M.A.X.
The Maximum Achievable Xfer Rate of a Silicon Graphics CHALLENGE™/Onyx™ Computer System

Thomas M. Ruwart
University of Minnesota
Army High Performance Computing Research Center
Graphics and Visualization Laboratory
tmr@ahpcrc.umn.edu

April 15, 1994

Abstract

Projects at the Army High Performance Computing Research Center's (AHPCRC) Graphics and Visualization Laboratory (GVL) at the University of Minnesota require a tremendous amount of I/O bandwidth and this appetite for data is growing. Silicon Graphics computers are used to perform the post processing, visualization, and animation of multi-terabyte size datasets produced by scientific simulations performed on AHPCRC supercomputers. The M.A.X. Experiment was designed to find the maximum achievable I/O performance of the Silicon Graphics CHALLENGE/Onyx class machines that run these applications. Running a fully configured Onyx machine with 12 - 150MHz R4400 processors, 512MB of 8-way interleaved memory, 31 fast/wide SCSI-2 channels each with a Ciprico disk array controller we were able to achieve a maximum sustained transfer rate of 509.8 megabytes per second. However, after analyzing the results it became clear that the true maximum transfer rate is somewhat beyond this figure and we will need to do further testing with more disk array controllers in order to find the true maximum.

Introduction

The Silicon Graphics CHALLENGE/Onyx computer system has an enormous I/O bandwidth that, to our knowledge, has not been fully explored. Researchers at the AHPCRC are working on projects that require significant I/O bandwidth from these computer systems [Woodward93]. The M.A.X. Experiment was designed to find the total sustainable I/O bandwidth of the CHALLENGE/Onyx computer systems that are key to these projects.

The cost of putting together enough high-speed disk subsystems to push the limits of the I/O bandwidth was expensive and remains so to this day. A fully configured CHALLENGE/Onyx computer system could support 32 fast-wide SCSI-2 channels each with 20 MBytes/second[1] of I/O bandwidth. Each SCSI channel would require a minimum of 5 high performance disk drives to saturate the 32 SCSI channels sufficiently to find the maximum I/O bandwidth. This would require a total of 160 disks which implies a great deal of device management and bus contention if these devices are not managed properly.

Instead of using individual disk drives, we connected a single high-speed disk array controller to each of 31 SCSI channels[2] on the Onyx system. These disk array controllers are much easier to obtain than disks and fewer of them are needed due to their individual high bandwidth. Furthermore, each disk array controller can easily saturate a single fast/wide SCSI-2 channel so fewer devices are needed (one per channel) resulting in less device management overhead.

Experimental Setup

Software

- IRIX Version 5.2, a UNIX SystemV Release 4 derivative
- lv - The Silicon Graphics Logical Volume Device Driver

Hardware

Onyx System Configuration
The system used in this experiment was a Silicon Graphics Onyx machine with the following configuration:

- 20 150 MHz R4400 Processors (12 Processors for 8-way interleaved memory configuration)
- CPU: MIPS R4400 Processor Chip Revision: 5.0
- FPU: MIPS R4010 Floating Point Chip Revision: 0.0
- Data cache size: 16 Kbytes
- Instruction cache size: 16 Kbytes

---

[1]MBytes/second = 1,000,000 bytes per second.
[2]Only 23 of the 24 available channels were used due to a minor cabling oversight on the part of the experimenters.

- Secondary unified instruction/data cache size: 1 Mbyte
- Main memory size: 512 Mbytes, 4- and 8-way interleaved
- 4 IO4 Power Channels
- 32 Fast-Wide Differential SCSI-2 channels
- 2GB System disk on SCSI channel 1

An Onyx system is basically a CHALLENGE with a graphics engine. Since this experiment did not make use of the graphics engine in the Onyx at any time, these results can be considered equally valid for a CHALLENGE.

Ciprico Disk Array and Diskless Array Description
The disk devices used in this experiment were Ciprico RF6710 disk arrays. Each RF6710 disk array is a RAID-3 device made up of 8 data drives plus 1 parity drive[Ciprico 93][Paterson89]. The number and type of disk arrays used were:

- 8 real disk arrays populated with Seagate ST12400N 2.5GB 3.5-inch disks.
- 23 diskless arrays populated with simulated Seagate Barracuda-2 2.5GB 3.5-inch disks.

Because the number of disks required to populate 31 disk array controllers was more than we could purchase or borrow, there were no disks on 23 of the 31 disk array controllers. Instead, they were programmed to act like *real* disk arrays when accessed. The *diskless* array controllers read and wrote data as any disk array would with the exception that data written to the diskless arrays was thrown away and data read was always zero. Consequently, no file system testing was possible and all testing was performed on *raw* devices.

The diskless array controllers have geometry characteristics based on the ST12400N disks but performance characteristics based on an array populated with Seagate Barracuda-2 disks, the higher performance version of the ST12400N disk. The data read from the array is always zero with the exception of the first 512-byte block on the array which will be kept in the controller memory and contains the volume header information.

The performance of the diskless array controllers depends on the type of access. For purely sequential access the seek and rotational latencies are zero. This is because on array controllers with real disks, sequential read and write operations make effective use of the data caches on the individual disks thus hiding rotational and seek delays. For any other access that involves a seek, an appropriate delay is inserted in the command processing to simulate the seek and rotational latencies. The seek time is estimated to be proportional to the seek distance and the rotational delay is set to half a revolution (4.1 milliseconds in this case). The disk drive being modeled is a Seagate Barracuda-2. The seek simulation feature was used for a different set of experiments but was not used in the M.A.X. experiment.

For sequential read operations, the performance of the diskless arrays was only 4% higher than an array disk real disks at moderate to large request sizes (Figure 1). Sequential write operations on the diskless arrays performed nearly identical to the read operations. It should be noted that the objective of this experiment was not to simulate a disk array but rather to saturate the I/O subsystem. Therefore, these performance differences are more of a benefit than a detriment.

Finally, the read operations on the real disk arrays perform better than write operations on real disks even when the write caches are used (Figure 2). However, this difference seems to be reasonably constant for small request sizes and becomes less significant at larger request sizes.

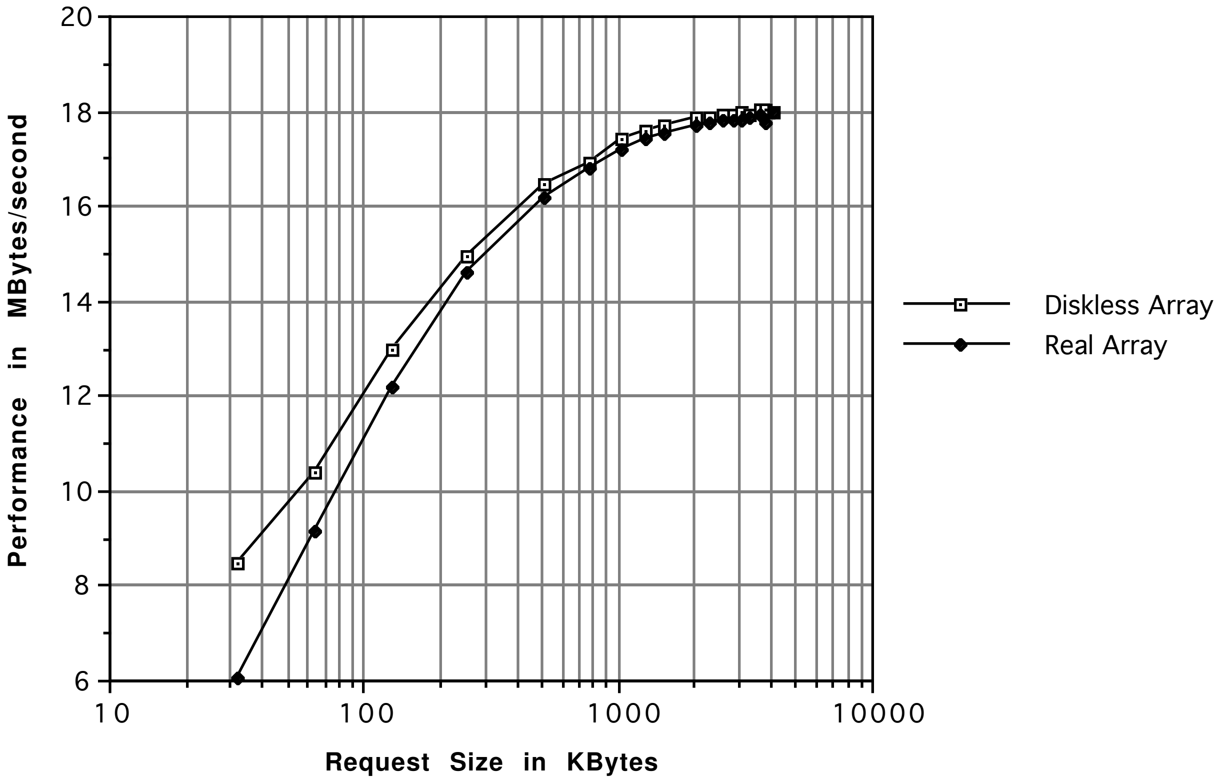


Figure 1. *Performance of read operations for diskless and real disk arrays for request sizes ranging from 32KBytes to 4096KBytes. At the lower request sizes, the diskless arrays are considerably faster than the real disk arrays. However, the performance curves converge at request sizes of 512KBytes and higher.*

