Final Report
National Aeronautics and Space Agency
Advanced Research Projects Agency

Diskless Supercomputers: Scalable, Reliable I/O for the Tera-Op Technology Base

Randy H. Katz
Principal Investigator

John K. Ousterhout and David A. Patterson
Co-Principal Investigators

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720

1. Project Overview

Computing is seeing an unprecedented improvement in performance; over the last five years there has been an order-of-magnitude improvement in the speeds of workstation CPUs. At least another order of magnitude seems likely in the next five years, to machines with 500 MIPS or more. The goal of the ARPA Teraop program is to realize even larger, more powerful machines, executing as many as a trillion operations per second. Unfortunately, we have seen no comparable breakthroughs in I/O performance; the speeds of I/O devices and the hardware and software architectures for managing them have not changed substantially in many years.

We have completed a program of research to demonstrate hardware and software I/O architectures capable of supporting the kinds of internetworked "visualization" workstations and supercomputers that will appear in the mid 1990s. The project had three overall goals:

High Performance. We developed new I/O architectures and a prototype system that can scale to achieve significant factors of improvement in I/O performance, relative to today's commercially-available I/O systems, for the same cost. This speedup can be achieved using a combination of arrays of inexpensive small format (3.5") magnetic disks [Patterson 88] coupled with a file system well-suited to a striped file organization [Ousterhout 89].

High Reliability. To support the high-performance computing of the mid-1990's, I/O systems cannot just be fast; they must also be large. The unreliability of current disks requires nightly backups on magnetic tape. This process is expensive and unmanageable even with our current systems, and will not scale to meet the needs of the 1990s. By using a small number of extra disks in our disk array to act as standby spares or error recovery disks, we have increased the availability of the magnetic media to the point where loss of data is no longer an issue. Our results make it possible to permit much larger file systems with increased file reliability and decreased operator intervention.

Scalable, Multipurpose System. Designers of mainframes and supercomputers have built machine-specific I/O processors, requiring a substantial redesign for each new generation. We are structuring our I/O architecture around emerging standard gigabit networks and high-performance file servers. Architects of new computers can now assure themselves high-performance I/O simply by providing an efficient network interface; they will not need to implement a large collection of controllers and channels for different I/O devices. As demands for I/O increase, more I/O systems can be added to a network if there is enough bandwidth available, or extra network interfaces if there is not. Moreover, since the I/O systems will be built from commercially-available computers, they can act either as I/O processors for supercomputers or as "super" file servers on the network. We have demonstrated how to build a multipurpose next-generation I/O system from off-the-shelf disks and CPUs.

2. Project Achievements

2.1. Hardware Group

Our research group has been responsible for laying the foundations for the fundamental algorithms needed in a disk array-based storage system, such as parity interleaving and data reconstruction schemes. Theoretical investigations have focused on the trade-offs between performance and reliability in storage systems (Garth Gibson), adaptive I/O benchmarks (Peter Chen), disk array performance models (Edward Lee), hierarchical storage management (Ethan Miller), and reliable, high performance tertiary storage systems (Ann Drapeau). Garth Gibson's 1991 Ph.D. dissertation [Gibson 91] was a runner-up in the annual ACM Dissertation Award Competition. A recent paper by Peter Chen and David Patterson, based on Chen's dissertation research, won the best paper award at the 1993 ACM Sigmetrics Conference. The rest of the research group has been particularly productive, as evidenced by the list of papers referenced in Section 6.

Theoretical studies do not provide sufficient understanding by themselves. We also wanted to understand the engineering issues of constructing large scale disk array systems. We have completed and demonstrated RAID-II, a second generation prototype file server/array controller, which we describe next.

2.1.1. The Second RAID Prototype

RAID-II is a high performance file system that couples a disk array through a high bandwidth interconnection network to staging memory and a HIPPI-based network interface (see Figure 1). The

