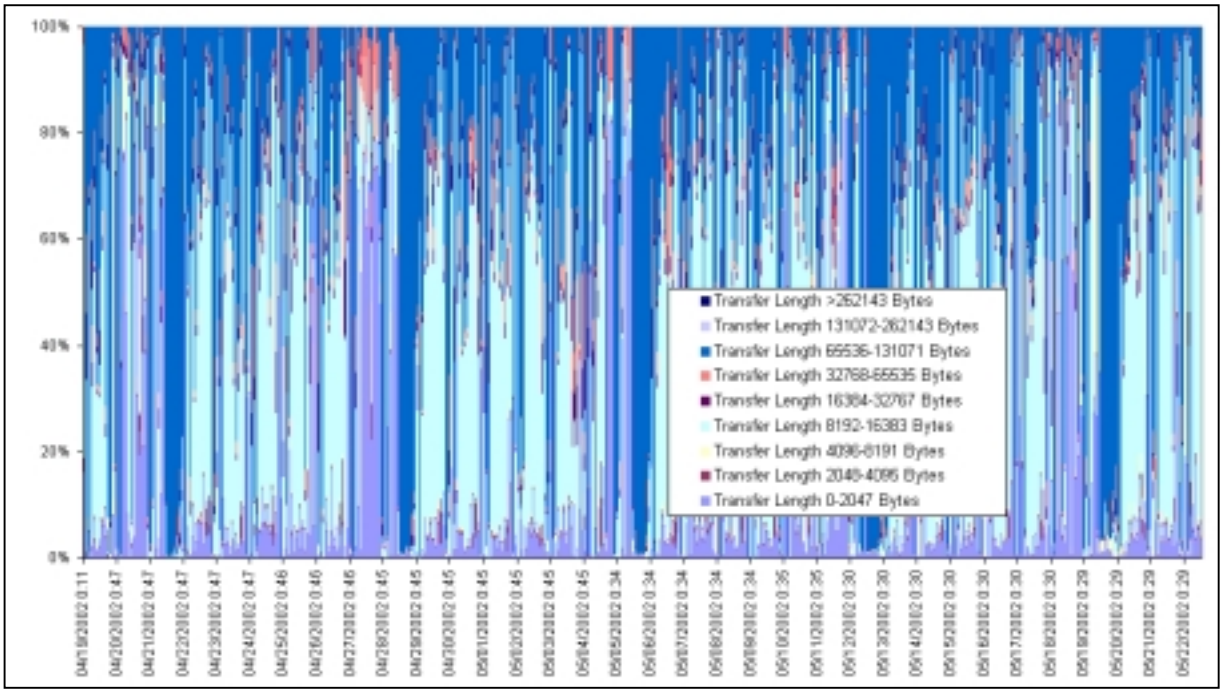


Another thing to look for in the latency histograms are changes in the data pattern. These can be correlated in time to other performance metrics or to known changes in the usage of the system to help gain understanding about why a change in performance may have occurred at a particular time.