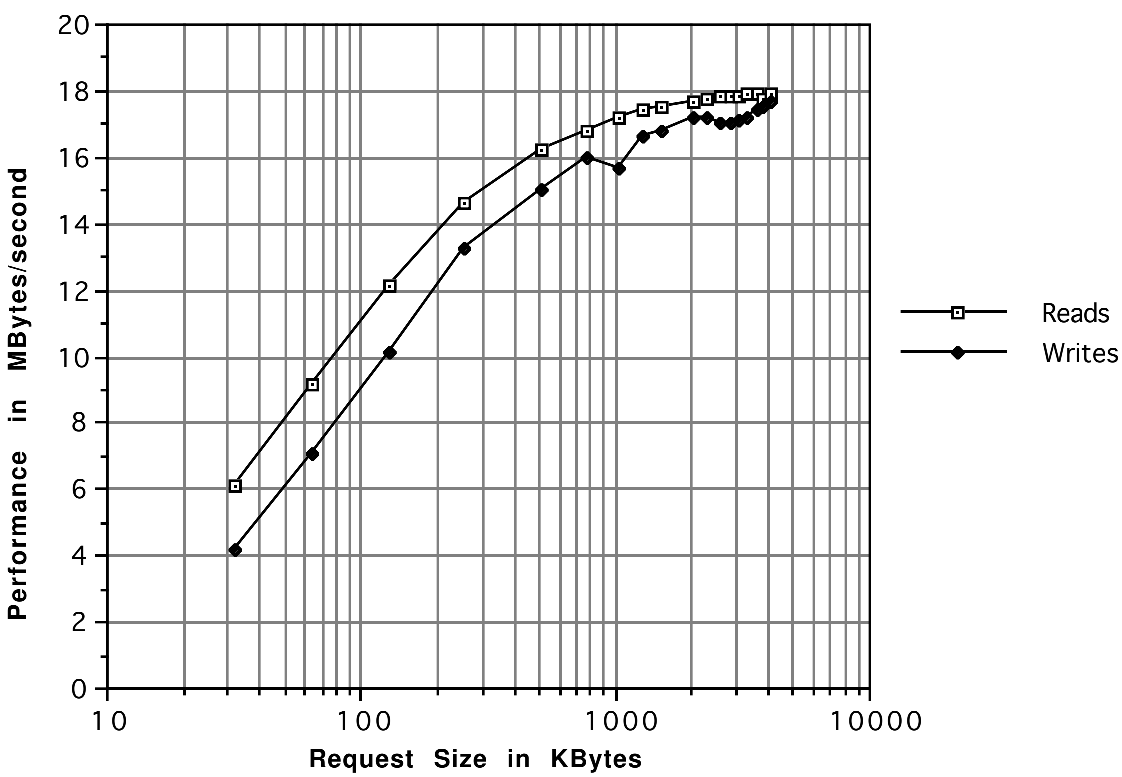
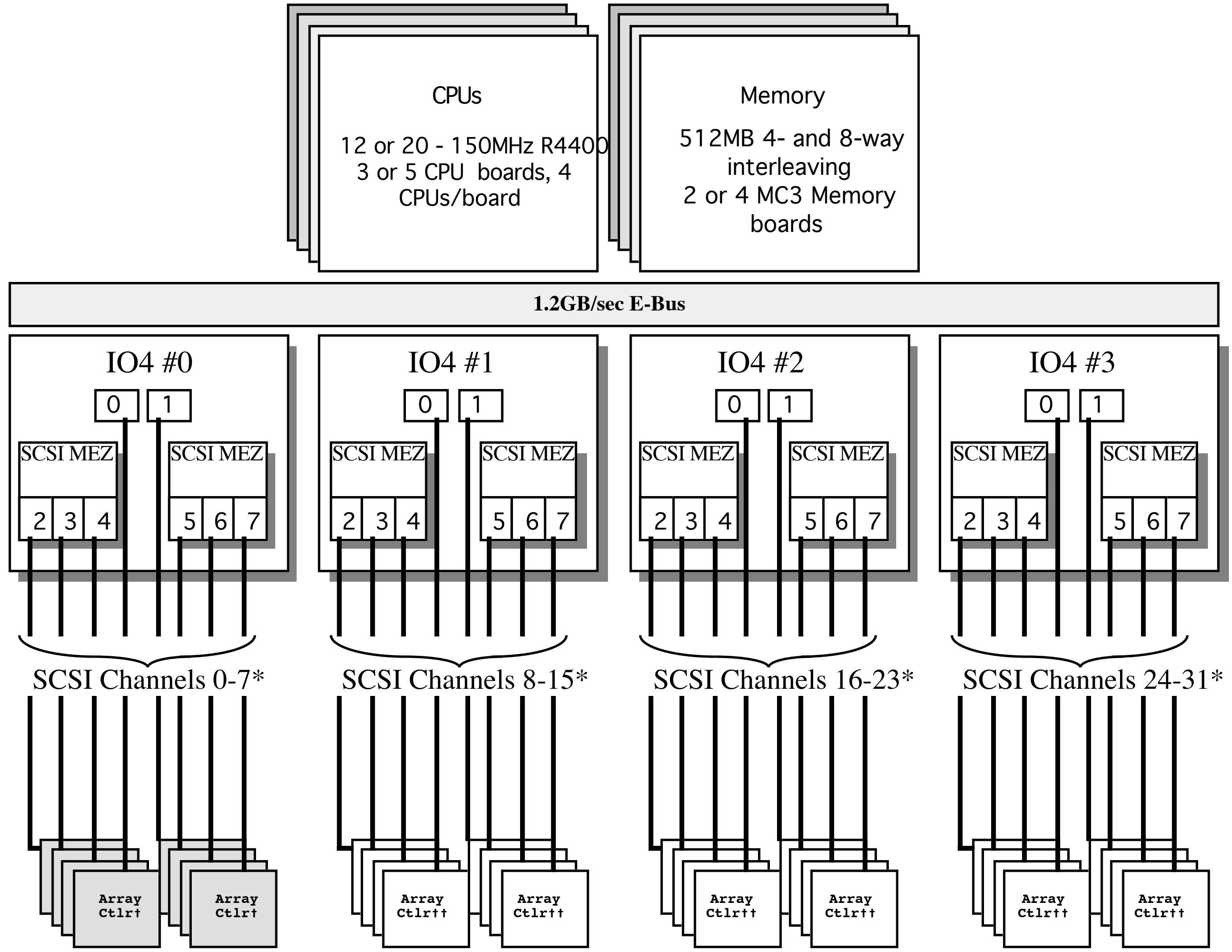


Figure 2. *Performance of a reads and writes versus request size on a real disk array. The write operations are cached on each of the individual disk drives within the array.*



* These are Ordinal number assignments. Actual channel numbers were different.
† These array controllers have real disks attached.
†† These array controllers have no disks attached. These are referred to as Diskless Arrays.

Figure 3. M.A.X. hardware configuration diagram.