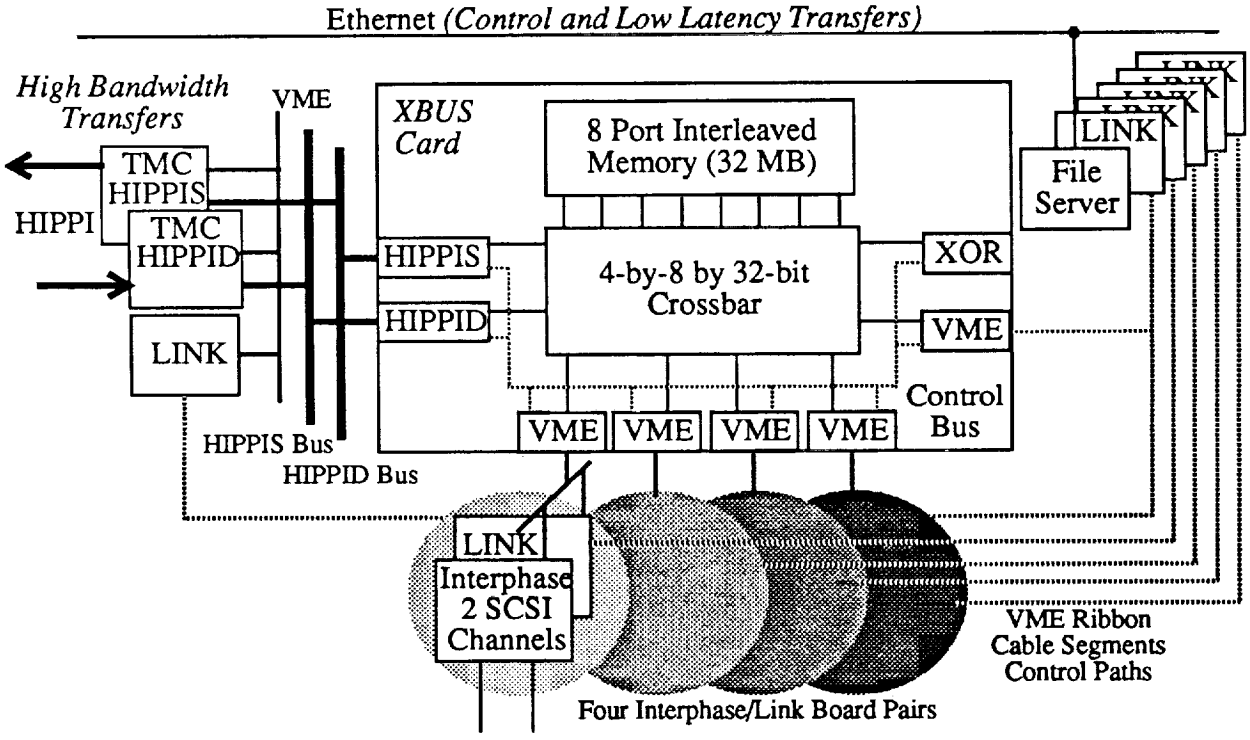

Figure 1: *RAID-II Organization*
A high bandwidth crossbar interconnection ties the network interface (HIPPI) to the disk controllers (ATC) via a multi-ported memory system. Hardware to perform the parity calculation is associated with the memory system. An internal control bus provides access to the crossbar ports, while external point-to-point VME links provide control paths to the surrounding SCSI and HIPPI interface boards.

architecture provides a high bandwidth connection between the network, memory, and the disk interfaces. The design is based on the observation that user data never needs to be seen by the file system and networking software that resides in the Server. The network interface, based on the HIPPI high speed interconnect, was provided by our industrial partner Thinking Machines Corporation of Cambridge, MA.

The crossbar provides twelve 32-bit, 40 MBytes per second ports. Four of these are dedicated to the memory system, which is the source or destination of all transfers on the interconnect. The remaining ports connect the crossbar to the TMC HiPPI source board, the TMC HiPPI destination board, the XOR unit, a VME control interface, and four ATC disk controller boards. The hardware resources of the RAID-II array controller are accessed over a VME-to-VME bus bridge that couples a server CPU to the controller's VME-based control bus. This path can be used to probe controller RAM, set-up the HiPPI boards, or configure the disk controller boards.

The original SCSI string subsystem was to be provided by Array Technologies Corporation of Boulder, CO. However, due to fragile VME interfaces, we were not successful in integrating these into our prototype. We made do with less capable disk controllers from Interphase Corporation.

The packaging design of RAID-II makes it possible to accommodate 144 3.5" formfactor disk drives in two 19" racks. Figure 2 illustrates the design of our integrated disk shelves. The disk drives were embedded in special mechanical housings to ease their "hot" replacement in the event of a failure. This capability was successfully shown at our January demonstration. An impedance controlled SCSI bus printed circuit board lies beneath the shelf to provide connectivity among the drives in a row. While the prototype uses 320 MByte drives from IBM Corporation, current technology has delivered over 1.2 GBytes in the same formfactor. Thus, in a two racks we have the capability to store over 172 GBytes.

