va 7100
performance
guide

understanding the
performance of the
va 7100 array

for information about the va 7100
and periodic updates to this guide
see the HP SureStore website at
http://www.hp.com/go/storage/

Hewlett-Packard Company

Hewlett-Packard Product Information

va 7100 performance guide – understanding the performance of the va 7100 array

Published: April 2001

Revision level 1.0

For the latest updates to this guide see
http://www.hp.com/go/storage/

Table of Contents

Introduction

What is the purpose of this Technical Guide?

The purpose of this Technical Guide is to present RAID 1+0 performance data for the VA 7100. The focus is on the performance of the array under normal operating conditions, such as when the array is fully functional, with no failed components. The goal is to demonstrate what performance can be expected from the VA 7100 when a user's working set fits in RAID 1+0.

Data is being presented in this Technical Guide for some simple workloads that either avoid the array's cache entirely or that are satisfied entirely from cache. In the former case we refer to the performance of the array as *back-end performance*, while in the latter case the performance of the array is referred to as *peak performance*.

The performance of any storage subsystem depends on a number of factors including the workload presented to the subsystem; the type and configuration of host computers generating the workload; whether file systems are used; and whether other software layers, such as Logical Volume Manager (LVM), are present.

Since the purpose of this document is to demonstrate the performance of the array, we have attempted to minimize host and host software effects by using a powerful server to generate workloads to raw devices instead of writing to volumes managed by a volume manager such as LVM, or to a file system managed by an operating system.

Note: All data presented below was obtained with the VA 7100 configured in *High Performance* mode.

Experimental Setup

How were the measurements made and the array tested?

All measurements were made using an 8-way, N-Class server, running HP-UX 11.00 equipped with 4 GB of main memory and A5158A fibre-channel host-bus adaptors.

The array tested was a VA 7100 populated with 15, 18.2 GB disk drives. The drives had a rotational frequency of 15 k rpm. The array was equipped with 1 GB of main memory per controller. The firmware was not specifically tuned for these tests.

In all tests one fibre-channel connection to the array was used. This connection passed from one fibre-channel host-bus adaptor to the primary controller of the array.

Workload Considerations

Having minimized the impact of the host, the most important factors that determine the performance of the VA 7100 are the properties of the workload used. The most important workload factors include:

● The intensity of the workload.

● Whether the workload is random or sequential.

● Access type (whether the requests are for reads, writes, or a mixture of reads and writes).

● Requests sizes (the length of the data requested).

● The locality of the set of requests comprising the workload as measured by the working set size, which is the total capacity of the data being requested by the workload.

How was the workload intensity varied?

To vary the workload intensity, we simply varied the number of host processes that were allowed to issue I/O requests to the array. This was done because the tool we used generates *closed-loop* workloads. Each process launches an I/O request to the array, waits for that request to be satisfied, does a computation (to measure performance), then launches the next request. By increasing the number of processes, the demand on the array is increased. In order to minimize the host *think time*, such as the time it takes for the host to do the measurement computation and release the next request to the array, we used a very fast host (*see below*).

What performance measurements were made?

We made performance measurements on both random and sequential workloads. For each type of workload we tested the array separately with 100% reads and 100% writes. In addition, we tested with a mixture of reads and writes for the random workload case. The read-write ratio used was 3:2. Such a workload gives an indication of how the array will perform under **On-line Transaction Processing Systems (OLTP)** applications. The sequential read workload results provides a measure of how well the array will perform during backups and in **Decision Support System (DSS)** environments. The sequential write workload results provide a measure of how well the array will perform under restore-from-tape and other sequential operations. For back-end performance measurements, we used 2 KB and 8 KB transfers for the random workloads and 64 KB transfers for the sequential workloads. For the peak

performance measurements we used transfer sizes of 1 KB and 128 KB for the random and sequential cases, respectively.

Benchmarking Procedures

Running representative performance benchmarks on the VA 7100 involves a few different considerations from those of traditional RAID arrays.

Notice the difference between the typical storage array and creating a LUN on the VA7100!