transfer length histogram

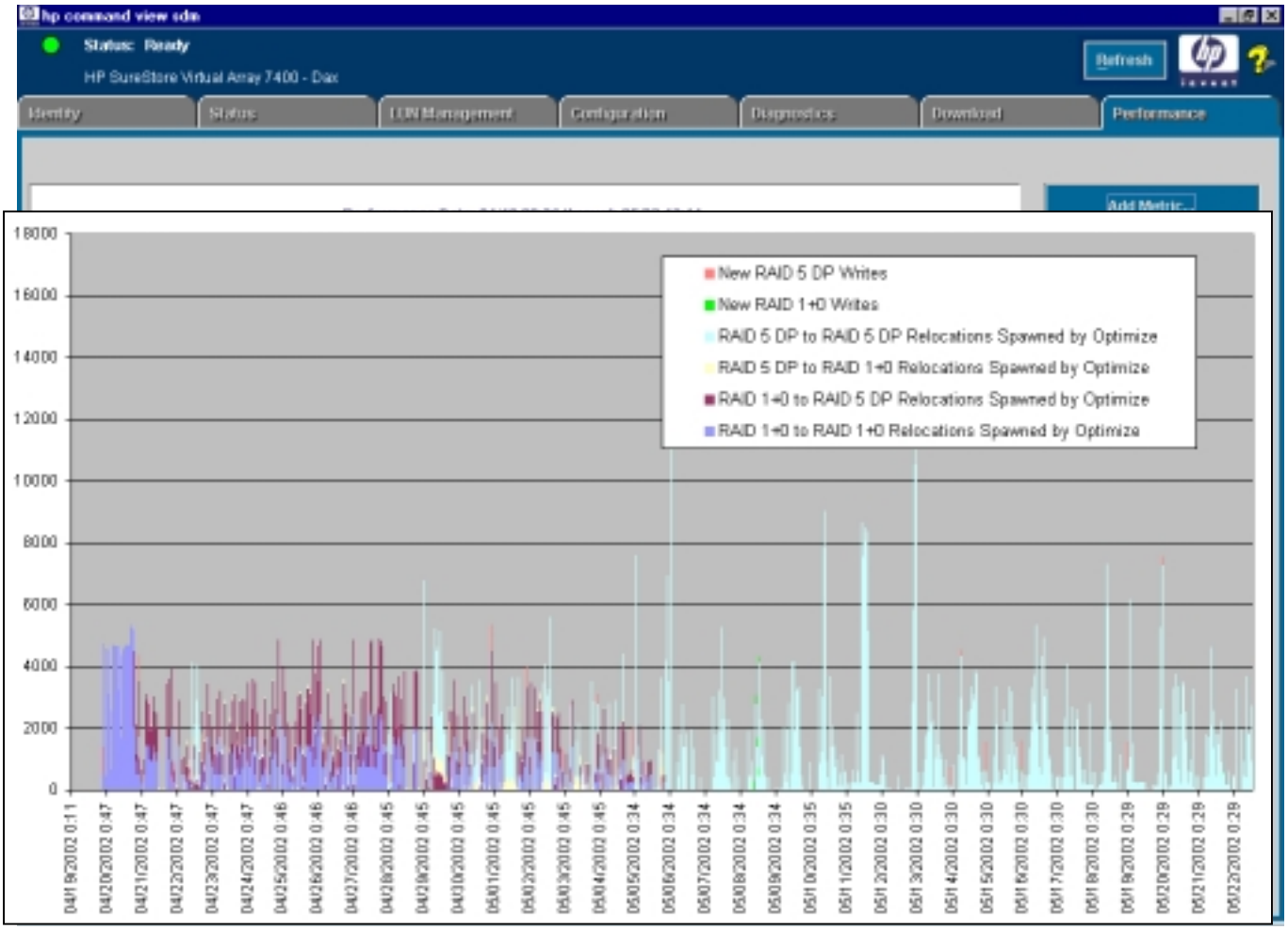
The transfer length histogram performance metrics counts the number of read and write commands whose lengths fall within pre-defined length ranges. There is a single transfer length histogram for both reads and writes. This information is helpful to characterize the array workload.





policy metrics

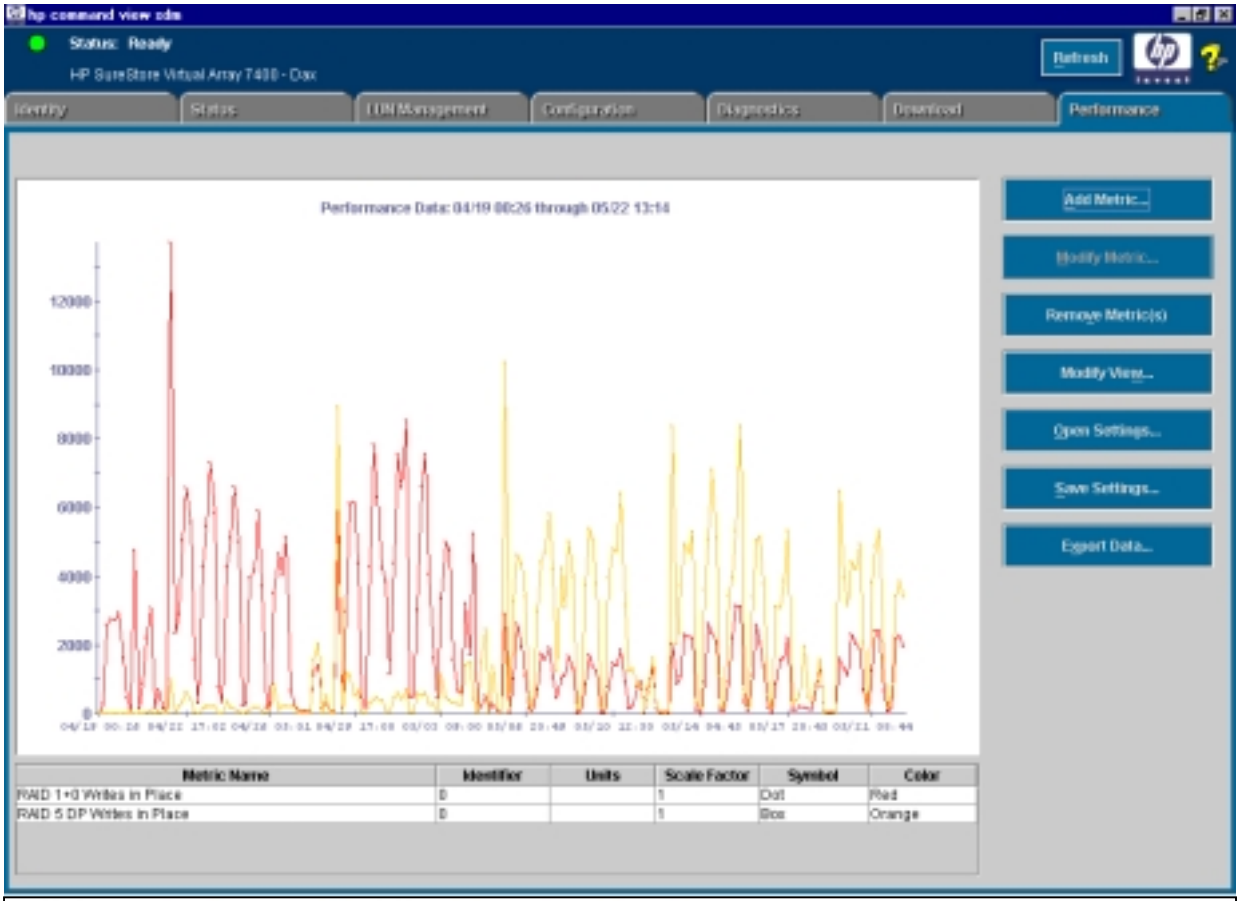
There is a set of performance metrics whose purpose is to monitor AutoRAID policy activity. These metrics are in the OPAQUE category and are not available in the performance GUI in any version of Command View SDM less than version 1.05. There is one set of these metrics for each redundancy group in the array (one set for a va7100 and two sets for a va7400).