We accomplished power distribution within the RAID-II chassis with embedded power supply shelves (see the disk racks in Figure 3). Each power supply shelf serves four drive shelves, two above and two below. This symmetric arrangement minimizes the need to cable high current, low voltage power from the supply to the drive. To provide a measure of fault isolation, each drive shelf is powered by an

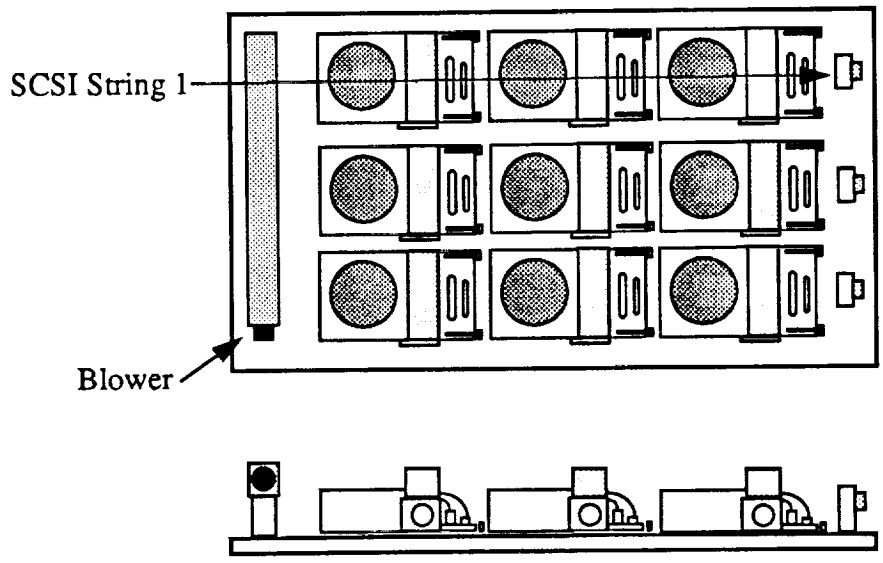

Figure 2: *RAID-II Disk Shelves*
The figure show the disk shelves used in the RAID-II prototype. Drives are placed flat in a 3-by-3 orientation on the shelf. This is in contrast to the "wall of disks," in which drives are placed on one face of the rack. For 3.5" disk drives, shelves yield a higher storage capacity per cubic foot than the disk wall.