Performance Evaluation Program

xdd - An I/O performance measurement tool

xdd is a program developed to measure I/O performance by reading or writing large amounts of data sequentially from a file or raw device. This program is intended to find the upper limit of performance of an I/O subsystem under specific, well-controlled operating parameters. *xdd* takes as command line parameters the target device to operate on, the operation to perform (read or write), the request size to use for each read/write operation, the number of read/write requests to perform, and the number of times to repeat the test in order to obtain a good statistical average. Furthermore, *xdd* can be instructed to limit the time to run each test in order to make the runtime more deterministic.

xdd provides three measures of I/O performance: (1) an aggregate transfer rate, (2) a table of time stamps detailing each request, and (3) the number of I/O operations completed during the test. Upon completion, *xdd* prints a single line of values indicating the request size (in 1024-byte blocks) , the average, high, and low I/O performance in units of 10^6-bytes per second, the number of I/O operations, the average, maximum, and minimum number of seconds to complete the specified number of requests, and the number of errors that occurred during the test.

The first set of performance values is the aggregate transfer rate and can be affected erroneously by individual I/O operations that may have "stalled" due to some outside influence. To help identify these outlying values a collection of high resolution time stamps are recorded in a file for further analysis. Before each I/O operation has been initiated, a time stamp is recorded in an internal memory array. This array is pre-allocated and page locked in order to avoid any paging interference that may negatively affect these values. After *xdd* has completed all passes of the requested test, the time stamp values are written to a file with header that contains the request size in 1024-byte blocks, the resolution of the time stamp values, and the number of time stamp entries.

In an attempt to minimize the impact of virtual memory management and process scheduling, the *xdd* text and data areas, the I/O buffer, and the time stamp table are page locked during initialization to avoid any page faults or program swapping during the performance test. The program also sets itself to a non-degrading, high priority in order to reduce scheduling side effects on the measurements.

xdd uses a single page-aligned memory buffer large enough to handle a single request. An I/O request to a single disk can range in size from 512-bytes up to a system defined maximum. Currently, this maximum is set to 4 MBytes (4*1024*1024 bytes), or more appropriately, 1024 pages[3]. The IRIX operating system allocates 1024 page mapping registers for each I/O request but in order to map any arbitrary 4MB I/O request, 1025 page mapping registers are required to map requests that do not start on page boundaries. Therefore, in order to issue an I/O request of 4MB it is necessary to page align the buffer to insure it can be mapped in 1024 page mapping registers.[4]

The Experiment

First, a test utilizing eight fast/wide SCSI-2 channels on a single IO4 was run to determine if the IO4 imposed any bandwidth limitations on the eight channels. The aggregate performance scaled linearly as the number of independently fully utilized channels was increased from 1 to 8. Hence, there are no bandwidth limitations within an IO4.

The principle testing involved three basic *access methods*. The first access method was the simultaneous *independent* access of 1 to 31 disk array controllers. The second access method used the Silicon Graphics Logical Volume (lv) striping device driver to access 2 to 31 devices as a single logical device. The third access method was a variation of the first whereby half the disk arrays would be reading data into memory while the other half would be writing data from memory to disk. This last test was intended to measure any bi-directional interference.

Each of these tests were performed using 4- and 8-way memory interleaving. The greater the interleaving, the higher the effective bandwidth into memory. Figure 4 describes the overall experimental test layout..

Independent Access		Logical Volume		Simultaneous Read/Write	
4-way Interleaved	8-way Interleaved	4-way Interleaved	8-way Interleaved	4-way Interleaved	8-way Interleaved
1-31 devs read and write	1-31 devs read	1-31 devs read and write	1-26 devs read	2-30 devs read/write	2-30 devs read/write

Figure 4. *The access methods and system memory configurations.*

Due to time constraints, write operations were not tested for the 8-way interleaved Independent Access and Logical Volume tests. However, it was observed in the 4-way interleaved memory testing that the overall

[3]The page size in IRIX 5.x is fixed at 4096-bytes.

[4]This problem with one to few page mapping registers exists in IRIX 5.2 but may not exist in later releases.

write performance tended to be slightly better than the read performance. It is believed that this characteristic holds true for the 8-way interleaved memory as well although it still needs to be verified.

Caveats

- In order to accommodate a shorter than expected testing schedule the 2-way interleaved memory testing was removed.
- The fully configured Onyx with 4-way interleaved memory was able to accommodate 20 processors (5 processor boards). However, the 8-way interleaved memory configuration required 2 extra memory boards that displaced 2 processor boards reducing the number of CPUs to 12 for this configuration. However, it should be noted that this would not be necessary on a CHALLENGE server which can be configured with 36 CPUs, 8-way interleaved memory, and 4 IO4s all at the same time.
- The diskless array controllers were measured to be about 4% faster than the real disk arrays at the top end of their performance curve (18.1 MBytes/sec versus 17.85 MBytes/sec).
- It is interesting to note that even with only 12 CPUs on the 8-way interleaved memory configuration, the I/O rate did not appear to be limited by the CPU performance.

Results

The results are presented by access method as described in figure 4. First the Independent Access results are presented (figures 5-9) followed by the Logical Volume results (figures 10-16) and finally the Simultaneous Read/Write results are presented (figure 17).

Independent Access Results

The total bandwidth of the 4-way interleaved memory configuration was tested by increasing the number of independently accessed arrays from 1 to 31 over request sizes ranging from 64KBytes[5] to 4096KBytes. Disk array controllers were added one at a time incrementing monotonically through each IO4 until all channels were running. This procedure was repeated for the 8-way interleaved memory configuration.

This access method yielded the best overall performance when compared to the logical volume and simultaneous read/write access methods. The 4-way interleaved memory configuration peaked at 392 MBytes/second accessing 27 devices with a request size of 768KBytes, dropping to 310 MBytes/second as more devices were added (figures 5-6). The 8-way interleaved memory configuration performance was measured at 509.8 MBytes/second accessing 31 devices with a request size of 2048KBytes (figures 7-8). Request size has a definite effect on the performance with request sizes larger than 512KBytes performing the best (figure 9).

Due to time constraints, testing was limited to read operations only.

Logical Volume Read and Write Tests

This series of tests were run to measure the read and write performance of logical volumes composed of 9 to 30 devices. Since a previous study [Ruwart93] characterized the read performance of logical volumes composed of 2 to 8 devices it was decided to start where that study left off in the interest of time.

The results of these tests are reported as Performance as a function of number of devices at two different *step sizes*. The step size of a logical volume is the maximum amount of data read off a single disk array in a single request. Thus, from the disk array's perspective, the step size is equivalent to a request size because this is what the disk array sees as a request from the host. The amount of data the *xdd* application actually requests from the logical volume was intentionally set to the step size times the number of devices in the logical volume in order to insure that all devices in the logical volume would be accessed for each application I/O request in the most optimal manner.

---

[5] 1 KByte = 1024 bytes.

As expected, the larger step size of 1024KBytes performed better than the smaller step size of 256Kbytes (figures 10-15). However, the performance did not seem to depend on the type of operation (figures 12 and 15) and slightly on the memory interleaving (figure 16). The peak performance of the logical volume access method was about 240 MBytes/second.

Simultaneous Read/Write Tests

The simultaneous read/write tests were run to measure any bi-directional interference when transferring data to and from different groups of I/O devices simultaneously. The motivation behind this testing has to do with large multi-media servers that must sustain a large bandwidth in *and* out of a system.

The results show a peak performance of 482 MBytes/second accessing a total of 30 disk arrays: 15 reading plus 15 writing using a request size of 1536KBytes and 8-way interleaved memory (figure 17). This is 97% of the straight read performance of 30 independent disk arrays. The 3% difference is attributed to the slightly lower performance of the individual disk array write operations (see figure 2). Since 15 of the 30 devices were writing data in the simultaneous read/write case, the aggregate performance of all 30 disk arrays is less than if all 30 devices were reading.