When a LUN or logical drive is created on the VA 7100, the array records the maximum size of the LUN but does not allocate any real space. Therefore, the array can be used immediately after logical drive creation. However, in typical storage arrays, once a LUN is created that LUN is formatted. In other words, a format pattern is written to each disk segment of the LUN. During formatting, either the LUN cannot be accessed by the user or, in cases where it can, the array's performance is greatly reduced, compared to the performance after formatting is finished. This formatting step can take several hours to finish.

In contrast, space on the VA 7100 is allocated *on-the-fly* simply by writing to created LUNs. When a write request is destined to an unallocated disk segment, a controller performs a space-allocation operation. If the write is destined to a disk segment that has already been allocated this step is omitted. Thus, the array's performance is lowest when the array is first installed and being loaded with new data. The performance will then improve over time as it is used.

> **Note:** The initial performance penalty for *new writes* is not nearly as extreme as that paid on typical arrays when they are being used during a LUN format phase. Furthermore, the penalty paid on typical arrays is for both read and write requests, whereas the penalty is only paid for writes on the VA 7100.

VA 7100 allocated space must be distinguished from file system entities such as files. Imagine that a file system on a VA 7100 LUN is completely full. If that file system is completely wiped out, say, by executing a Unix *newfs* command, the new file system will be completely empty. However, all the disk segments of the VA 7100 LUN will remain allocated. Therefore, writing new files to the newly created file system will cause *non-allocating writes* to occur. These do **not** suffer the allocation penalty mentioned above. To cause the disk segments of the LUN to become unallocated, that LUN must first be deleted then re-created.

Another consequence of *on-the-fly* allocation is that if a random write workload is imposed on a newly created LUN, the data is stored in the array sequentially. Thus, if such an operation is followed by a sequential operation on the same LUN, a random workload is presented to the disks.

For the reasons just discussed, before any of the measurements reported here were made, VA 7100 LUNs were created and filled sequentially. This ensured that random workloads would indeed be random as viewed by the disk drives.

Prefilling LUNs for benchmarking.

There are two ways that LUNs can be pre-filled for benchmarking. The LUNs can be filled serially (the pre-fill can be performed one LUN at a time), or the LUNs can be filled concurrently (the LUNs are filled in parallel). The serial method should be used if testing with subsets of LUNs is desired. The concurrent method works just fine if the intent is to impose workloads on all the LUNs concurrently.

Note: The serial method of prefilling was used to obtain the results of this paper.

Measurement Results

In this section we present the measurement results. We will begin with a discussion of the VA 7100's performance under random workloads. Followed by a discussion on the array's performance under sequential workloads.

Random Workloads

As stated in the Introduction, we measured the performance of the VA 7100 for 2 KB and 8 KB transfers using 100% read and 100% write workloads and a workload having a 3:2 read-write ratio, such as an OLTP workload. The data was obtained by varying the number of I/O processes from one (1) to some maximum value in powers of two (2) in order to control concurrency or the intensity of the workload.

The measurement results for 2 KB transfers are shown in graphical form in Figure 1 on page 5. The plot depicts mean response time, measured in milliseconds (ms), versus the throughput, measured in the number of I/O operations that complete per second (IOPs). Such a curve is commonly referred to as a *saturation curve*. Notice that the saturation throughput for reads is about 3200 IOPs, and the saturation throughput for the OLTP workload is about 2500 IOPs.

Notice the pure-write workload behavior!

The behavior of the curve for the *pure-write* workload shows an interesting feature. The data-points corresponding to the three (3) lowest response times were obtained by using 1, 2, and 4 I/O processes, respectively. The throughput actually drops in moving from the 4 process case to the 8 process case. This transition occurs as a result of cache saturation. The array is able to destage data in the background when 4 or less processes are used. With 8 processes, the array's cache behaves as a FIFO; every incoming write must wait for the data in

a cache page to destage to disk. The vertical line, therefore, represents the saturation throughput of the disk drives. It is important to note that the peak throughput of nearly 1800 IOPs achieved for writes shows the rate at which the array can write data to disk from the cache. We are not showing the much higher rate that the host can write data to the cache.

Figure 2 on page 6 is similar to Figure 1 on page 5, except that it shows data for 8 KB transfers.

> **Note:**   Use of the larger transfer size has negligible effect on the saturation curves.