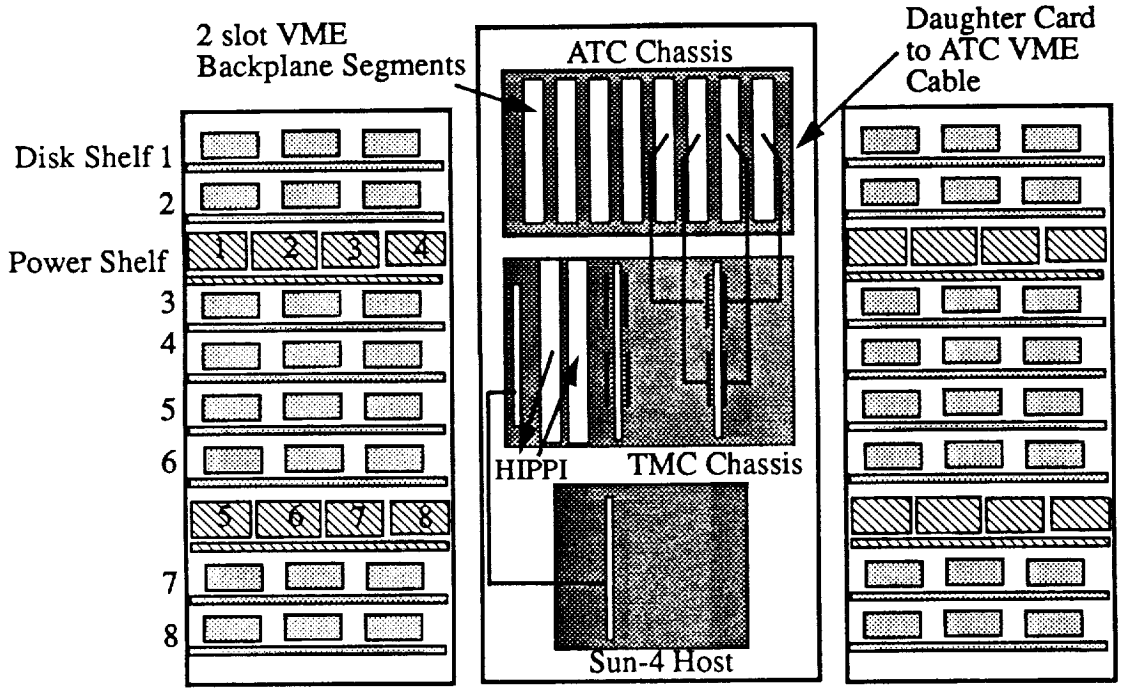

Figure 3: *RAID-II Hardware Racks*
The RAID-II hardware rack contains a SUN-4/280 host at the bottom, a TMC chassis in the middle, and a VME backplane at the top. The TMC chassis holds the HIPPI interface board set, RAID-II XBUS card(s), and a VME-VME link board to allow the host to control the boards in the chassis. The VME backplane is partitioned into independent 2-slot segments for interfacing the disk drives.

independent AC-DC power converter, so a power supply shelf holds four power supplies. Each of these supplies is independently fused. If one should fail, it cannot affect its neighbors via the AC distribution system. Similarly, a power supply drives four strings on a disk shelf. Each of these is independently fused and switched, to minimize fault propagation in the DC distribution system. Finally each drive is individually fused, further isolating power faults, thus protecting the system from the failure of a single drive.

The XBUS card is the single most complex printed circuit board every designed at Berkeley. It makes extensive use of surface mount technology. We chose surface mount to achieve high component density and to minimize interconnect lengths, especially in the crossbar. To further increase density, we partitioned the design into a main board and four (identical) daughter cards (see Figure 4). The daughter cards interface between the crossbar and the VME interfaces necessary to interface to the ATC string boards. This has the advantage of using the vertical space above and below the main board, In addition, it simplifies the layout task, since the daughter card physical design is done once and used four times.

We designed the boards using RACAL's printed circuit board software. While the design of the daughter card proceeded fairly smoothly using 8 mil lines and spaces, the complexity of the XBUS card eventually exceeded RACAL's ability to handle it. We were forced to wait one month for beta releases to fix the software limitations.

Table 1 summarizes some critical statistics of the two boards we designed, the XBUS and the daughter cards, and compares these with our previous "most complicated board," the SPUR CPU board. In addition, we show the statistics for the original 8-Port version of the XBUS card, which could not be successfully constructed by our board fabricator because of poor yields on blind vias. Blind vias make

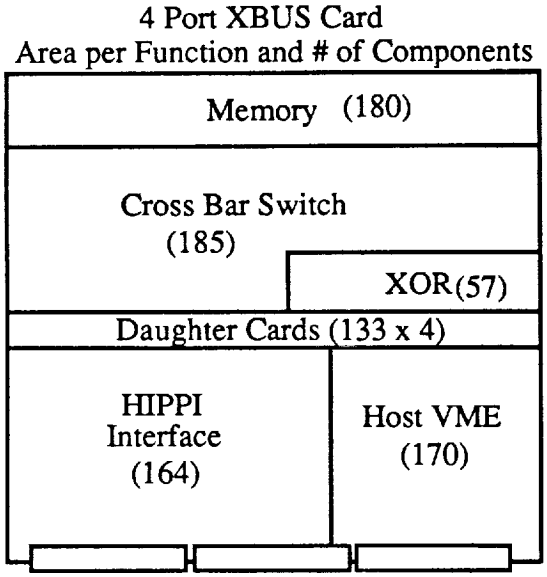

Figure 4: *XBUS Card Layout*
The major subsystems on the board are the interleaved memory system, the crossbar switch, the XOR engine, the daughter card interfaces, the HIPPI source and destination interfaces, and the host VME control interface.

Table 1: IOC PRINTED CIRCUIT BOARDS

Board	XBUS (4-Port)	XBUS (8-Port)	Daughter Card	SPUR CPU Board
Technology	Double Sided SMT/Thru Hole	Double Sided SMT/Blind Vias	SMT/Thru Hole	All Thru Hole
Size, inches	18.4 x 18.3	18.4 x 18.3	6.5 x 8.2	15.7 x 14.44
Layer Count	18 layers, 14 routing	20 layers, 18 routing	10 layers, 8 routing	10 layers, 8 routing
Line/Space (mils)	6/6	6/6	8/8	8/8
Pin Density pins/sq in	54.7	74.5	47	45
# Nets/Connections	1789/13,203	2147/19,171	162/1211	1472/7041
Total Holes	16,429	23,116	2556	12,702
Via Size	0.016	0.016	0.016	0.025

connections between internal layers without taking space on the top and bottom routing layers. Without them, we were forced to reduce the complexity of the board, to reduce the pins/in2. Nevertheless, the 4-port board still has dramatically increased pin densities and number of holes on the board. This is remains a good measure of the evolving complexity of the board-level systems the community is now able to design.