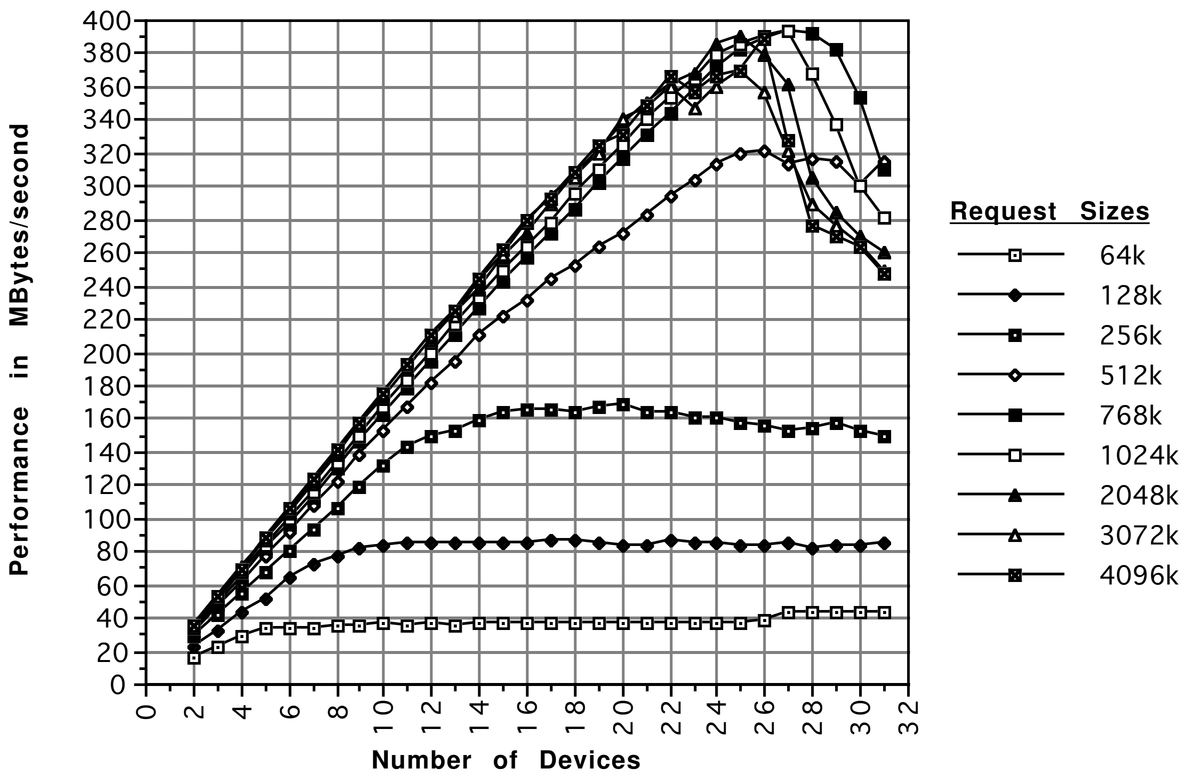


Figure 5. *The performance curves for read operations using request sizes 64-4096-KBytes using 4-way interleaved memory. The performance peaked at 393 MBytes/second using 27 devices with a request size of 768-KBytes.*

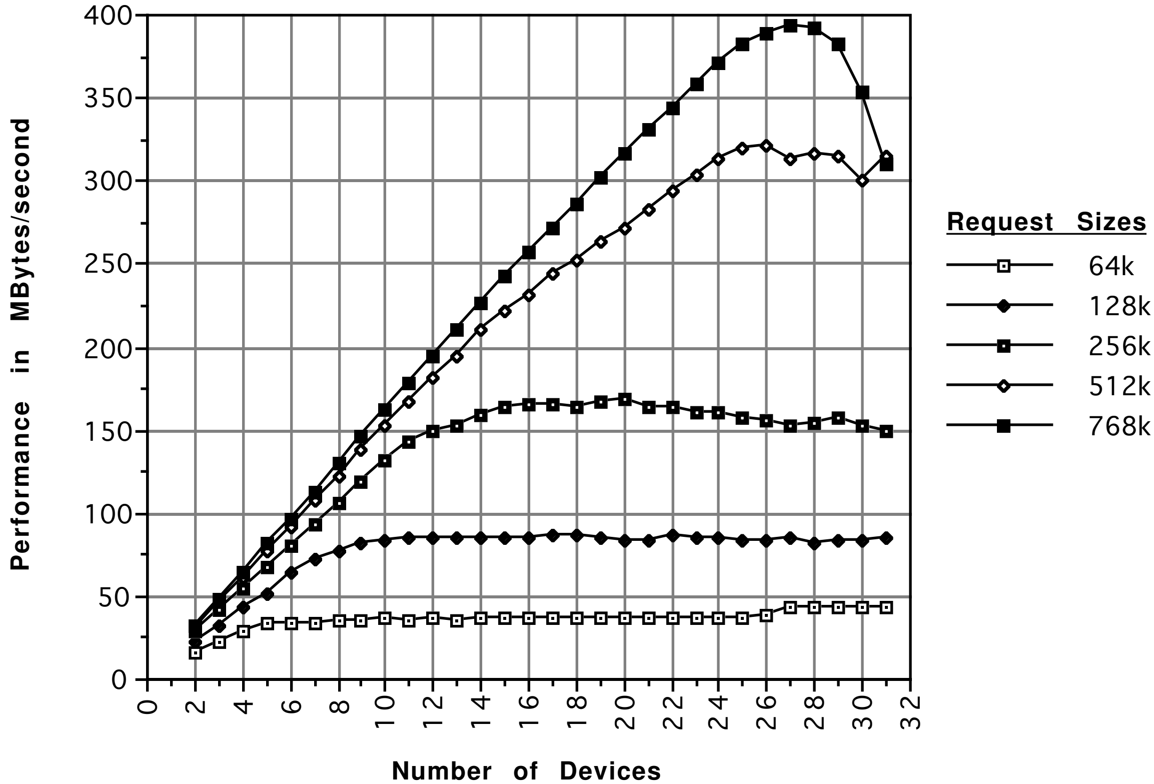


Figure 6. *The performance curves for read operations using request sizes 64-768-KBytes using 4-way interleaved memory. The performance peaked at 393 MBytes/second using 31 devices with a request size of 768-KBytes.*

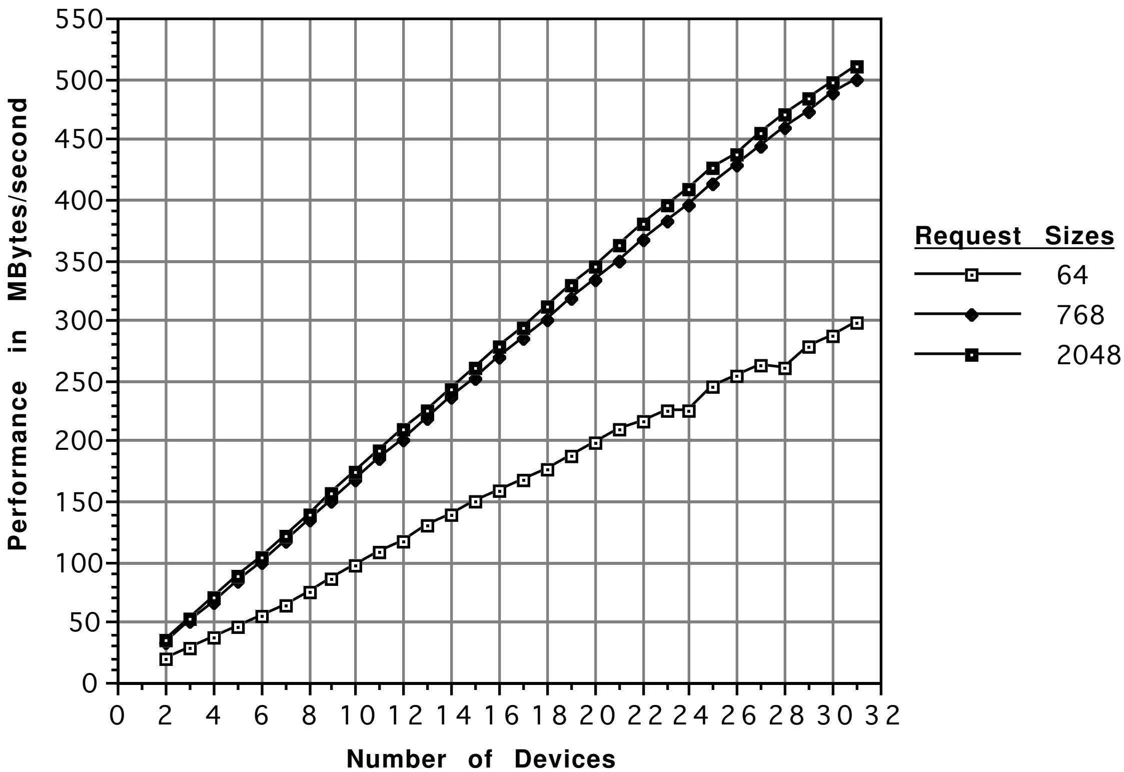


Figure 7. *The performance curves for read operations using request sizes 64, 768, and 2048-KBytes using 8-way interleaved memory. The performance peaked at 509.8 MBytes/second using 31 devices with a request size of 2048-KBytes.*