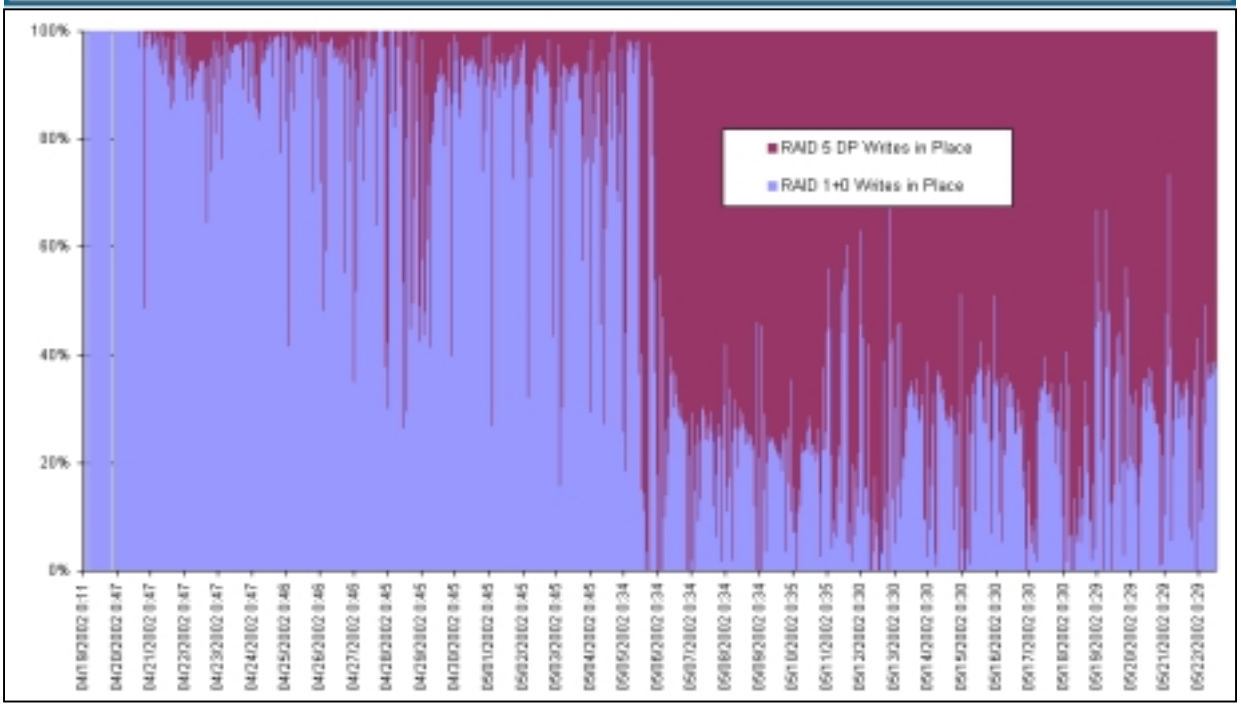


The MS Excel stacked area chart is a useful way to view this data. The new writes metrics indicate the rate at which new data is being added to the RAID 1+0 and RAID 5DP storage areas. The relocations metrics indicate the rate at which data is being migrated within or between the RAID 1+0 and RAID 5DP storage areas. The goal of these migrations is to maintain the write working set in the RAID 1+0 storage area as data is being added to the system by new writes. The new writes and migrations metrics count the number of write operations or migration operations.