2.1.2. Performance of RAID-II

On January 15, 1993, we were able to demonstrate a fully functional RAID-II controller, simultaneously serving digitized video streams to three SUN workstation clients (two SUN-3s and a SUN-4). Each client consumed approximately 3 MBytes/second (this is about ten uncompressed video frames per second), the effective limit of its memory system.

The demonstration was completely client limited. In a loop-back mode, the controller was able to deliver 38.5 MBytes/seconds from memory to the HIPPI interfaces. This comes very close to the system's design goal of 40 MBytes/second. These are peak performance numbers. Sustained performance for disk transfers was 20 MBytes/second on reads and 18 MBytes/second. This is roughly two orders of magnitude better performance than a single disk drive on a conventional UNIX file server.

The hardware was fully fault tolerant. Part of the January demonstration was the "hot" replacement of a disk drive on which part of the video stream had been stored. After a brief pause of a few hundred milliseconds, the server recovered and was able to deliver the streams at 90% of the rate as before the failure. This is a real testament to the robustness of the hardware and software. The parity calculations used during reconstruction run at memory speeds, and represent minimal overhead.

The performance of RAID-II has been so good that we are encouraged to use it as a storage server in several follow on projects. See Section 4 below for more details.

2.2. Sprite Operating System Group

This contract provided support for five major areas of research within the Sprite network operating system: log-structured file systems, RAID network and disk support, file system analyses, techniques for file server crash recovery, and a merger of Sprite and Mach. The sections below summarize the results from each of these areas.

2.2.1. Log-Structured File Systems

Our most significant result was to design, implement, and analyze a new approach to disk storage management called a log-structured file system, or LFS. An LFS represents all the information on disk in a log-like structure where data can be written sequentially in large blocks. In principle, an LFS can provide an order of magnitude better performance than traditional file systems where the disk is accessed randomly in small blocks. To test the LFS principles, we implemented a prototype LFS as part of the Sprite network operating system and have used it for several years to store more than 10 Gbytes of data. This implementation demonstrates more than 10x improvement in disk efficiency for a variety of workloads and achieves as much as a factor of 60x improvement in some extreme cases. The performance improvements are even greater when RAID disk subsystems are involved: a traditional file system will incur as much as a 400% performance penalty for updating parity blocks on writes, whereas there the penalty is only about 10-15% under LFS.

In the Sprite implementation we solved two significant problems: (a) how to locate information in the log efficiently when it is read, and (b) how to manage the disk to retain large blocks of contiguous free space. Of these problems, the second problem was the most difficult to solve; we used extensive simulations to develop a novel "cost-benefit" policy for garbage collection that segregates old slowly-changing data from younger data that is likely to be deleted quickly. Although we originally feared that the overhead of garbage collection might be as high as 300% relative to the bytes written, measurements of the Sprite implementation show that the cost-benefit policy has an overhead of about 50% of the bytes written. This results in significantly better overall file system performance than any existing disk storage manager for today's engineering/office workloads.

Although the prototype implementation of LFS was made on Sprite, it has been rewritten for BSD UNIX with substantial improvements. This version of LFS should in turn be portable to many of the commercial UNIX-like operating systems.

2.2.2. RAID Network and Disk Support

We carried out several additional projects to support the RAID disk and networking structures. First, we added support to Sprite for Ultranet and FDDI networks. In theory Ultranet can sustain bandwidths of up to 100 Mbytes/sec, and FDDI up to 12 Mbytes/sec. In both cases, performance was disappointing. Using standard workstations we were unable to achieve more than about 3-5 Mbytes/sec. of bandwidth for either network, mostly due to memory bandwidth limitations on the workstations. When the Ultranet was used in conjunction with the RAID controller card we eventually achieved bandwidths up to 10 Mbytes/sec. but only with very large packets. Latency is also a problem with both networks. Ultranet latency is 2-4x Ethernet latency. With performance tuning we were able to get FDDI latency down under Ethernet latency, but only about 20% better; clearly network latency is not improving as fast as network bandwidth, which suggests that newer networks will not benefit operations that transfer only small amounts of data.