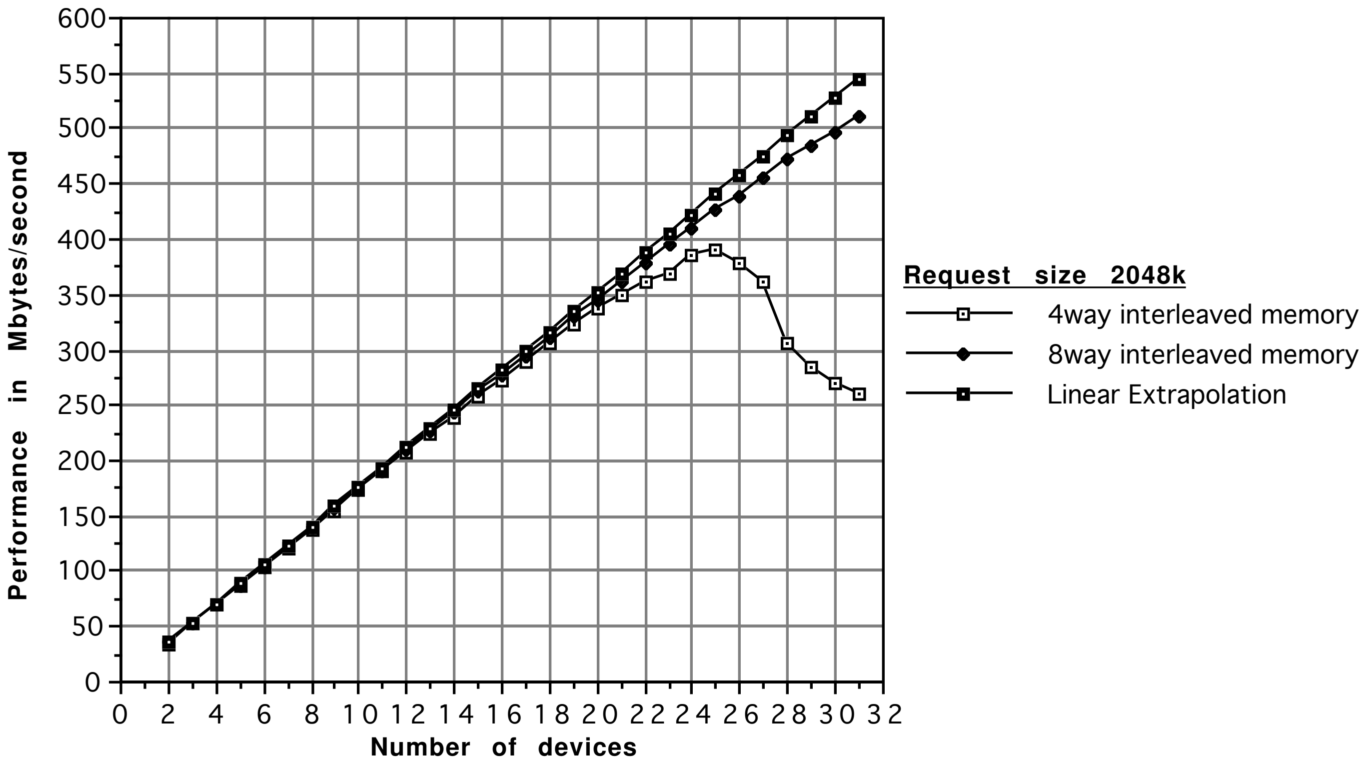


Figure 8. *The performance curves for read operations using a request size of 2048 KBytes, 8-way versus 4-way interleaved memory, for independent processes accessing 2 to 31 disk arrays. The performance using 4-way interleaved memory tracks the 8-way performance curve up to 390MBytes per second where it drops off noticeably while the 8-way performance curve continues with no signs of tapering off.*

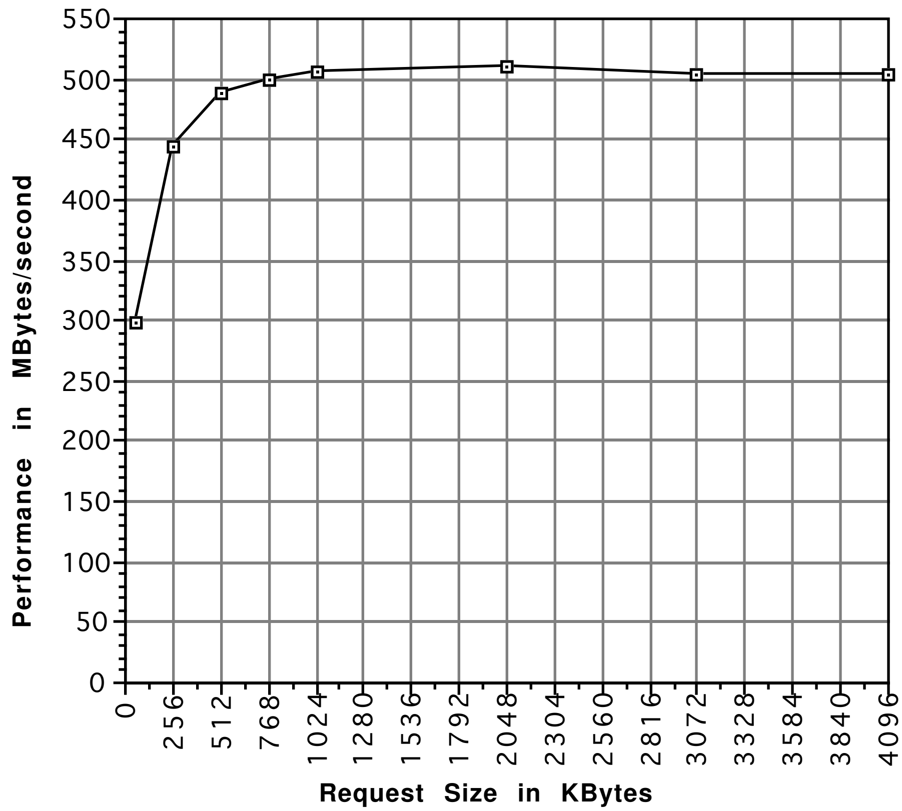


Figure 9. *The performance curve for read operations using request sizes 64-4096-KBytes using 8-way interleaved memory. The performance peaked at 509.8 MBytes/second using 31 devices with a request size of 2048-KBytes*