FIGURE 1. Random I/O Performance for 2 KB Transfers
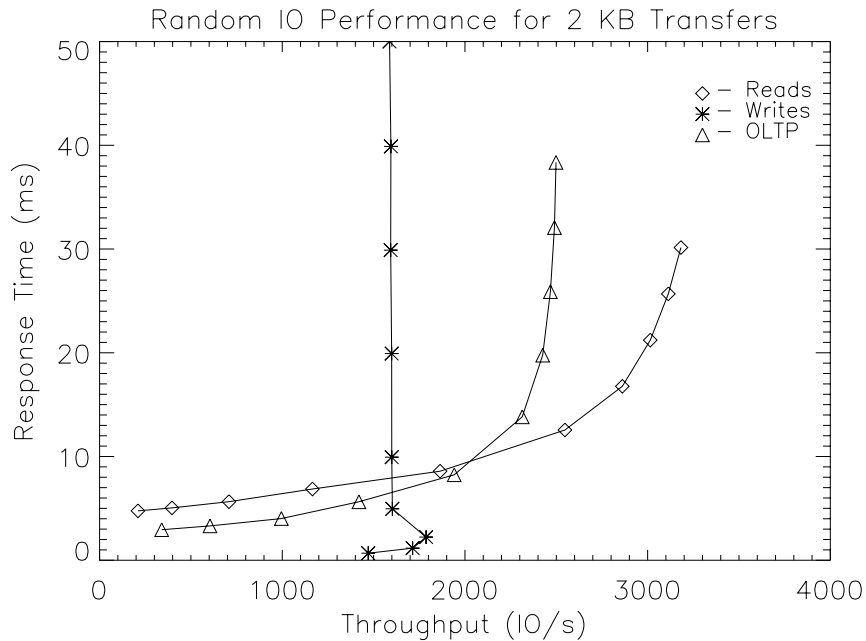
Response time versus throughputs for 2 KB transfers. The three curves are for all reads shown on the graph as diamonds (◊); all writes shown on the graph with asterisks (*); and the OLTP workload is shown on the graph as triangles (Δ).

FIGURE 2. Random I/O Performance for 8 KB Transfers
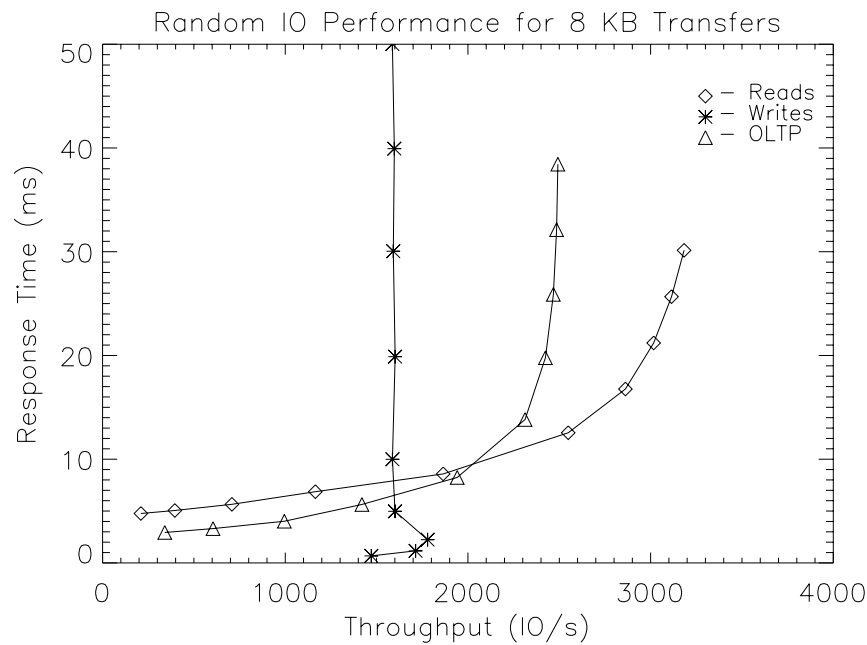
The above figure shows the response time versus throughputs for 8 KB transfers. The three data sets graphed are for reads, writes, and the OLTP workload.