We also made a number of improvements in the Sprite disk subsystem to support LFS and disk arrays. The changes included a restructuring of the disk drivers to provide better support for asynchronous operations, plus the development of a striping disk driver to support RAIDs.

Lastly, we developed a new software structure to support high-bandwidth access to RAIDs. Sprite originally used a caching approach to file access, which works well with small files that are overwritten frequently. However, for very large transfers the caching approach results in unnecessary copies, yields little benefit from re-use, and makes it hard to pipeline data flow for maximum performance. Thus we developed an approach where the memory on the RAID controller is used as a buffer rather than a cache, and the software is organized to operate asynchronously with very large transfer sizes and heavy use of pipelining. This approach appears to be simpler than the caching approach while also providing much greater bandwidth for the large file transfers we anticipate. We have also redesigned the protocols between the RAID server and its clients so that clients need not use the Sprite remote procedure call protocol. Instead, clients use standard TCP sockets to connect to the RAID server.

2.2.3. File System Analyses

During the RAID project we carried out three major file system analyses. The first analysis gathered trace data of file system usage on Sprite and used it to measure system usage and the effectiveness of caching in Sprite. We found the following major results: file access rates have increased by a factor of 5-8 over the last 5 years, and the use of process migration can increase throughput by another factor of seven; client-level file caches now achieve read hit rates of 70% or more, using about 4 Mbytes of memory per client; and paging rates have dropped substantially, only accounting for 10-20% of server traffic even on diskless clients.

Our second analysis used the trace data to estimate the effectiveness of name caching in a network file system. We found that client-level name caching is highly effective in a distributed system, even when workstations cooperate closely through the use of process migration, and even for heavily shared directories such as a global /tmp directory. Contrary to our expectations, whole-directory caches are more effective than caches of individual entries, even in the presence of shared directories.

The third analysis used the trace data to estimate the effectiveness of non-volatile memory (NVRAM) for reducing traffic between clients and servers. Although we found that it can indeed reduce the traffic, the current high cost of NVRAM makes it more cost-effective at present to add more volatile memory instead. In the future, when memory sizes get larger and NVRAM prices drop, it appears likely that NVRAM will become cost-effective.

2.2.4. Crash Recovery

Our fourth major research area has been crash recovery. The most common approach to fault-tolerance in computer systems is to try to keep the system from ever crashing, usually at high cost and complexity. We have explored an alternative approach where the system is allowed to crash but

recovers its state transparently and very quickly, so that users are barely aware of the crash.

Sprite's log-structured file system already reduces recovery time dramatically since it can use its log to restore disk consistency without scanning the entire disk. The reduces file system restart time from several tens of minutes on traditional systems to just a few seconds.

Our recovery research also includes several facets related to network file systems, including (a) dealing with server contention during the "recovery storms" that occur when file system clients attempt to reconnect to servers (b) changes to the recovery protocol to reduce the amount of work that must be done at recovery time, and (c) an approach called the "recovery box" in which the server uses non-volatile memory to save enough state so that clients need not participate at all in recovery.

With all of the above techniques combined, we have succeeded in reducing file server recovery time to a few tens of seconds, measured from when the server detects a fatal error until it has rebooted and is responding normally to clients again. We are currently experimenting with additional enhancements that should reduce this time even further.

2.2.5. Sprite-Mach Merger

Our final major project was to see if Sprite could be ported to run as a user-level server process on top of the Mach 3.0 microkernel. We hoped that this would reduce the size of the Sprite kernel and make it easier for us to port Sprite to new platforms. Although we expected some performance penalty in comparison to a native Sprite implementation, we hoped that the penalty would be small. We carried out the port in 1991 and 1992, getting far enough along with it to run many benchmarks and measure the performance.

Sprite-on-Mach was 22% smaller than the native Sprite kernel in lines of code, and it eliminated 95% of the machine-dependent C code (all but 1300 lines) and all but 4 lines of assembler code. Unfortunately Sprite-on-Mach was significantly slower than native Sprite. Simple remote procedure calls slowed down by a factor of almost 3x, and the Andrew file system benchmark originally ran 7x slower on Sprite-on-Mach than on native Sprite. With performance tuning we were able to speed up the Andrew benchmark substantially, but it still ran 2.5x more slowly than native Sprite. The remaining problems have to do with conflicts between Sprite's memory management model and the facilities provided by Mach (which result in excessive copying) and context switching and communication overheads in the Mach kernel.