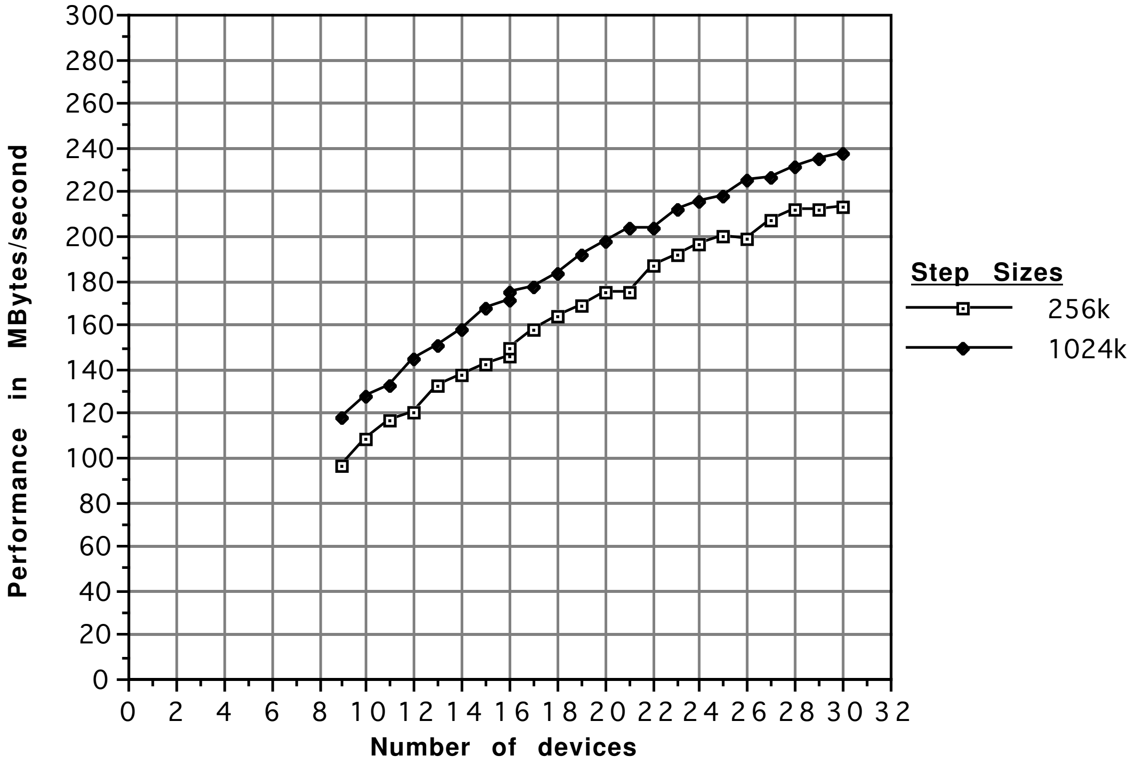


Figure 10. *The performance curves for read operations using step sizes 256 KBytes and 1024 KBytes, 4-way interleaved memory, and a single logical volume consisting of 9 to 30 disk arrays. The performance peaked at 236.9 MBytes/second using 30 devices with a step size of 1024-KBytes.*

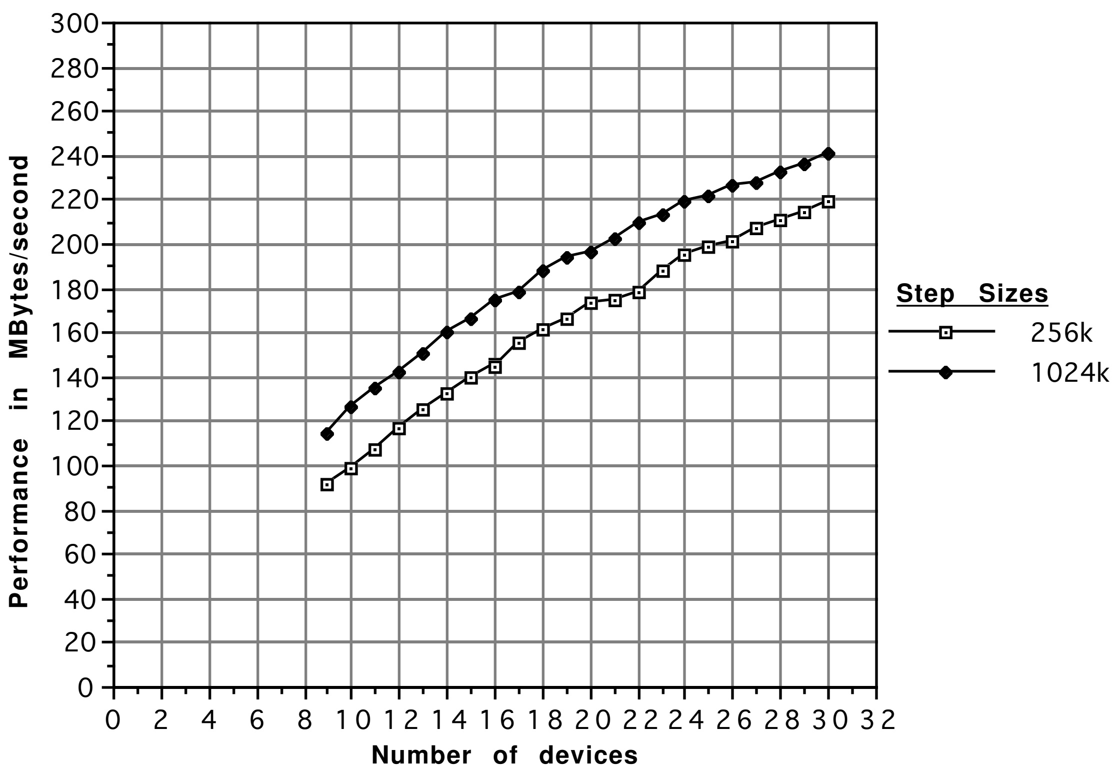


Figure 11. *The performance curves for write operations using step sizes 256 KBytes and 1024 KBytes, 4-way interleaved memory, and a single logical volume consisting of 9 to 30 disk arrays. The performance peaked at 241 MBytes/second using 30 devices with a step size of 1024-KBytes*

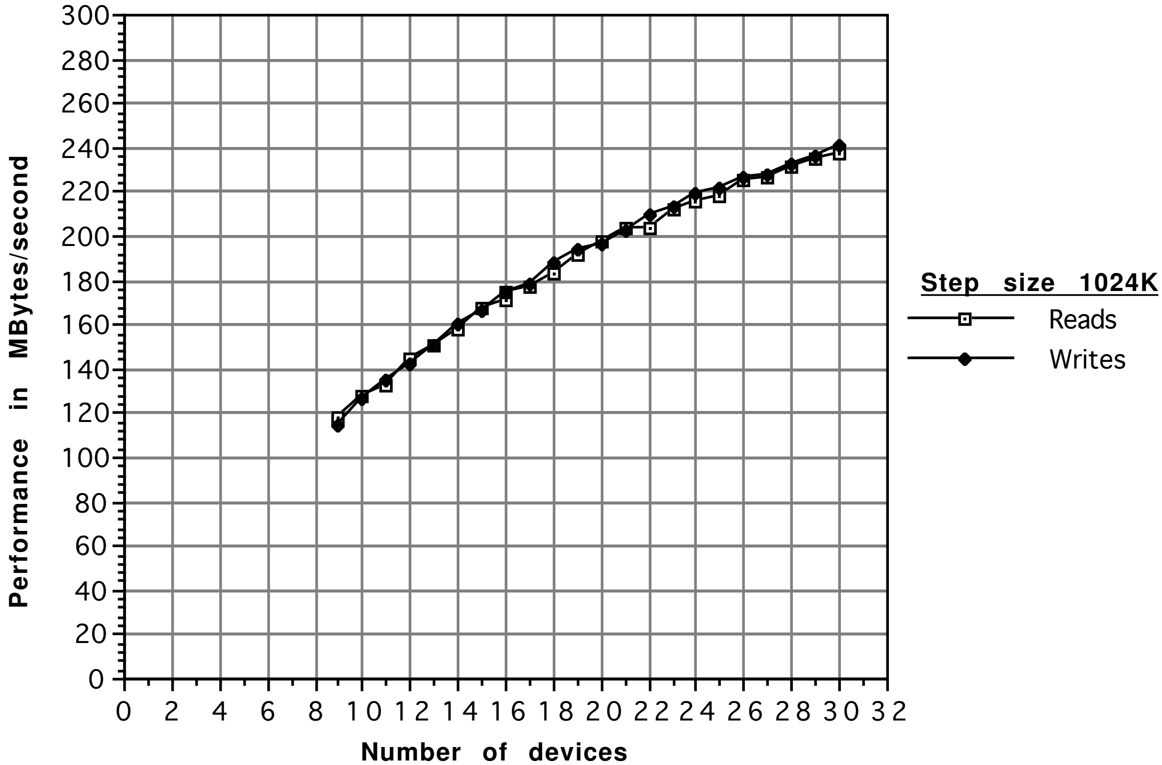


Figure 12. *The performance curves for read and write operations using a step size of 1024 KBytes, 4-way interleaved memory, and a single logical volume consisting of 9 to 30 disk arrays. The performance of the write operations was slightly better than the read operations.*