writes in place

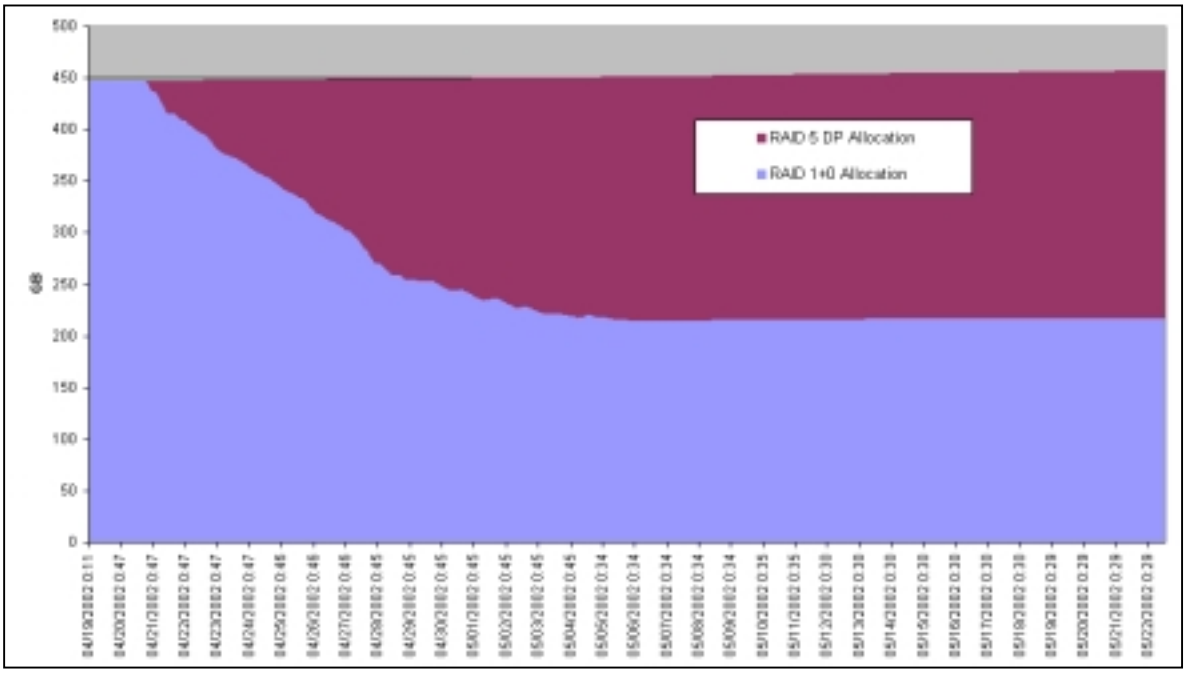
The writes in place metrics are in the OPAQUE category. There is a set for each redundancy group and they are not available in the GUI for any version of Command View SDM less than version 1.05. The relative comparison of RAID 1+0 writes in place to RAID 5DP writes in place is an indication of the extent to which AutoRAID is providing a performance benefit by keeping the write working set in the RAID 1+0 storage area. Write performance will be enhanced as more writes in place are satisfied from RAID 1+0 as opposed to RAID 5DP.





RAID physical allocations

The RAID 1+0 and RAID 5DP allocation metrics are in the ARRAY category and are a summation for all redundancy groups. These metrics indicate the amount of data physically stored in the respective RAID areas. This is to be contrasted with the allocated LUN capacity of the array. Allocated LUN capacity will not consume physical space until it is written. The more data stored in RAID 1+0, the better the overall performance potential for writes as there will be a higher probability that writes in place will occur in RAID 1+0 rather than RAID 5DP.



summary

The performance metric data views presented in this paper are the most common views that would be used to begin a performance analysis. They are not the only views possible but serve as examples of the analysis capabilities that are available.

for more information

For additional information on HP StorageWorks Virtual Arrays and other HP storage products and solutions, please call your local HP sales representative or visit the HP storage Web site at http://www.hp.com/go/storage.

06/02