3. Technology Transfer

Throughout the duration of the project, we have had extensive interactions with industry. Our industrial affiliates, companies that have made significant cash, personnel, or equipment donations to the project, have included: Array Technologies Corporation (a wholly owned Tandem subsidiary), Control Data Corporation, Digital Equipment Corporation, Eastman Kodak, Emulex, Exabyte Corporation, Hewlett-Packard Corporation, Intel Supercomputer Division, International Business Machines, NCR, Open Software Foundation, Seagate, Sequent, Storage Technologies, Sun Microsystems, Thinking Machines Corporation, UltraNetwork Technologies, Inc., and Xerox Corporation. Other companies that have participated in our research retreats include: Amdahl, Auspex Systems, Convex, Interphase, SF2, and Texas Instruments.

A recent market survey, written by Montgomery Securities, predicated that RAID would grow into a $7.8 billion industry by 1994. RAID systems would be available in all major computer system market segments, including supercomputers, mainframes, minicomputers, workstation file servers, and PC file servers. RAID vendors include such diverse companies as Array Technologies Corporation (a Tandem subsidiary), Compaq, Data General, Dell, Hewlett-Packard, IBM, Maximum Strategies, Micropolis, NCR, Storage Dimensions, and StorageTek. It is interesting to note that every single one of these companies is U.S.-based.

4. Future Plans

We plan on using the RAID-II prototype in several follow-on projects over the next 12-18 months. These include: (1) a file server on the Blanca Gigabit testbed, (2) a RAID storage system as part of a distributed, hierarchical storage system being developed as part of the Sequoia Project, and (3) a high capacity, high transaction rate file server for a wireless network. We describe each of these in the next paragraphs.

RAID-II will be used as the high performance file server on a local area gigabit testbed funded by CNRI and the Department of Energy, in conjunction with Lawrence Berkeley Laboratories. The local area network is based on HIPPI switches from Network Systems Corp. and it will interconnect RAID-II, a high speed digitizing camera from PSI Systems, several high performance workstations, and Berkeley's CM-5. Eventually the testbed will be integrated with the Blanca long haul gigabit testbed.

RAID-II is also being used as the disk-level subsystem of an hierarchical storage system that includes optical disk jukeboxes and automated tape robots. This system is being interface to T1/T3 wide area networks to serve as the storage and communications component of the Sequoia Project.

Finally, in conjunction with Professor Bob Brodersen, we are studying strategies to place compressed video and audio streams on top of a multidisk system. The application is to support large numbers of wireless terminals simultaneously viewing multiple compressed video programs.

5. Summary

Relying on new ideas in operating system for file caching and I/O buffering, and exploiting the arrival of low-cost disks, we have demonstrated significant factors of improvement in performance and reliability over current commercially-available systems. We have implemented of a second generation disk array prototype, suitable for supercomputer file service. The operating system group has developed a write optimized log structured file system, and continues to measure its performance. The project has been highly visible, providing the technological vision that is leading to a major new industrial sector dominated by U.S. firms.

6. References

[Baker 90] Baker, M., J.K. Ousterhout, "Availability in the Sprite Distributed File System," Proceeding 4th ACM SIGOPS European Workshop - Fault Tolerance Support in Distributed Systems, Bologna, Italy, (September 1990).

[Baker 91] Baker, M., Hartman, J., Kupfer, M., Shirriff, K., and Ousterhout, J., "Measurements of a Distributed File System," Proc. 13th Symposium on Operating Systems Principles, (October 1991), pp. 198-212.

[Chen 93a] Chen, P. M., E. K. Lee, A. L. Drapeau, K. Lutz, E. L. Miller, S. Seshan, K. Shirriff, D. A. Patterson, R. H. Katz, "Performance and Design Evaluation of the RAID-II Storage Server," International Parallel Processing Symposium Workshop on I/O in Parallel Computer Systems, (April 1993).

[Chen 93b] Chen, P. M., E. K. Lee, A. L. Drapeau, K. Lutz, E. L. Miller, S. Seshan, K. Shirriff, D. A. Patterson, R. H. Katz, "Performance and Design Evaluation of the RAID-II Storage Server," Journal of Parallel and Distributed Databases, in press.

[Drapeau 93] Drapeau, A. L., R. H. Katz, "Analysis of Striped Tape Systems," IEEE Mass Storage Symposium, Monterey, CA, (March 1993).

[Douglis 91] Douglis, F., and Ousterhout, J., "Transparent Process Migration: Design Alternatives and the Sprite Implementation," *Software—Practice and Experience*, Vol. 21, No. 8, (August 1991), pp. 757-785.