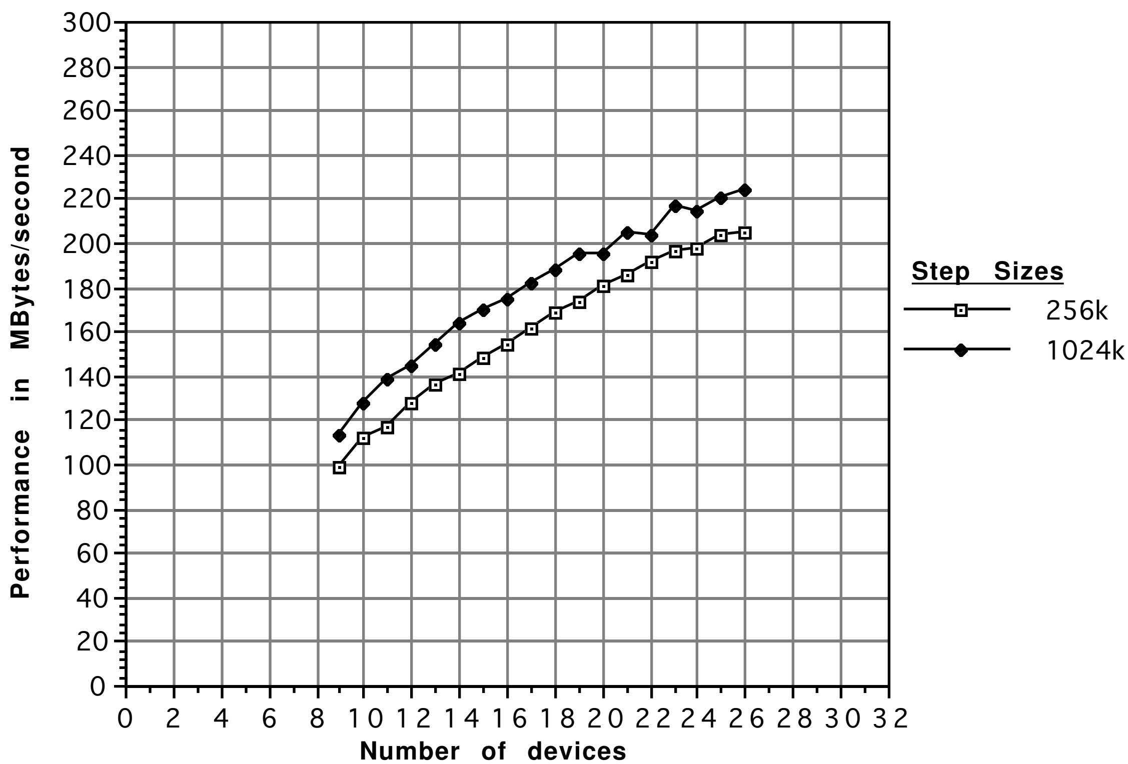


Figure 13. *The performance curves for read operations using step sizes 256 KBytes and 1024 KBytes, 8-way interleaved memory, and a single logical volume consisting of 9 to 26 disk arrays. The performance peaked at 223.9 MBytes/second using 26 devices with a step size of 1024-KBytes.*

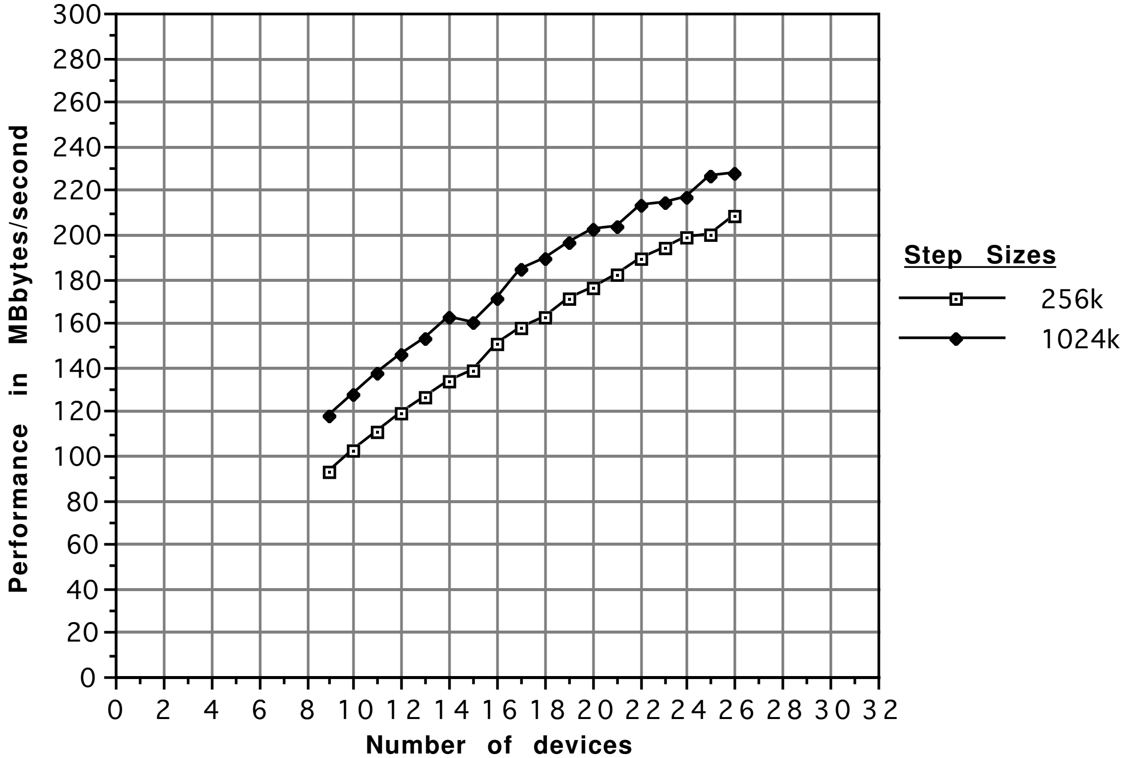


Figure 14. *The performance curves for read operations using step sizes 256 KBytes and 1024 KBytes, 8-way interleaved memory, and a single logical volume consisting of 9 to 26 disk arrays. The performance peaked at 228.2 MBytes/second using 26 devices with a step size of 1024-KBytes.*

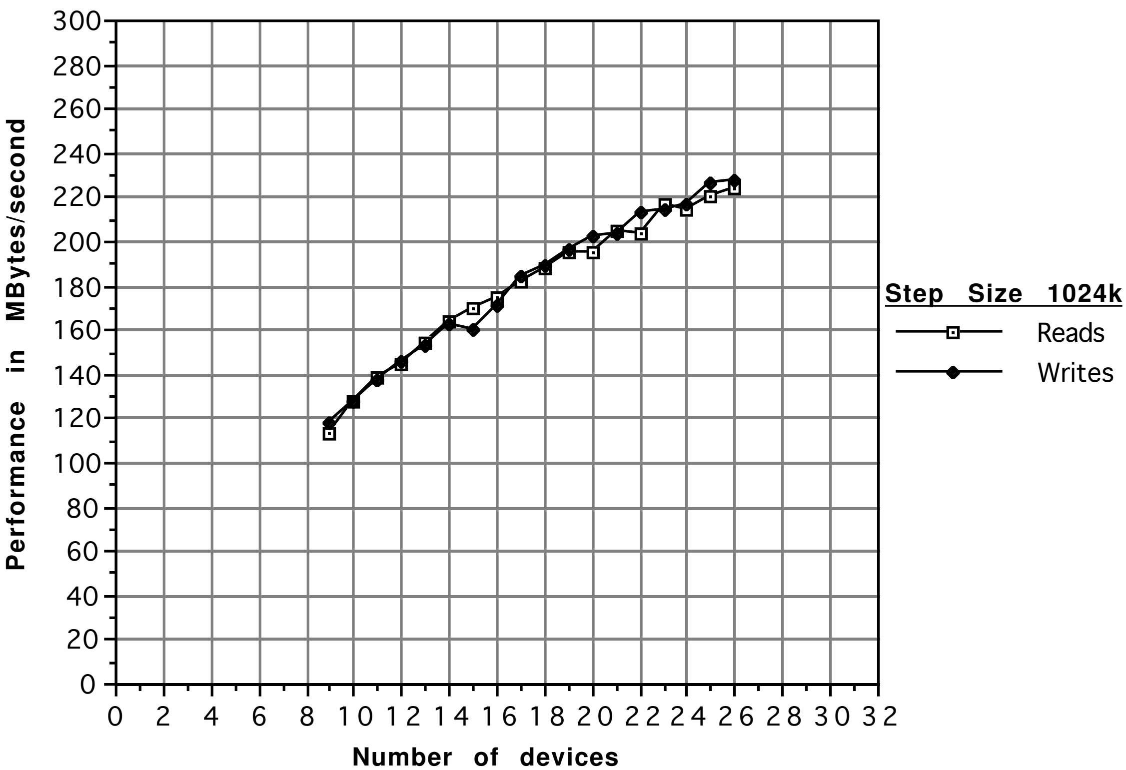


Figure 15. *The performance curves for read and write operations using a step size of 1024 KBytes, 8-way interleaved memory, and a single logical volume consisting of 9 to 26 disk arrays. The performance of the write operations was slightly better than the read operations in most cases but are still very close*