TABLE 1. Summary of random Workload Performance at ≤ 25 ms (IOPs)

Transfer Size (KB)	Reads	Writes	OLTP
2	3,110	1,790	2,270
8	3,110	1,790	2,270

Sequential Workloads

In this section, performance data for the VA 7100 is presented when it is subjected to sequential workloads. The data summarized in Figure 3 on page 7 shows a plot of response time versus throughput for sequential reads (depicted with diamonds) and sequential writes (depicted with triangles).

The data was obtained, for both reads and writes, by first following the prefill procedure discussed in the previous section on "Benchmarking Procedures" on page 3. After the prefill was finished, we obtained the data by measuring the sequential performance with a single sequential stream, two sequential streams, four streams, and so forth.

Data measuring, using a single stream.

For the *single stream* case, the data-point for reads corresponds to the lowest throughput data-point, while the data-point for writes corresponds to the highest throughput. The reason that the highest throughput for writes is obtained with one stream is that the write cache of the array is able to efficiently destage data to disk in the background. This is similar to the results we obtained for the random workloads.

Data measuring, using an increased number of streams.

As the number of streams is increased, the read throughputs increase smoothly as do the mean response times. At a throughput of about 85 megabytes per second (MB/s), the throughput response time curve bends upward steeply. At about 90MB/s the mean response time is approximately 17 ms.

The curve for the writes bends backwards in throughput from its peak of about 42MB/s to about 36MB/s as the write cache quickly saturates. In saturation, before the cache can accept data it must first make room by destaging data. The situation is the same as it was for the random writes.

FIGURE 3. Sequential I/O Performance for 64 KB Transfers
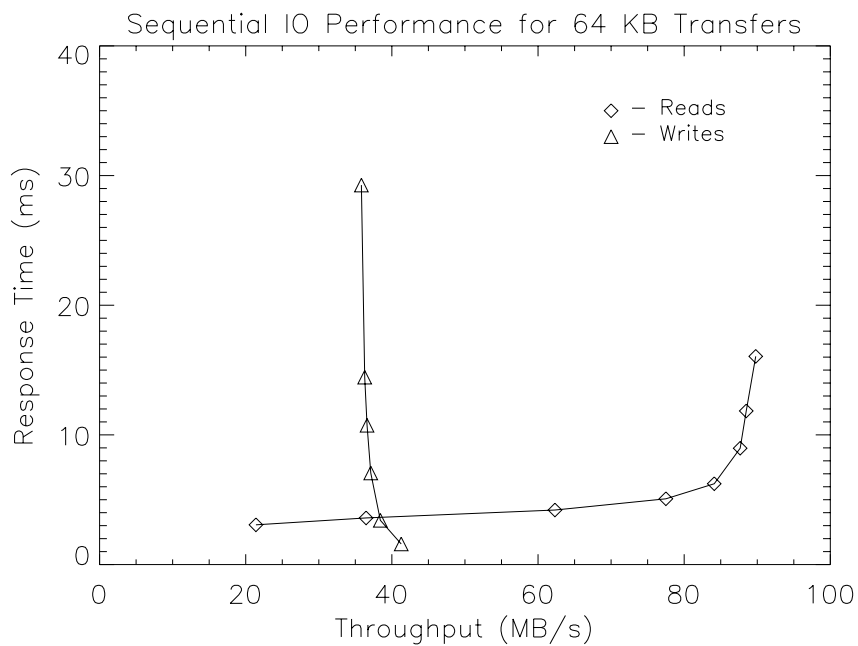
The above figure shows the response time versus throughputs for 64 KB sequential transfers. The two data sets graphed are for reads and writes. Data plotted on the graph as diamonds (◊) correspond to read performance, while data plotted on the graph as triangles (Δ) correspond to write performance workload.

Peak Performance

We conclude this section with a short discussion of peak throughputs for the VA 7100. These peak numbers correspond to workloads whose I/O requests are satisfied from cache 100% of the time. We expect that workloads whose I/O requests can be satisfied in this fashion will be rare, if they exist at all. Because of this, we used transfer sizes that yielded the highest throughputs. Although we believe such numbers are unrealistic, we show them because it seems to be a common practice in the industry to do so. The peak throughputs are summarized below in Table 2.

TABLE 2.   Summary of Peak Throughputs

Workload	Transfer Size (KB)	Throughput
Sequential Read	128	93.1 MB/s
Sequential Write	128	92.7 MB/s
Random Read	1	12,400 IOPs