[Gibson 90] Gibson, G. A., "Redundant Disk Arrays: Reliable, Parallel Secondary Storage," PhD Dissertation, UCB CSD 91/613, (December 1990).

[Gibson 93] Gibson, G., L. Hellerstein, R. Karp, R. Katz, D. Patterson, "Coding Techniques for Handling Failures in Large Disk Arrays," Algorithmica, in press.

[Hartman 90] Hartman, J., J.K. Ousterhout, "Performance Measurements of a Multiprocessor Sprite Kernel," Proceedings USENIX Summer Conference, (June 1990).

[Hartman 92] Hartman, J.H., J.K. Ousterhout, "Zebra: A Striped Network File System," UCB CSD 92/683, (May 1992).

[Katz 91] Katz, R. H., D. A. Patterson, J. K. Ousterhout, T. E. Anderson, "Robo-Line Storage: Low Latency, High Capacity Storage Systems over Geographically Distributed Networks," UCB CSD 90/651, (November 1991).

[Katz 92a] Katz, R. H., "Network-Attached Storage Systems," Proceedings Scalable High Performance Computing Conference, Williamsburg, VA, (April 1992).

[Katz 92b] Katz, R. H., "High Performance Network- and Channel-Attached Storage," Proceedings of the I.E.E.E., V. 80, N. 8, (August 1992).

[Katz 93a] Katz, R. H., W. Hong, "The Performance of Disk Arrays in Shared Memory Database Machines," Journal of Parallel and Distributed Databases, V. 1, N. 2, (April 1993).

[Katz 93b] Katz, R. H., P. M. Chen, A. L. Drapeau, E. K. Lee, E. L. Miller, S. Seshan, D. A. Patterson, "RAID-II: Design and Implementation of a Large Scale Disk Array Controller," VLSI System Design Conference, Seattle, WA, (March 1993).

[Lee 91] Lee, E. K., R. H. Katz, "An Analytical Model for Disk Array Performance and Its Application," UCB CSD 91/660, (November 1991).

[Lee 93a] Lee, E. K., R. H. Katz, "An Analytic Performance Model of Disk Arrays and Its Application," ACM SIGMETRICS Conference, San Diego, CA, (May 1993).

[Lee 93b] Lee, E. K., R. H. Katz, "The Performance of Parity Placement in Disk Arrays," IEEE Transactions on Computers, in press.

[Miller 91a] Miller, E. L., R. H. Katz, "Input/Output Behavior of Supercomputing Applications," 10th Mass Storage System Symposium, Monterey, CA, (October 1991).

[Miller 91b] Miller, E. L., R. H. Katz, "Input/Output Behavior of Supercomputing Applications," Supercomputing '91, Alburquerque, NM, (November 1991).

[Miller 93a] Miller, E. L., R. H. Katz, "An Analysis of File Migration in a UNIX Supercomputing Environment," Winter 1993 USENIX Conference, San Diego, CA, (January 1993)

[Miller 93b] Miller, E. L., R. H. Katz, "RAMA: A File System for Massively Parallel Computers," IEEE Mass Storage Symposium, Monterey, CA, (March 1993).

[Ousterhout 89] Ousterhout, J. K., F. Douglas, "Beating the I/O Bottleneck: A Case for Log-Structured File Systems," ACM Operating Systems Review, V 23, N 1, (January 1989), pp. 11-28.

[Patterson 88] Patterson, D. A., G. Gibson, R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Proceedings A.C.M. SIGMOD Conference, Chicago, IL, (May 1988), pp. 109-116.

[Rosenblum 90] Rosenblum, M., J.K. Ousterhout, "The LFS Storage Manager," Proceedings USENIX Winter Conference (June 1990).

[Rosenblum 91] Rosenblum, M., J. Ousterhout, "The Design and Implementation of a Log-Structured File System," Proc. 13th Symposium on Operating Systems Principles, (October 1991), pp. 1-15.

[Seshan 93] Seshan, S., R. H. Katz, "Interfacing a High Performance Disk Array File Server to a Gigabit LAN," submitted to ACM SIGCOMM Conference, (March 1993).

[Seltzer 90] Seltzer, M., P. Chen, J. Ousterhout, "Disk Scheduling Revisited," Proceedings USENIX

Winter Conference (January 1990).

[Shirriff 92] Shirriff, K., J. Ousterhout, "A Trace-Driven Analysis of Name and Attribute Caching in a Distributed System," Proc. 1992 Winter USENIX Conference, (January 1992), pp. 315-332.