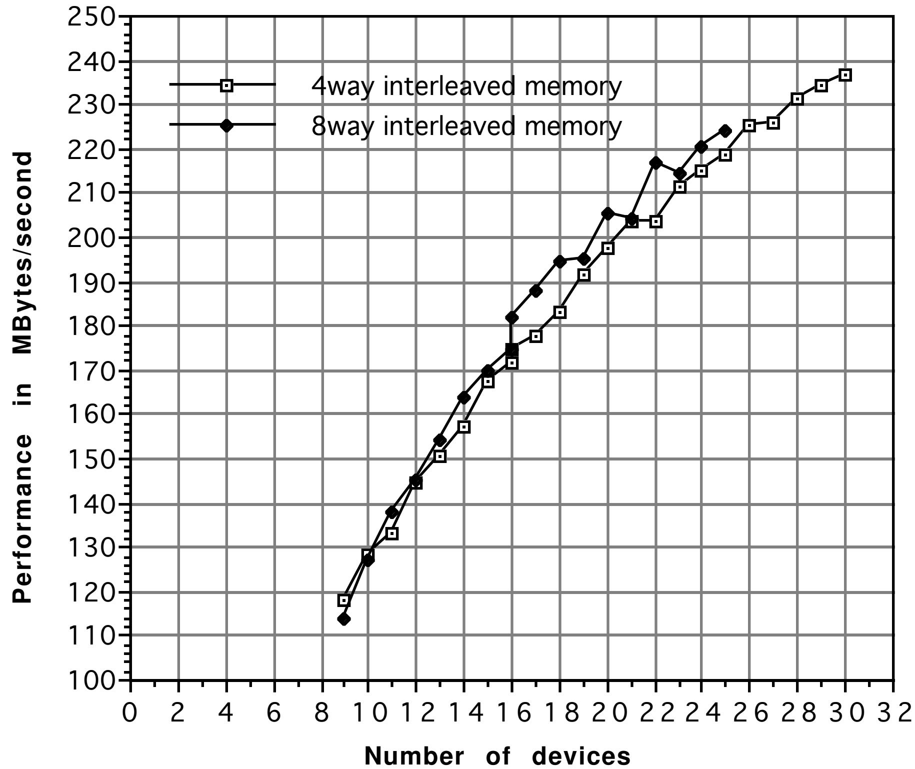


Figure 16. *The performance curves for read operations using a step size of 1024 KBytes, 8-way versus 4-way interleaved memory, and a single logical volume consisting of 9 to 30 disk arrays (26 for the 8-way case). The performance using 8-way interleaved memory was slightly better than the 4-way interleaved memory configuration.*

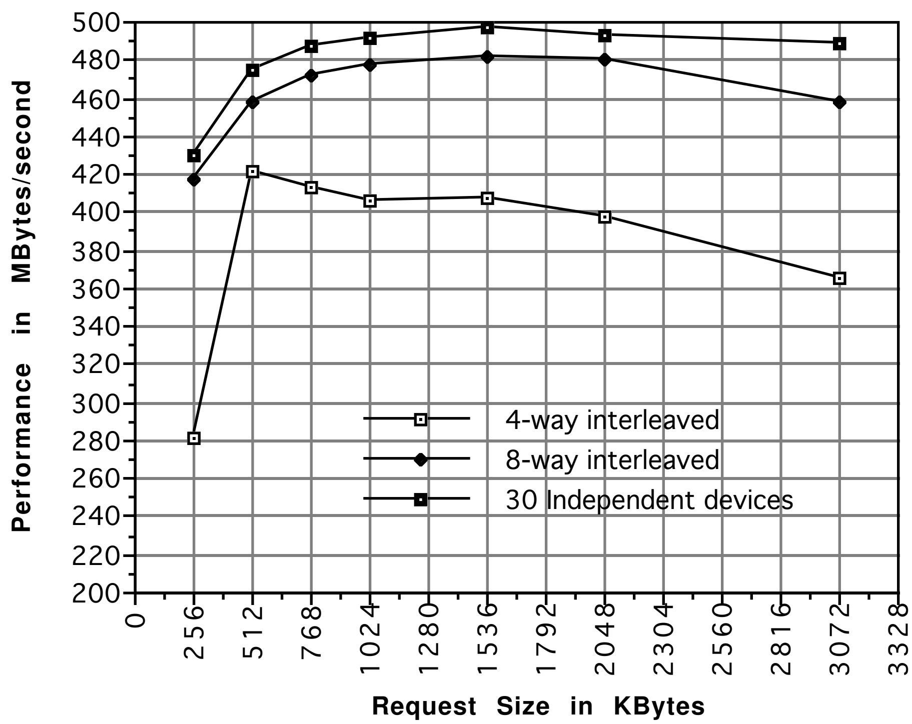


Figure 17. *The performance curves for simultaneous read and write operations using a request sizes ranging from 128KBytes to 3072KBytes, 8-way versus 4-way interleaved memory, reading from 15 disk arrays while writing to 15 other disk arrays. The performance using 8-way interleaved memory was consistently better than the 4-way interleaved memory configuration. The top curve represents read operations on 30 independent devices.*

Conclusions

The M.A.X. experiment demonstrated a sustained performance of 509.8 MBytes/second reading data from 31 independent disk arrays simultaneously into an 8-way interleaved memory subsystem on the CHALLENGE/Onyx system. However, the *maximum* achievable transfer rate was not observed because 31 disk arrays were not enough to saturate the I/O subsystem. This statement is based on the results for the 4-way interleaved memory configuration whereby the performance hits a maximum and degrades as more devices are added. This effect was not observed for the 8-way interleaved memory configuration. Therefore, we believe that the actual *maximum* I/O performance of the CHALLENGE/Onyx is greater than 510 MBytes/second.

The logical volume testing showed a maximum transfer rate of approximately 240 MBytes/second for reading or writing. The memory configuration did not have any effect on the overall performance of any logical volume configuration.

Finally, the simultaneous read/write tests demonstrated a maximum performance of 482 MBytes/second using 30 disk arrays: reading from 15 while simultaneously writing to 15 others. Since this performance is measured over 30 devices, it is estimated that 31 devices would provide an additional 16 MBytes/second for a total sustained performance of 498 MBytes/second.

The M.A.X. experiment was a success and exceeded our expectations inasmuch as we expected to observe a peak performance less than 500 MBytes/second. Had we known that the peak would have been higher we would have designed the experiment to utilize far more disk array controllers and SCSI-2 channels. The Silicon Graphics CHALLENGE/Onyx system architecture has proven to have a very efficient I/O subsystem that has a tremendous usable bandwidth.

Future Work / Related Work

- *Perform 8-way interleaved memory testing on a CHALLENGE and more processors, 6 IO4's, and 48 fast/wide SCSI-2 channels with a theoretical peak bandwidth of 960MBytes/second.*
- *File System Testing with 160 Real Disks and/or 32 <u>Real</u> Disk Arrays*
- *Testing with Multiple 100-MByte/second HiPPI and/or Fibre Channel Devices*
- *Bit rate consistency testing for multimedia applications*

Acknowledgments

Silicon Graphics, Inc. for providing the hardware required to attach the disk arrays to the Onyx machine and Ciprico, Inc. for providing the disk array controllers and engineering that went into making them believe they had real disks attached.

References

[Ciprico93] ``RF6700 Controller Board Reference Manual," Publication Number 21020236 A, Ciprico, Inc., Plymouth, MN, August 1993.

[Paterson89] D.A. Paterson, P.M. Chen, G. Gibson, and R.H. Katz, ``Introduction to redundant arrays of inexpensive disks (raid)," Proc. IEEE Compcon, Spring 1989.

[Ruwart93] T.M.Ruwart and M.T. O'Keefe, ``Performance Characteristics of a 100MB/second Disk Array," Army High Performance Computing Research Center Preprint Series, no. 93-123, 1993.

[Woodward93] P.R. Woodward, ``Interactive Scientific Visualization of Fluid Flow,"IEEE Computer, vol. 6, no. 10, pp. 13-26, October 1993.